Universität Bielefeld

Fakultät für Mathematik

Bachelorarbeit

im Studiengang Mathematik

zum Thema:

Primal-Dual Active Set Strategy for Optimal Control Problems with Partial Differential Equations and Control Constraints

---

vorgelegt von

Juan Carlos Chavarría Morales

Matrikel-Nr: 4016452

1. Erstgutachterin: Dr. Tabea Tscherpel

2. Zweitgutachter: Prof. Dr. Lars Diening

Bielefeld, 6. August 2021

Abstract

This work is devoted to the implementation of a numerical strategy to solve Optimal Control Problems containing partial differential equations as restrictions. In addition, we have considered an admissible set restriction for the control variable.

Optimal Control Problems involving the type of restrictions above mentioned have gained importance in the last two decades and have been extensively studied [15],[24],[14], etc. Applications could be found in fields such diverse as Physics, Optimal Design, Pure and Applied Mathematics and Medicine [24], [14], [16]. Therefore, the scope of this Bachelor-Thesis is wide and oriented to audiences with a solid background in Mathematics, but whose main interests could lie in other disciplines.

The Literature [24], [14] consulted normally treats Optimal Control Problems (OCP) starting with classical Poisson equation and then jumping to more complicated models such as Navier-Stokes system. The first goal of this work is to fill this gap introducing an intermediate model in order to simplify the study of more complicated OCP using accessible mathematical language. Nevertheless, the OCP considered is *per se* interesting since it arises in applied sciences such as Fluid Dynamics [11].

Nowadays there exist several computer tools to solve OCP like the one described in this work [3], [21], [1]. In addition, knowledge of concepts from Functional and Numerical Analysis are necessary to provide an adequate theoretical context [24], [14]. The interface from theory to computer implementation requires Finite Element Method (FEM) also necessary as a prerequisite [24], [14]. The second and most important contribution of this work is precisely to use this mathematical framework to computationally implementate a numerical scheme and then compare with the results from a new software released this year [3]. In the same direction, there are good chances after this work to implementate our approach to a wide type of OCP.

Future research include also the study of another type of OCP, for example with new EDP restrictions, or the implementation of different resolution algorithms. The question that remains is the possibility of extending our approach study its limitations. Whatever the case, the learning achieved in this work allows tackling new challenges in the area of Optimal Control with the tools acquired in the development of this Thesis and the support of the Bachelor courses.

Kurzfassung

Diese Arbeit widmet sich der Anwendung einer numerischen Strategie, um optimale Steuerungsprobleme zu lösen, die partielle Differentialgleichungen als Beschränkungen enthalten. Außerdem haben wir eine zulässige Mengenbeschränkung für die Steuerungsvariable betrachtet.

Optimale Steuerungsprobleme (OSP), die die oben genannten Beschränkungen enthalten, haben in den letzten zwei Jahrzehnten an Bedeutung gewonnen und sind intensiv untersucht worden [15],[24],[14] etc. Anwendungen konnten in diversen Bereichen wie z.B. Physik, optimales Design, reine und angewandte Mathematik und Medizin gefunden werden [24], [14], [16]. Daher ist der Umfang dieser Bachelor-Arbeit weit und für Leserinnen und Leser mit einem soliden mathematischen Hintergrundwissen ausgelegt, deren Hauptforschungsinteresse jedoch in anderen Disziplinen liegt.

Die gängige Literatur zum Thema OSP [24], [14] umfasst normalerweise optimale Steuerungsprobleme, die mit der klassischen Poisson-Gleichung beginnen und dann zu komplizierteren Modellen wie dem Navier-Stokes-System übergehen. Das Hauptziel dieser Arbeit ist es, diese Lücke durch Einführen eines Zwischenmodells zu schließen und so die Untersuchung komplizierterer OSP zu erleichtern. Dabei soll eine zugängliche mathematische Sprache genutzt werden. Gleichwohl ist das hier betrachtete OSP an sich interessant, weil es in angewandten Wissenschaften wie z.B. der Strömungsdynamik vorkommt [11].

Heutzutage existieren verschiedene Computerprogramme zur Lösung von wie dem in dieser Arbeit beschriebenen [3], [21], [1] OSP. Darüber hinaus ist als theoretischer Hintergrund die Kenntnis von Konzepten der funktionellen und numerischen Analyse erforderlich. Die Schnittstelle von der Theorie zur Computerimplementierung erfordert als Voraussetzung die Finite-Elemente-Methode (FEM) [24], [14]. Der zweite und wichtigste Beitrag dieser Arbeit ist es, dieses mathematische Rahmenwerk zu nutzen, um ein numerisches Schema anzuwenden und dann mit den Ergebnissen eines in diesem Jahr erschienenen Software-Programms zu vergleichen [3]. So bestehen auch gute Möglichkeiten, mithilfe des Ansatzes dieser Arbeit weitere Arten von OSP zu bearbeiten.

Zukünftige Forschungen könnten auch die Untersuchung anderer Arten von OSP beinhalten, z.B. mit neuen EDP-Beschränkungen oder der Anwendung verschiedener Lösungsalgorythmen. Es bleibt die Frage, inwieweit sich der erarbeitete Ansatz erweitern lässt und wo er an seine Grenzen stößt. In jedem Fall ermöglichen die Erkenntnisse dieser Arbeit, neue Herausforderungen im Bereich der optimalen Steuerung mithilfe der erarbeiteten Hilfsmittel anzunehmen. Nicht zuletzt können diese ein entlastendes Hilfsmittel in Bachelorkursen darstellen.

Declaration of Originality

I hereby declare, Juan Carlos Chavarría Morales, that I wrote the present work independently and without outside help. I also assure that I have not used any other sources than those given and that I have marked all statements taken literally or analogously from other works as such, and that the submitted work was neither completely nor in essential parts the subject of another examination procedure.

Bielefeld, 6 August 2021

Eigenständigkeitserklärung

Hiermit erkläre ich, Juan Carlos Chavarría Morales, dass ich die hier vorliegende Arbeit eigenständig und ohne fremde Hilfe verfasst habe. Ebenso versichere ich, dass ich keine anderen als die angegebenen Quellen und alle wörtlich oder sinngemäß aus anderen Werken übernommenen Aussagen als solche gekennzeichnet habe, und dass die eingereichte Arbeit weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen ist.

Bielefeld, 6. August 2021

Contents

1. Introduction and Motivation **1**
1.1. Motivation . . . 2

2. Optimal Control Problems **4**
2.1. Introduction . . . 4
2.2. Optimal Control Problem for ADR Equation . . . 5

3. Optimality Theory **8**
3.1. Existence of Optimal Controls . . . 8
3.2. First Order Optimality Conditions . . . 9
3.2.1. Quadratic Optimization in Hilbert Spaces . . . 9
3.3. Optimality for the ADR problem . . . 10
3.3.1. On the pointwise optimality conditions . . . 12

4. Finite Element Approximation of the Optimal Control Problem **14**
4.1. A Finite Element Method Scheme . . . 14
4.2. Convergence of the Finite Element Method . . . 14

5. Primal-Dual Active Set (PDAS) Strategy to Solve Optimality System **19**
5.1. PDAS Strategy in Continuous Problem . . . 19
5.2. Mixed formulation of Optimality System . . . 20

6. Numerical Results **23**
6.1. Parameters and exact solutions . . . 23
6.2. Implementation in `FENICS` . . . 23
6.2.1. Numerical Results . . . 24
6.3. Implementation in `cashocs` . . . 25

7. Conclusions and Outlook **28**

A. Implementation in `fenics` **B**
A.1. Main Program . . . B
A.2. Optimality system Solver . . . D
A.3. L^2-Projection . . . E
A.4. Indicator Functions . . . E
A.4.1. χ_a . . . E
A.4.2. χ_b . . . E

B. Implementation in `cashocs` **F**
B.1. Main Program . . . F

1. Introduction and Motivation

Most of the human endeavours involve, or at least try, some degree of control in the behavior of some phenomenona or system. This generates the question: Since we know how a system will behave, how can we *change it* to obtain a *desired* response? The goal of the Control Theory is to answer this question. In particular, this work deals on a system modelated by a partial differential equation (PDE).
A careful reading shows that the *Control Theory* and the *Optimal Control Theory* answer two different but related, questions. Let's analyze this: A general form to formulate a control problem is the following. Given a system, and a way to control it, which can be formulated as

$$A.y = B.u,$$

where A and B are operators, which specify the model of the system and how the control acts on it respectively; y is the state of the system and u is the control, taken from an admissible control set U_{ad}. If in addition a *cost functional* $J(u)$ is given, which expresses the *cost of using the control* u, the question that answers the Optimal Control Theory is to find a control u, in the space of admissible controls U_{ad}, such that

$$\min_{u \in U_{ad}} J(u).$$

In this case, the relation between the control u and the state y is given by a PDE and the boundary conditions. We must proceed carefully in the election of the set of admissible controls. A poor decision will minimise the cost functional, but if this control does not produce the desired results on the system, then the resolution effort is pointless.
Our goal is then to determine an admissible control, called *optimal control* which gives us a satisfactory state and minimices the functional J. The next question to answer is the existence (and unicity) of a pair *control, state* and its respective calculation. To obtain the general solution we must use normally numerical methods. The first step to solve numerically requires a discretisation of the control problem. To this end we proceed using finite elements (see, for example, [24] or [2]). An inmediate consequence of this approach is the error- and convergence analysis of the method. To perform this analysis we will see that it is necessary to impose certain regularity conditions on the optimal control. We can derive this framework from the first order optimality conditions. Another important tool in the error estimates are the second order conditions, which in our case (linear differential equation and quadratic functional cost) are determined by the problem. Once we have a discrete control problem, we must use some optimisation algorithm to solve. In summary, we can list the difficulties in carrying out the study of our optimal control problem as follows [5]:

1. Existence of a solution.
2. Optimality conditions of first and second order (System of PDE's).
3. Discretization of Optimality System.
4. Numerical solution of the associated discrete problem and convergence analysis.

It is important to mention, that this framework corresponds to (1) Continuos Optimisation Problem (2) Optimise (3) Discretise and (4) Solve scheme [10]. However another schemes are possible by switching steps (2) and (3).

1.1. Motivation

This work is devoted to a control problem for the Advection–Diffusion–Reaction (ADR) equation, whose derivation comes from fluid mechanics [25]. Let y be a a scalar function which represents a physical property, such as heat or concentration. If y is conserved and can only vary through exchange between material particles, reaction of the system itself or through external sources. Then the conservation law for y is given by

$$\partial_t y - \varepsilon \Delta y + \underline{\mathrm{b}} \cdot \nabla y + \kappa y = g, \quad \text{in some domain } \Omega, \tag{1.1}$$

where ϵ is the diffusion coefficient, κ is the reaction parameter (which is assumed positive, but could be negative) and $\underline{\mathrm{b}}$ is the velocity field associated with the advective process. Altough the ADR equation arises naturally in fluid mechanics, is often found in several applications such as Economy (Black-Scholes equation), acoustics, heating transfer, medicine and many others [11], [16]. Also, we normally need to prescribe some boundary conditions in Ω. These conditions have a capital importance, because depending on the type of boundary condition (Dirichlet, Neumann or Robin) will determine the kind of optimal control problem to be solved and also the numerical scheme. We will give some directions in the next section. We will consider in particular the stationary ADR equation, which leads to an elliptic partial differential equation. Roughly speaking, the control u will be inserted into the source term in order to produce a desire response, or state of the system y. When is possible to do minimising the value of the cost functional J we say that our solution pair (y, u) is optimal.
We have extensively used two standard texts from the optimal control literature with PDE's constraints, [15], [24]. In these texts the study of optimal control problems begins with the study of Poisson equation:

$$\begin{aligned} & \min\ J(y,u) \\ \text{subject to:} \quad & -\Delta y = u, \qquad \text{in some domain } \Omega \\ & u \in U_{ad}, \end{aligned}$$

plus some boundary condition for the state variable y. Here U_{ad} denotes the admissible set of the control variable u. However, the path to study other optimal control problems governed by more complex equations of state can often be abrupt for the inexperienced reader. Nonetheless, it is possible to find literature dedicated to the Optimal Control Problem for the ADR equation [17], but normally the prerequisites needed could be too technical for an undergraduate mathematic student or a non mathematic reader. This work tries to fill this gap by providing a basic language to explain and solve an Optimal Control Problem which is halfway between the aforementioned Poisson model and more complex problems.

Pseudo Newton methods like the *Primal-dual Active Set Strategy* [15], [24], are extensively used in Optimal Control Problems because their computational efficiency

performing the iterations to find the optimal solution, where the control problem is formulated as a quadratic minimisation problem in which the restrictions are (1) the state equation, in this case a PDE; (2) the boundary conditions and (3) the possible values that the control can reach, the so-called *Admissible Set*. Generally speaking, following the list above and [15], in order to apply this strategy we must first have existence and uniqueness of a solution of the PDE and an optimality condition for the control. This has determined the outline of this work: First section deals with existence and uniqueness of a solution determined by the restriction given by the PDE and the boundary condition. This is achieved using standard arguments from Functional Analysis,[24] ,[4], [2]. In the next section, we determined optimality conditions, where we replicated the scheme showed in [24]. In the last two sections we implementate a Finite Element Method (FEM) to solve our problem together the Primal-dual Active Set Strategy, perform the respective error analysis and show the numerical resolution . For a detailed discussion on this issue we suggest [6] or [24]. In the implementation stage (Section 5.2)cwe have introduced a Mixed Formulation to simplify the computational effort.

2. Optimal Control Problems

2.1. Introduction

Our problem belongs to a large class of minimisation problems. In an abstract form we formulate

$$\min_{u \in U_{ad}} J(y, u), \tag{2.1}$$

where U_{ad} is a closed subset of admissible control functions [22] and $y \in Y$ for some function space to specify which solves the problem

$$\mathcal{D}(y, u) = 0. \tag{2.2}$$

Here $\mathcal{D}$ is some differential operator. We assume that J is bounded from below and that 2.2 is well posed. Our attention lies in a particular subclass of problems, namely

$$\mathcal{D}(y) = u + g, \tag{2.3}$$

where g is a given function in the image space of the operator $\mathcal{D}$. In other optimization problems the control u may also enter the differential operator (for example shape optimization, [19]). We are not concerned in this type of problems, but the simplest case, when $y = 0$ in Γ and 2.3 is obtained solving: find

$$y \in H_0^1(\Omega) = \{y \in H^1(\Omega) : y = 0 \text{ on } \Gamma\} \tag{2.4}$$

such that

$$B(y, v) = (u + g, v), \quad \forall v \in H_0^1(\Omega). \tag{2.5}$$

Here Ω is a bounded connected domain in $\mathbb{R}^d$ with Lipschitz boundary Γ [24] and the functions u and g belong to $L^2(\Omega)$. We have considered Dirichlet boundary condition in this work, but other boundary conditions could be in an analog way presented modifying the cost functional J [24]. B is a bilinear form which represents the weak formulation of an elliptic differential operator. For this problems, we consider integral type cost functionals

$$J(y, u) := \frac{1}{2}\|y - y_\Omega\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\|u\|^2_{L^2(\Omega)}, \quad \lambda > 0,$$

and the set of admissible control functions includes *box* constraints, namely,

$$U_{ad} := \{u \in L^2(\Omega) : u_a \leq u(x) \leq u_b, \text{ a.e. } x \in \Omega\}.$$

It is well known that under the above assumptions these optimal control problems have unique solutions (e.g., see Section 2, Chapter 2 in [18]). Although literature [24], [14], etc. normally enfolds the theory through the optimal control problem for the Poisson equation, we focus our efforts in the optimal control problem for the ADR equation. This problem has *per se* a physical relevance [11], but also from the pure mathematic point of view is interesting because it is useful to introduce more complicated control problems like Navier Stokes equations [7].

2.2. Optimal Control Problem for ADR Equation

Our work is devoted to the *distributed* Optimal Control Problem for the stationary ADR equation

$$\min J(y,u) \quad \text{where} \quad J(y,u) := \frac{1}{2}\|y - y_\Omega\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\|u\|^2_{L^2(\Omega)}, \tag{2.6}$$

subject to the restrictions:

1. The *system to state* condition, given by the Advection-Diffusion-Reaction (ADR) equation, with homogeneous Dirichlet boundary condition

$$\begin{cases} -\varepsilon\Delta y + \underline{b}\cdot\nabla y + \kappa y &= u + g \quad \text{in } \Omega, \\ y &= 0 \qquad \text{on } \Gamma, \end{cases} \tag{2.7}$$

2. The *Admissibility or Box* condition for the control variable u

$$u \in U_{ad} := \{u \in L^2(\Omega) : u_a \le u(x) \le u_b, \quad \text{a.e. } x \in \Omega\}. \tag{2.8}$$

We say that the control problem is *distributed* because the control u acts in the whole domain Ω. Depending on the boundary condition the cost functional J could be adapted to the control requirements. See for example [24], [14], etc. We make the following assumptions:

Supposition **2.1.** *For the problem* (2.6) – (2.8) *we have that:*

- $\lambda > 0$, $\varepsilon > 0, \kappa \ge 0$ *are given scalars.* $y_\Omega \in L^2(\Omega)$*, represents the desire state,* $g \in L^2(\Omega)$ *and* $\underline{b} \in W^{1,\infty}(\Omega)$ *is a null divergence vector field in* Ω*.*
- *The bounds* $u_a < u_b$ *define the sets of admissible controls acting on* Ω*. The elements of* U_{ad} *are called admissible Controls.*

We define the norm of our solution space $H^1_0(\Omega)$ as

$$\|v\|_\Omega := \left(\varepsilon\|\nabla v\|^2_{L^2(\Omega)} + \kappa\|v\|^2_{L^2(\Omega)}\right)^{1/2},$$

which is equivalent to the norm $H^1(\Omega)$. In fact, the next existence result for (2.7) explains this claim.

Theorem **2.1 Existence.** *Under the assumptions (2.1), for each* $u \in U_{ad}$ *there exists an unique solution* $y \in H^1_0(\Omega)$ *of* (2.7)*. Moreover, there exists a constant* $c_P > 0$*, which only depends on* Ω *such that*

$$\|y\|_\Omega \le c_p\left(\|g\|_{L^2(\Omega)} + \|u\|_{L^2(\Omega)}\right). \tag{2.9}$$

Proof. Following [18] the idea of the proof is to check Lax–Milgram's Lemma in order to obtain the well-posedness of the problem and the uniqueness of a solution when the

data $u+g$ is given. [1] To do this, we use the weak problem associated to (2.7). In other words, we want to find $y \in H_0^1(\Omega)$ such that $B(y,v) = F(v)$, for all $v \in H_0^1(\Omega)$, where

$$B(y,v) = \int_\Omega (\varepsilon \nabla y \cdot \nabla v + \underline{\mathrm{b}} \cdot \nabla y\, v + \kappa y v)\, dx \tag{2.10}$$

$$G(v) = \int_\Omega (u+g) v\, dx. \tag{2.11}$$

In order to apply the result of Lax-Milgram we must verify that: (1) B is continuous, which is $|B(u,v)| \leq C_{cont}\|y\|_\Omega \|v\|_\Omega$, $\forall y, v \in H_0^1(\Omega)$. (2) B is coercive. That is $B(v,v) \geq C_{coer}\|v\|_\Omega^2$, $\forall v \in H_0^1(\Omega)$ and (3) G is continuous in the sense: $G(v) \leq c\|v\|_\Omega$, $\forall v \in H_0^1(\Omega)$. First, we have

$$\begin{aligned}
|B(y,v)| =& \left| \int_\Omega (\varepsilon \nabla y \cdot \nabla v + \underline{\mathrm{b}} \cdot \nabla y v + \kappa y v)\, dx \right| \\
\leq& \varepsilon \|\nabla y\|_{L^2(\Omega)} \|\nabla v\|_{L^2(\Omega)} + \|\,\underline{\mathrm{b}}\,\|_{L^\infty(\Omega)} \|\nabla y\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} + \kappa \|y\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \\
=& \sqrt{\varepsilon}\|\nabla y\|_{L^2(\Omega)} \sqrt{\varepsilon}\|\nabla v\|_{L^2(\Omega)} + \frac{\|\,\underline{\mathrm{b}}\,\|_{L^\infty(\Omega)}}{\sqrt{\varepsilon\kappa}} \sqrt{\varepsilon}\|\nabla y\|_{L^2(\Omega)} \sqrt{\kappa}\|v\|_{L^2(\Omega)} \\
&+ \kappa \|y\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)} \\
\leq& \max\left\{1, \frac{\|\,\underline{\mathrm{b}}\,\|_{L^\infty(\Omega)}}{\sqrt{\varepsilon\kappa}}\right\} \left(\sqrt{\varepsilon}\|\nabla y\|_{L^2(\Omega)} + \sqrt{\kappa}\|y\|_{L^2(\Omega)}\right) \left(\sqrt{\varepsilon}\|\nabla v\|_{L^2(\Omega)} + \sqrt{\kappa}\|v\|_{L^2(\Omega)}\right) \\
\leq& 2 \max\left\{1, \frac{\|\,\underline{\mathrm{b}}\,\|_{L^\infty(\Omega)}}{\sqrt{\varepsilon\kappa}}\right\} \|y\|_\Omega \|v\|_\Omega.
\end{aligned}$$

We have used Cauchy–Schwarz and $(a+b)^2 \leq 2(a^2+b^2)$ inequalities and *built* the norm of the state space. Defining $C_{cont} = 2\max\left\{1, \frac{\|\,\underline{\mathrm{b}}\,\|_{L^\infty(\Omega)}}{\sqrt{\varepsilon\kappa}}\right\}$ we are done.
To proof the second statement, we note that $\forall v \in H_0^1(\Omega)$:

$$B(v,v) = \int_\Omega \left(\varepsilon \nabla v \cdot \nabla v + \kappa v^2\right) dx + \int_\Omega \underline{\mathrm{b}} \cdot \nabla v\, v\, dx = \|v\|_\Omega^2 + \int_\Omega \underline{\mathrm{b}} \cdot \nabla v\, v\, dx.$$

We claim that the second term in the right hand side is null. To see this, we use the identity $\nabla \cdot (v^2\, \underline{\mathrm{b}}) = 2v \nabla v \cdot \underline{\mathrm{b}} + v^2 \nabla \cdot \underline{\mathrm{b}}$, which gives

$$\int_\Omega \underline{\mathrm{b}} \cdot \nabla v\, v\, dx = \frac{1}{2} \int_\Omega \nabla \cdot (v^2\, \underline{\mathrm{b}})\, dx - \frac{1}{2} \int_\Omega v^2 \nabla \cdot \underline{\mathrm{b}}\, dx.$$

The second integral in the right hand side is zero since $\underline{\mathrm{b}}$ is a null divergence vector field. We use Gauß theorem for the remaining integral to obtain

$$\int_\Omega \underline{\mathrm{b}} \cdot \nabla v\, v\, dx = \frac{1}{2} \int_\Gamma (v^2\, \underline{\mathrm{b}}) \cdot d\underline{\sigma} = 0, \qquad \text{since } v \in H_0^1(\Omega).$$

[1] The result can be found in numerous texts of Functional Analysis. For the sake of simplicity we use the standard reference of this report, [24].

Then $B(v,v) = \|v\|_\Omega^2 = C_{coerv}\|v\|_\Omega^2$, with $C_{coerv} = 1$ and we have our result. Finally, using Cauchy–Schwarz inequality and the definition of norm of the state space we obtain

$$G(v) \leq \|g+u\|_{L^2(\Omega)}\|v\|_{L^2(\Omega)} = \frac{\|g+u\|_{L^2(\Omega)}}{\sqrt{\kappa}}\sqrt{\kappa}\|v\|_{L^2(\Omega)} \leq \frac{\|g\|_{L^2(\Omega)} + \|u\|_{L^2(\Omega)}}{\sqrt{\kappa}}\|v\|_\Omega.$$

Choosing $c = \frac{\|g\|_{L^2(\Omega)}+\|u\|_{L^2(\Omega)}}{\sqrt{\kappa}}$ we have that G is continuous.
For the last part, if we set $v = y$ and use the previous results in the weak formulation we obtain

$$\begin{aligned} C_{coerv}\|y\|_\Omega^2 \leq B(y,y) &= G(y) \leq \frac{\|g\|_{L^2(\Omega)} + \|u\|_{L^2(\Omega)}}{\sqrt{\kappa}}\|y\|_\Omega \\ \Rightarrow \|y\|_\Omega &\leq \frac{1}{\sqrt{\kappa}}\left(\|g\|_{L^2(\Omega)} + \|u\|_{L^2(\Omega)}\right), \end{aligned}$$

which gives the continuous dependence on data. The proof is then complete. □

REMARK **2.1.** *The theorem* (2.1) *gives a relation between the state y and the control u, say $y(u)$. This allows us to write 2.6 in terms of u.*

Now we focus our attention on obtain optimality conditions for the optimal control problem.

DEFINITION **2.1.** *We call a control $\bar{u} \in U_{ad}$ optimal and its associated state optimal state $\bar{y} = y(\bar{u})$ if*

$$J(\bar{y},\bar{u}) \leq J(y(u),u), \qquad \forall u \in U_{ad}.$$

Theorem 2.1 guarantees the existence of the following continuous linear operator in view of 2.9:

DEFINITION **2.2.** *The* Control to State *operator is defined as $\Xi : L^2(\Omega) \to H_0^1(\Omega)$, which assigns a state $y(u)$ to each control u.*

Since $\|y\|_{L^2} \leq \|y\|_{H^1}$ then the state space $H_0^1(\Omega) \subset H^1(\Omega)$ is linear embedded in $L^2(\Omega)$ (see [24]). Therefore, we can *extend* the range of Ξ to $L^2(\Omega)$ through the following operator $E_Y : H^1(\Omega) \to L^2(\Omega)$ which to each function of H^1 asignates the same element in L^2, which is also linear and continuous. In this way, through composition we can define the Solution Operator

$$S = E_Y\Xi : L^2(\Omega) \to L^2(\Omega),$$

where $S(u) = y(u)$. This operator allows us to work with admissible controls and reduce (2.9) to the following minimisation problem

$$\min_{u\in U_{ad}} f(u) := \frac{1}{2}\|Su - y_\Omega\|_{L^2(\Omega)}^2 + \frac{\lambda}{2}\|u\|_{L^2(\Omega)}^2. \tag{2.12}$$

We say that f is the *Reduce Functional.*

3. Optimality Theory

Taking 2.12 as starting point, we develop the optimality conditions for our Optimal Control Problem. Like in Calculus in one or several variables, when one optimises a function some conditions for existence of critical points are needed. Further, one seeks for a classification of these critical points using to that end Hessian matrices and eventually Karush-Kuhn-Tucker conditions. A relief in this context is obtain when the objective function is convex, which ensures that the founded critical points are minimums. In this new context, we try to proof existence and unicity of functions (control functions, and through $y = Su$ state functions) which minimise the reduced functional. Therefore, we need to extend the definitions from classical Calculus to function spaces and to profit from the convexity of the reduced functional 2.12. The first two parts of this section deal with these matters, while in the last one we apply the results of the previous subsections to the ADR problem, which leads to a first version of the optimality system which will allow us to solve our problem.

3.1. Existence of Optimal Controls

Our first result deals with the existence of an Optimal Control for the minimisation problem of the reduced functional (2.12). This argument will be replicated in other contexts and therefore we present the most general version. We refer to [24] or [14] for a detailed proof

Theorem **3.1.** *Let $\{U, \|\cdot\|_U\}$ and $\{H, \|\cdot\|_H\}$ two real Hilbert spaces and $U_{ad} \subset U$ a bounded closed convex non-empty set, as well as $y_d \in H$ and a constant $\lambda \geq 0$ given. Moreover, let $S : U \to H$ be a linear continuous operator. Then the quadratic optimization problem on Hilbert spaces*

$$\min_{u \in U_{ad}} f(u) := \frac{1}{2}\|Su - y_d\|_H^2 + \frac{\lambda}{2}\|u\|_U^2 \tag{3.1}$$

admits an optimal solution $\bar{u}$. If besides $\lambda > 0$ or S is injective, then the solution is uniquely determined.

As a direct consequence of the preceding result, we obtain the existence and uniqueness for the elliptic optimal control problem (2.6)–(2.8) [24]:

Theorem **3.2.** *Suppose that the conditions of* (2.1) *are satisfied. Then, the problem* (2.6)–(2.8) *with $g = 0$ possesses at least one optimal control $\bar{u}$. If, in addition $\lambda > 0$, the solution is unique.*

Proof. We apply the previous theorem for $U = H = L^2(\Omega)$, $y_d = y_\Omega$ and $S = E_Y G$. The set $U_{ad} = \{u \in L^2(\Omega) : u_a \leq u \leq u_b \text{ a.e. in } \Omega\}$ is closed, bounded and convex. Then, from the theorem (3.1) it follows that the minimisation problem (2.12) admits at least one solution $\bar{u}$ which is unique if $\lambda > 0$. If $Su = 0$, we have that the operator S for this problem is injective. In fact, if $Su = 0$, then $y = 0$ for (2.9) and inserting this into the differential equation follows that $u = 0$ almost everywhere in Ω. This gives the injectivity of S and we have uniqueness for this case. □

REMARK **3.1.** *In the proof of last theorem, $\bar{u}$ we obtain the limit of a weakly convergent sequence $\{u_{n_k}\}$. Since the control-to-state operator $\Xi : \mathrm{L}^2(\Omega) \to H_0^1(\Omega)$ is a continuous linear operator, it is also weakly continuous. This implies that the series of states $\{y_{n_k}\}$ converges weakly in $H_0^1(\Omega)$ to $\bar{y} = \Xi\bar{u}$. Also, the case $g \neq 0$ could be in an analog way treated simply defining $w \coloneqq u + g \in L^2(\Omega)$, rescaling U_ad and repeating the previous proof.*

It is possible [24] extend the set U_{ad} allowing $u_a = -\infty$ and/or $u_b = +\infty$. In this case, U_{ad} is not bounded and therefore is not sequentially weak compact. However, we still have existence for the case $\lambda > 0$.

3.2. First Order Optimality Conditions

The aim of this section is to provide similar arguments to the Calculus in several variables for optimality conditions in the context of our cost functional. The necessary conditions that we will find allow us to go deeper into the structure of an optimal control and the numerical verification of its optimality.

3.2.1. Quadratic Optimization in Hilbert Spaces

In order to prove the existence of optimal controls, we have transformed the control problems under analysis into quadratic optimization problems reduced in terms of u to

$$\min_{u \in U_{ad}} f(u) := \frac{1}{2}\|Su - y_d\|_H^2 + \frac{\lambda}{2}\|u\|_U^2. \tag{3.2}$$

For this minimisation problem, the following fundamental result can be applied. It is the key to deriving first-order necessary conditions in the presence of control constraints and is in complete analogy to the well known theory for optimality conditions for functions in $\mathbb{R}^d$. The proof could be found in [24] or [14].

LEMMA **3.1.** *If C denotes a non-empty convex subset of a real Banach space U, and f is a real Gâteaux-differentiable application on an open subset of U containing C. If $\bar{u}$ is the solution of the problem*

$$\min_{u \in C} f(u),$$

then it solves the inequality

$$f'(\bar{u})(u - \bar{u}) \geq 0 \qquad u \in C. \tag{3.3}$$

Reciprocally, if $\bar{u} \in C$ solves the above inequality and f is convex, then $\bar{u}$ is a solution of the minimisation problem $\min_{u \in C} f(u)$.

The above result gives a *necessary* condition, which is *sufficient* in the case of convexity. It is also called *first-order optimality condition.* Let us apply this result to the problem (3.2).

THEOREM **3.3.** *Suppose that we have U, H Hilbert spaces, a non-empty convex subset $U_{ad} \subset U$, some $y_d \in H$ and $\lambda \geq 0$ are given. Moreover, let $S : U \to H$ be a linear and*

continuous operator. Then $\bar{u} \in U_{ad}$ is a solution of the minimisation problem (3.2) *if and only if it satisfies the variational inequality*

$$(S^*(S\bar{u} - y_d) + \lambda\bar{u}, u - \bar{u})_U \geq 0, \quad \forall u \in U_{ad}. \tag{3.4}$$

Proof. Usig the chain rule for Gâteaux and Fréchet derivatives (see, for example Theorem 2.20 in [24]), the gradient of the functional defined by (3.2) has the form

$$f'(\bar{u}) = S^*(S\bar{u} - y_d) + \lambda\bar{u}, \tag{3.5}$$

where S^* denotes the *adjoint* operator of S. Then our result follows as consequence of the previous Lemma. □

Next, we focus our attention in this operator S^*.

3.3. Optimality for the ADR problem

The ADR problem defined in the first section states that under the assumptions (2.1) we must find a pair (y, u) such that minimises $J(y, u)$ defined by 2.6 subject to 2.7 and 2.8.
We will denote by S to the solution operator of the boundary value problem, which according to the previous section muss satisfies the variational inequality (3.4) adapted to our case, that is

$$(S^*(S\bar{u} - y_\Omega) + \lambda\bar{u}, u - \bar{u})_U \geq 0, \qquad \forall u \in U_{ad}, \tag{3.6}$$

where the adjoint operator S^* is yet to be determined. We replicate the analysis of [24] for our case in order to determine S^*. To that end, we refer to the next pair of problems: *Find $y, p \in H_0^1(\Omega)$, such that for $u, z \in L^2(\Omega)$ given, solve*

$$\left\{ \begin{array}{rcll} -\varepsilon\Delta y + \underline{b}\cdot\nabla y + \kappa y &=& u + g & \text{in } \Omega, \\ y &=& 0 & \text{on } \Gamma, \end{array} \right. \quad \left\{ \begin{array}{rcll} -\varepsilon\Delta p - \underline{b}\cdot\nabla p + \kappa p &=& z & \text{in } \Omega, \\ p &=& 0 & \text{on } \Gamma, \end{array} \right. \tag{3.7}$$

We claim that the problem on the right is the dual problem associated to our state equation. To see this, we consider $p, \in H_0^1(\Omega)$ as test function for the problem of the left side to write the weak formulation of this problem, which yields

$$\begin{aligned} &\int_\Omega \{\varepsilon\nabla y \cdot \nabla p + (\underline{b}\cdot\nabla y + \kappa y)\ p\}\, dx = \\ &\int_\Omega \{\varepsilon\nabla p \cdot \nabla y - \underline{b}\cdot\nabla p\ y + \kappa p\ y\ \}\, dx = \int_\Omega (u + g)p\, dx. \end{aligned}$$

The later is the weak formulation of the problem at the right hand side in 3.7, where $y \in H_0^1(\Omega)$ is used this time as test function. This allow us to conclude

$$\int_\Omega zy\, dx = \int_\Omega p(u + g)\, dx. \tag{3.8}$$

In view of our previous discussion $S : L^2(\Omega) \to L^2(\Omega)$, we have that $S^* : L^2(\Omega) \to L^2(\Omega)$ and this is given by

$$(z, Su)_{L^2(\Omega)} = (S^* z, u)_{L^2(\Omega)}, \qquad \forall z, u \in L^2(\Omega)\,.$$

Now, taking $u = u + g$, we have that $y = Su$, and we find that

$$(z, Su)_{L^2(\Omega)} = (z, y)_{L^2(\Omega)} = (p, u + g)_{L^2(\Omega)},$$

according to (3.8). The mapping $z \mapsto p$ is linear and continuous from $L^2(\Omega) \to L^2(\Omega)$ and given that z, u could be arbitrary choose and S^* is uniquely determined. Summarizing:

LEMMA **3.2.** *For the boundary value problem* (2.7)*, the adjoint operator* $S^* : \mathrm{L}^2(\Omega) \to \mathrm{L}^2(\Omega)$ *is given by*

$$S^* z = p,$$

where $p \in H_0^1(\Omega)$ *is solution of the boundary value problem*

$$\begin{aligned} -\varepsilon \Delta p - \underline{\mathrm{b}} \cdot \nabla p + \kappa p &= z, \qquad \text{in } \Omega \\ p &= 0, \qquad \text{on } \Gamma \end{aligned} \tag{3.9}$$

DEFINITION **3.1.** *The weak solution* $p \in H_0^1(\Omega)$ *of the adjoint problem 3.9 is called* adjoint state *associated to* $\bar{y}$*.*

The right hand side of the adjoint equation belongs to $L^2(\Omega)$ since $y_\Omega \in L^2(\Omega)$ and $\bar{y} \in Y = H_0^1(\Omega) \hookrightarrow L^2(\Omega)$. By (2.1), we have that (3.9) admits a unique solution $p \in H_0^1(\Omega)$.

Making $z = y - y_\Omega$, we conclude from the previous Lemma that

$$S^*(S\bar{u} - y_\Omega) = S^*(\bar{y} - y_\Omega) = p, \tag{3.10}$$

which, applied to (3.6) gives

$$(p + \lambda \bar{u}, u - \bar{u})_{L^2(\Omega)} \geq 0, \quad \forall u \in U_{ad}.$$

Thus, the next result follows directly from the variational inequality

THEOREM **3.4.** *Suppose that* $\bar{u}$ *is an optimal control for the ADR problem and let* $\bar{y}$ *their associated state. Then the adjoint equation* (3.9) *has an unique weak solution which satisfies the variational inequality*

$$\int_\Omega \left(p(x) + \lambda \bar{u}(x)\right)\left(u(x) - \bar{u}(x)\right)\, dx \geq 0, \qquad \forall u \in U_{ad}. \tag{3.11}$$

Reciprocally, any control $\bar{u} \in U_{ad}$ *which together with the associated state* $\bar{y} = y(\bar{u})$ *and the solution* p *of* (3.9)*, satisfies the variational inequality* (3.11) *is optimal.*

The sufficiency of the previous theorem follows from the convexity of f. Therefore, a control u, together with the optimal state y and the adjoint state p, for the ADR problem is optimal if the triple $(y, p, u) \in H_0^1(\Omega) \times H_0^1(\Omega) \times U_{ad}$ satisfies the following *optimal system*

$$\boxed{\begin{aligned} &-\varepsilon \Delta y + \underline{\mathrm{b}} \cdot \nabla y + \kappa y = u + g \\ &-\varepsilon \Delta p - \underline{\mathrm{b}} \cdot \nabla p + \kappa p = y - y_\Omega \\ &(p + \lambda \bar{u}, u - \bar{u})_{L^2(\Omega)} \geq 0, \quad \forall u \in U_{ad}. \end{aligned}} \tag{3.12}$$

3.3.1. On the pointwise optimality conditions

The last inequality in our optimality system 3.12 is still not enough to perform any numerical intent to solve our OCP. However, we will see that using some projection operator we can establish an optimality system consisting of 3 equations. To that end, we formulate the variational inequality (3.11) from the *pointwise* point of view. The next result marks the first step. We follow the outline proposed in [24].

Lemma **3.3.** *A necessary and sufficient condition to satisfy the variational inequality* (3.11) *is that for almost every* $x \in \Omega$,

$$\bar{u}(x) = \begin{cases} u_a & \text{if} \quad p(x) + \lambda\bar{u}(x) > 0 \\ \in [u_a, u_b] & \text{if} \quad p(x) + \lambda\bar{u}(x) = 0 \\ u_b & \text{if} \quad p(x) + \lambda\bar{u}(x) < 0. \end{cases} \tag{3.13}$$

An equivalent condition is given by the variational inequality in $\mathbb{R}$,

$$(p(x) + \lambda\bar{u}(x))\,(v - \bar{u}(x)) \geq 0 \qquad \forall v \in [u_a, u_b],\ \text{a.e. } x \in \Omega. \tag{3.14}$$

Next, we note that a simple rearrangement of terms in (3.14) gives

$$(p(x) + \lambda\bar{u}(x))\,\bar{u}(x) \leq (p(x) + \lambda\bar{u}(x))\,v \qquad \forall v \in [u_a, u_b], \tag{3.15}$$

for almost $x \in \Omega$. Here like in (3.14), v is understand as a constant.

Theorem **3.5.** *A control* $\bar{u} \in U_{ad}$ *is optimal for* (2.6)–(2.8) *if and only if satisfies together with the adjoint state* p *of* (3.9) *one of the following minimisation conditions for almost all* $x \in \Omega$*:*

- Weak minimum principle

$$\min_{v \in [u_a, u_b]} \{(p(x) + \lambda\bar{u}(x))\,v\} = (p(x) + \lambda\bar{u}(x))\,\bar{u}(x).$$

- *Or* The minimum principle

$$\min_{v \in [u_a, u_b]} \left\{ p(x)v + \frac{\lambda}{2}v^2 \right\} = p(x)\bar{u}(x) + \frac{\lambda}{2}\,\bar{u}(x)^2.$$

The point conditions obtained can be used to obtain more information if we know the sign of λ. The consequences are

Case 1: $\boldsymbol{\lambda = 0}$. In this case, using (3.13) we have that a.e.,

$$\bar{u}(x) = \begin{cases} u_a & \text{if} \quad p(x) > 0 \\ u_b & \text{if} \quad p(x) < 0. \end{cases} \tag{3.16}$$

In the points $x \in \Omega$ where $p(x) = 0$, there is no information of $\bar{u}(x)$ that we can obtain. Otherwise, we say that $\bar{u}$ is a *bang-bang control*, which means that the values of $\bar{u}$ coincide a.e. with some of the bounds u_a or u_b.

Case 2: $\lambda > 0$. We could interpretate that the second of the relations in (3.13) noting that $\bar{u}$ is undeterminaded if $\lambda\bar{u} + p = 0$. However, this is incorrect, since these relation inmediatly gives that

$$\bar{u}(x) = -\frac{1}{\lambda}p(x)$$

and therefore is an indicator to use the minimum condition. This is specified with the following result.

THEOREM **3.6.** *If $\lambda > 0$, then $\bar{u}$ is optimal for (2.6)–(2.8) if and only if satisfies, together with the adjoint state p, the projection formula*

$$\bar{u}(x) = \Pi^{\perp}_{[u_a,u_b]}\left(-\frac{1}{\lambda}p(x)\right) \qquad a.e.\ x \in \Omega, \tag{3.17}$$

where, for real numbers $a \leq b$, $\Pi^{\perp}_{[a,b]}$ denotes the projection of $\mathbb{R}$ on $[a, b]$,

$$\Pi^{\perp}_{[a,b]}(u) := \min\{b, \max\{a, u\}\}.$$

Proof. This is a direct consequence of the previous theorem. In fact, the solution of the quadratic optimisation problem in $\mathbb{R}$ formulated in terms of the minimum principle

$$\min_{v\in[u_a,u_b]}\left\{p(x)v + \frac{\lambda}{2}v^2\right\} = p(x)\bar{u}(x) + \frac{\lambda}{2}\,\bar{u}(x)^2$$

is given by the projection formula (3.17). □

Cases $\lambda > 0$ and $U_{ad} = L^2(\Omega)$ are not considered in this work. We suggest [24] for a study of this situations. We will implementate the finite element method in the next sections on the case $\lambda > 0$ and U_{ad} given by 2.8.

4. Finite Element Approximation of the Optimal Control Problem

Following our outline, after establishing the foundations of existence and uniqueness of the triple (y, p, u), we focus our attention in the discretisation of the optimal system 3.12 and its convergence. This is the first stage of our resolution algorithm. Later we will solve the discrete optimization problem with the help of the strategy shown in the next section.

4.1. A Finite Element Method Scheme

From previous sections, we know that an equivalent formulation of the control problem is given by the non-linear system of equations:
Find $(y, p, u) \in H_0^1(\Omega) \times H_0^1(\Omega) \times U_{ad}$ *such that*

$$\left\{ \begin{array}{rcll} B(y,v) - (u,v)_{L^2(\Omega)} & = & (g,v)_{L^2(\Omega)} & \forall\ v \in H_0^1(\Omega), \\ -(y,v)_{L^2(\Omega)} + B^*(p,v) & = & -(y_\Omega, v)_{L^2(\Omega)} & \forall\ v \in H_0^1(\Omega), \\ (p + \lambda u, \bar{u} - u)_{L^2(\Omega)} & \geq & 0 & \forall\ \bar{u} \in U_{ad}, \end{array} \right. \tag{4.1}$$

where the bilinear forms are given by

$$B(y,v) = \int_\Omega (\varepsilon \nabla y \cdot \nabla v + \underline{\mathrm{b}} \cdot \nabla y\, v + \kappa y v)\ dx \quad \text{and}$$
$$B^*(p,v) = \int_\Omega (\varepsilon \nabla p \cdot \nabla v - \underline{\mathrm{b}} \cdot \nabla p\, v + \kappa p v)\ dx.$$

In order to introduce a FEM scheme we consider the following Lagrange finite element space:

$$V_h := \{ v \in \mathcal{C}^0(\overline{\Omega}) :\ v_{|K} \in \mathbb{P}_1(K),\ \forall\ K \in \mathcal{P} \} \cap H_0^1(\Omega), \tag{4.2}$$

where $\mathcal{P}$ refers to a triangulation of Ω, formed by triangular elements K. Also $\mathbb{P}_l(K)$ is the set of all polynomials of degree less or equal to $l \in \mathbb{N}_0$ and $v_{|K}$ represents the restriction of the function v to the element K. To approximate the control variable we define in the same way the set,

$$U_{ad}^h = \{ t \in L^\infty(\Omega) :\ t_{|K} \in \mathbb{P}_0(K),\ \forall\ K \in \mathcal{P} \} \cap U_{ad}. \tag{4.3}$$

Now, the finite element discretisation for our control problem is the following:
Find $(y_h, p_h, u_h) \in V_h \times V_h \times U_{ad}^h$ *such that*

$$\left\{ \begin{array}{rcll} B(y_h,v_h) - (u_h,v_h)_{L^2(\Omega)} & = & (g_h,v_h)_{L^2(\Omega)} & \forall\ v_h \in V_h, \\ -(y_h, w_h)_{L^2(\Omega)} + B^*(p_h,w_h) & = & -(y_\Omega, w_h)_{L^2(\Omega)} & \forall\ w_h \in V_h, \\ (p_h + \lambda u_h, \bar{u}_h - u_h)_{L^2(\Omega)} & \geq & 0 & \forall\ \bar{u} \in U_{ad}^h. \end{array} \right. \tag{4.4}$$

4.2. Convergence of the Finite Element Method

In what follows we will deal with the study of the convergence of the finite element scheme. For this purpose, we will rewrite the previous problems in a convenient way:

$$\left\{ \begin{array}{c} B(y,v) = (g+u,v)_{L^2(\Omega)}, \\ B^*(p,v) = (y - y_\Omega, v)_{L^2(\Omega)}, \\ (p + \lambda u, \bar{u} - u)_{L^2(\Omega)} \geq 0, \end{array} \right. \qquad \left\{ \begin{array}{c} B(y_h,v_h) = (g+u_h,v_h)_{L^2(\Omega)}, \\ B^*(p_h,v_h) = (y_h - y_\Omega, v_h)_{L^2(\Omega)}, \\ (p_h + \lambda u_h, \bar{u}_h - u_h)_{L^2(\Omega)} \geq 0. \end{array} \right. \tag{4.5}$$

It is useful to recall that in the previous systems that each state, weak or dicrete depends on its respective control, that is, $y = y(u)$ or $y_h = y_h(u_h)$. In addition we define the following auxiliar problems: Given the optimal control $u \in U_{ad}$ and its discretisation $u_h \in U_{ad}^h$, we define the pair $(y_h(u), r_h) \in V_h \times V_h$, as the solution of the system

$$\begin{cases} B(y_h(u), v_h) &= (g + u, v_h), \\ B^*(r_h, v_h) &= (y_h(u) - y_\Omega, v_h). \end{cases} \tag{4.6}$$

Besides we define $q_h \in V_h$ as the solution of

$$B^*(q_h, v_h) = (y(u) - y_\Omega, v_h). \tag{4.7}$$

The previous equalities are valid por any $v_h \in V_h$. Finally we define the following orthogonal projection,

$$\Pi^\perp : L^2(\Omega) \to \mathbb{P}_0(\mathcal{P}),$$

which for each element $K \in \mathcal{P}$, fulfills

$$\int_K (j - \Pi^\perp(j)) = 0.$$

We note that $\Pi^\perp(u) \in U_{ad}$ if $u \in U_{ad}$. In fact, since $u_a \leq u \leq u_b$, then

$$u_a \leq \frac{1}{|K|} \int_K u \leq u_b,$$

and we conclude that $\Pi^\perp(u)_{|K} = \frac{1}{|K|} \int_K u$, for all $K \in \mathcal{P}$. Besides if we recall the following optimal Poincaré inequality (see [2]):

$$\|j - \Pi^\perp(j)\|_{L^2(K)} \leq \frac{h_K}{\pi} \|\nabla(j)\|_{L^2(K)} \quad \forall\, j \in H^1(K). \tag{4.8}$$

If we take $\bar{u} = u_h$ and $\bar{u}_h = \Pi^\perp(u)$ in the variational inequalities (4.5), we have that

$$(p + \lambda u, u_h - u)_{L^2(\Omega)} \geq 0, \tag{4.9}$$

$$(p_h + \lambda u_h, \Pi^\perp(u) - u + u - u_h)_{L^2(\Omega)} \geq 0. \tag{4.10}$$

Adding this optimal inequalities we note that

$$\lambda \|u - u_h\|^2_{L^2(\Omega)} \leq (p - p_h, u_h - u)_{L^2(\Omega)} + (p_h + \lambda u_h, \Pi^\perp(u) - u)_{L^2(\Omega)}. \tag{4.11}$$

Now we will bound the second term of the right hand side of the inequality above. Considering the definition of the orthogonal projector and using that $u_{h|K} \in \mathbb{P}_0(K)$

and $p_h \in L^2(\Omega)$, for all $K \in \mathcal{P}$, it follows that

$$\begin{aligned}
(p_h + \lambda u_h, \Pi^{\perp}(u) - u)_{L^2(\Omega)} &= \sum_{K\in\mathcal{P}} \left((p_h, \Pi^{\perp}(u) - u)_{L^2(K)} + \lambda \underbrace{(u_h, \Pi^{\perp}(u) - u)_{L^2(K)}}_{=0} \right) \\
&= \sum_{K\in\mathcal{P}} (p_h, \Pi^{\perp}(u) - u)_{L^2(K)} \\
&= \sum_{K\in\mathcal{P}} (p_h - \Pi^{\perp}(p_h), \Pi^{\perp}(u) - u)_{L^2(K)} \\
&\leq \sum_{K\in\mathcal{P}} \left(\frac{h_K}{\pi}\right)^2 \|\nabla(p_h)\|_{L^2(K)} \|\nabla(u)\|_{L^2(K)}, \\
&\leq \max_{K\in\mathcal{P}} \{h_K^2 \pi^{-2}\} \left(\sum_{K\in\mathcal{P}} \|\nabla(p_h)\|^2_{L^2(K)} \right)^{1/2} \left(\sum_{K\in\mathcal{P}} \|\nabla(u)\|^2_{L^2(K)} \right)^{1/2} \\
&\leq C\, h^2 \|\nabla(p_h)\|_{L^2(\Omega)} \|\nabla(u)\|_{L^2(\Omega)},
\end{aligned}$$

where we have used the fact that the optimal control u hat $H^1(\Omega)$ regularity (see Lemma 1.1 in [6]). Taking now $w_h = p_h$ in (4.5), and using the coercivity of B^*, we have that

$$\begin{aligned}
\varepsilon \|\nabla(p_h)\|^2_{L^2(\Omega)} &\leq B^*(p_h, p_h) \\
&= (y_h - y_\Omega, p_h)_{L^2(\Omega)} \\
&\leq \left(\|y_h\|_{L^2(\Omega)} + \|y_\Omega\|_{L^2(\Omega)} \right) \|p_h\|_{L^2(\Omega)} \\
&\leq \frac{c_P}{\sqrt{\varepsilon}} \left(\frac{1}{\sqrt{\kappa}} \sqrt{\kappa} \|y_h\|_{L^2(\Omega)} + \|y_\Omega\|_{L^2(\Omega)} \right) \sqrt{\varepsilon} \|\nabla(p_h)\|_{L^2(\Omega)},
\end{aligned}$$

where we used the Cauchy–Schwarz inequality. Hence

$$\sqrt{\varepsilon} \|\nabla(p_h)\|_{L^2(\Omega)} \leq \frac{c_P}{\sqrt{\varepsilon}} \left(\frac{1}{\sqrt{\kappa}} \sqrt{\kappa} \|y_h\|_{L^2(\Omega)} + \|y_\Omega\|_{L^2(\Omega)} \right).$$

In the same way, we have

$$\kappa \|y_h\|^2_{L^2(\Omega)} \leq B(y_h, y_h) = (g + u_h, y_h)_{L^2(\Omega)} \leq \frac{1}{\sqrt{\kappa}} \left(\|g\|_{L^2(\Omega)} + \|u_h\|_{L^2(\Omega)} \right) \sqrt{\kappa} \|y_h\|_{L^2(\Omega)}.$$

Then,

$$\sqrt{\kappa} \|y_h\|_{L^2(\Omega)} \leq \frac{1}{\sqrt{\kappa}} \left(\|g\|_{L^2(\Omega)} + \|u_h\|_{L^2(\Omega)} \right),$$

which allow us to obtain

$$\sqrt{\varepsilon} \|\nabla(p_h)\|_{L^2(\Omega)} \leq \frac{c_P}{\sqrt{\varepsilon}} \left(\frac{1}{\kappa} \left(\|g\|_{L^2(\Omega)} + \|u_h\|_{L^2(\Omega)} \right) + \|y_\Omega\|_{L^2(\Omega)} \right),$$

and finally, as $u_h \in U^h_{ad}$, we have that $u_a \leq u_h \leq u_b$, which leads us to conclude

$$\sqrt{\varepsilon} \|\nabla(p_h)\|_{L^2(\Omega)} \leq C \left(\|g\|_{L^2(\Omega)} + \|y_\Omega\|_{L^2(\Omega)} + \max\{|u_a|, |u_b|\} \right).$$

This allow us to bound (4.11), as

$$\lambda\|u-u_h\|^2_{L^2(\Omega)} \leq (p-p_h, u_h-u)_{L^2(\Omega)} + Ch^2. \tag{4.12}$$

Now, we add the function q_h to the first term in the previous right hand side, defined in (4.7), to obtain

$$(p-p_h, u_h-u) \;=\; (p-q_h, u_h-u) + (q_h-p_h, u_h-u).$$

Recalling that $p \in H^1_0(\Omega)$ and $q_h \in V_h$, are solution of the following problems

$$\begin{aligned} B^*(p,v) &= (y(u)-y_\Omega, v) \quad \forall\ v \in H^1_0(\Omega), \\ B^*(q_h, v_h) &= (y(u)-y_\Omega, v_h) \quad \forall\ v_h \in V_h, \end{aligned}$$

respectively. This tell us that q_h corresponds to a discretisation of finite elements for p, and from the classic theory of FEM (see for example Theorem 3.16 in [9]), we have that

$$\|p-q_h\|_\Omega \leq Ch.$$

For the second term we repeat slightly the procedure in the next fashion, with r_h defined in (4.6),

$$(q_h-p_h, u_h-u) = (q_h-r_h, u_h-u) + (r_h-p_h, u_h-u).$$

This time, using the discrete and auxiliar problems, making explicit $y_h = y_h(u_h)$, we have that

$$\begin{array}{rcl} B(y_h(u_h), v_h) & = & (g+u_h, v_h) \\ B(y_h(u), v_h) & = & (g+u, v_h) \end{array} \Rightarrow B(y_h(u_h)-y_h(u), v_h) = (u_h-u, v_h)$$

If we choose $v_h = r_h - p_h \in V_h$, then $B(y_h(u_h)-y_h(u), r_h-p_h) = (u_h-u, r_h-p_h) = (r_h-p_h, u_h-u)$ because of the symmetry of the inner product in $L^2(\Omega)$. Now, using the dual problem we find that

$$\begin{aligned} (r_h-p_h, u_h-u) &= B^*(r_h-p_h, y_h(u_h)-y_h(u)) \\ &= (y_h(u)-y_h(u_h), y_h(u_h)-y_h(u)) \\ &= -\|y_h(u)-y_h(u_h)\|^2_{L^2(\Omega)} \leq 0. \end{aligned} \tag{4.13}$$

On the other hand, by Cauchy-Schwarz and Poincaré inequalities we have that

$$(q_h-r_h, u_h-u) \leq \|q_h-r_h\|_{L^2(\Omega)}\|u_h-u\|_{L^2(\Omega)} \leq \frac{1}{\sqrt{\kappa}}\|q_h-r_h\|_\Omega\|u_h-u\|_{L^2(\Omega)}.$$

Now, if we use the coercivity of B^*, it follows that

$$\begin{aligned} \|q_h-r_h\|^2_\Omega &\leq B^*(q_h-r_h, q_h-r_h) = (y(u)-y_h(u), q_h-r_h)_{L^2(\Omega)} \\ &\leq \|y(u)-y_h(u)\|_{L^2(\Omega)}\|q_h-r_h\|_{L^2(\Omega)} \leq \frac{1}{\kappa}\|y(u)-y_h(u)\|_\Omega\|q_h-r_h\|_\Omega, \end{aligned}$$

and noting that $y_h(u)$ is a finite element discretisation for $y(u)$ which contains the same right hand side allows us to use similar arguments as before to conclude that $\|y(u) - y_h(u)\|_\Omega \leq Ch$. In resume, we have that

$$\|q_h - r_h\|_\Omega \leq Ch.$$

Using again Cauchy–Schwarz inequality, $ab \leq \frac{1}{2\lambda}a^2 + \frac{\lambda}{2}b^2$ and previous results we obtain

$$\begin{aligned}\lambda\|u - u_h\|^2_{L^2(\Omega)} &\leq C\left(\|p - q_h\|_\Omega + \|q_h - r_h\|_\Omega\right)\|u - u_h\|_{L^2(\Omega)} + \underbrace{(r_h - p_h, u_h - u)_{L^2(\Omega)}}_{\leq 0} \\ &\leq C\left(\|p - q_h\|^2_\Omega + \|q_h - r_h\|^2_\Omega\right) + \frac{\lambda}{2}\|u - u_h\|^2_{L^2(\Omega)} \\ &\leq C\,h^2 + \frac{\lambda}{2}\|u - u_h\|^2_{L^2(\Omega)}.\end{aligned}$$

With this, we have obtained the following:

THEOREM **4.1 Control Convergence.** *Let (y, p, u) be the solution of the weak problem* (4.1) *and (y_h, p_h, u_h) the discrete FEM solution of* (4.4)*. Then, under the assumptions of Lemma 1.1 in [6] and Theorem 3.16 in [9], there exist a constant C, which only depends on Ω and the physical parameters such that*

$$\lambda\|u - u_h\|_{\mathrm{L}^2(\Omega)} \leq C\,h. \tag{4.14}$$

Now, to obtain the convergence for the state and its adjoint, we notice that using the coercivity of B and substracting the continuous and discrete state equations in (4.5) and then using Cauchy–Schwarz that

$$\begin{aligned}\|y - y_h\|^2_\Omega &\leq B(y - y_h, y - y_h) = (u - u_h, y - y_h)_{L^2(\Omega)} \\ &\leq \frac{1}{\sqrt{\kappa}}\|u - u_h\|_{L^2(\Omega)}\sqrt{\kappa}\|y - y_h\|_{L^2(\Omega)} \\ &\leq \frac{1}{\sqrt{\kappa}}\|u - u_h\|_{L^2(\Omega)}\|y - y_h\|_\Omega, \\ \Rightarrow \|y - y_h\|_\Omega &\leq \frac{1}{\sqrt{\kappa}}\|u - u_h\|_{L^2(\Omega)}.\end{aligned}$$

In the same way, we conclude that

$$\begin{aligned}\|p - p_h\|^2_\Omega &\leq B^*(p - p_h, p - p_h) = (y - y_h, p - p_h)_{L^2(\Omega)} \\ &\leq \frac{1}{\kappa}\sqrt{\kappa}\|y - y_h\|_{L^2(\Omega)}\sqrt{\kappa}\|p - p_h\|_{L^2(\Omega)} \leq \frac{1}{\kappa}\|y - y_h\|_\Omega\|p - p_h\|_\Omega \\ &\leq \frac{1}{\kappa^{3/2}}\|u - u_h\|_{L^2(\Omega)}\|p - p_h\|_\Omega, \\ \Rightarrow \|p - p_h\|_\Omega &\leq \frac{1}{\kappa^{3/2}}\|u - u_h\|_{L^2(\Omega)}.\end{aligned}$$

Summarizing we have,

THEOREM **4.2 State and Adjoint Convergence.** *Let (y, p, u) the solution of the weak problem* (4.1) *and (y_h, p_h, u_h) the FEM solution of* (4.4)*. Then, under the assumptions of Lemma 1.1 in [6] and Theorem 3.16 in [9], there exists constants C and $\tilde{C}$, which only depend on Ω and the physical parameters such that*

$$\|p - p_h\|_\Omega \leq C\,h \quad \textit{and} \quad \|y - y_h\|_\Omega \leq \tilde{C}\,h. \tag{4.15}$$

5. Primal-Dual Active Set (PDAS) Strategy to Solve Optimality System

There exist several strategies to obtain the solution of our Control Problem (see for example [14], Chapter 2). In this section we present a pseudo Newton method ([14]) which allows the application to a wide number of quadratic minimisation problems, as the one we discuss. These methods begin with a feasible solution and obtain a descent sequence where each point in the sequence is feasible.
This type of methods are an invitation besides to generalize into the solution of control problems through Lagrange multipliers. This treatment is not discuss in the present work, but it can be found for example in [15]. We develop first the method for the continuous case and then we implement it in the discrete case. We have already garanteed the convergency of the finite element method presented in the previous section.

5.1. PDAS Strategy in Continuous Problem

The Optimal Control problem under consideration is minimize the Reduce functional 2.12, subject to the Box condition 2.8 and the ADR boundary value problem 2.7. According to the projection relation (3.17), the optimal control u satisfies

$$\bar{u}(x) = \Pi^{\perp}_{[u_a,u_b]}\left(-\lambda^{-1}p(x)\right) = \min\{u_b, \max\{u_a, -\lambda^{-1}p(x)\}\},$$

where the adjoint state p is solution of (3.9). Therefore, defining

$$\mu = -(-\lambda^{-1}p + \bar{u}) = -\lambda^{-1}f'(\bar{u}),$$

by Lemma 3.2, (3.5) and (3.10). We have then that (3.13) takes the form

$$\bar{u}(x) = \begin{cases} u_a & \text{if} \quad -\lambda^{-1}p(x) < u_a \quad (\Leftrightarrow \mu(x) < 0) \\ -\lambda^{-1}p(x) & \text{if} \quad -\lambda^{-1}p(x) \in [u_a, u_b] \quad (\Leftrightarrow \mu(x) = 0) \\ u_b & \text{if} \quad -\lambda^{-1}p(x) < u_b \quad (\Leftrightarrow \mu(x) > 0). \end{cases} \tag{5.1}$$

In the first case, $\mu(x) < 0$ and thus $\bar{u}(x) + \mu(x) < u_a$, because the definition of μ and the fact that $\bar{u} = u_a$. The same argument could be use in the third case. In the second case, $\mu(x) = 0$, hence $\bar{u}(x) + \mu(x) = -\lambda^{-1}p(x) \in [u_a, u_b]$. In summary, if we make $u = \bar{u}$ the following relations are fulfilled

$$u(x) = \begin{cases} u_a & \text{if} \quad u(x) + \mu(x) < u_a \\ -\lambda^{-1}p(x) & \text{if} \quad u(x) + \mu(x) \in [u_a, u_b] \\ u_b & \text{if} \quad u(x) + \mu(x) < u_b. \end{cases} \tag{5.2}$$

Reciprocally, if $u \in U_{ad}$ fulfills the previous relation, then it satisfies the projection condition and then it is optimal. For example, let's consider $u = u_a$, from where $\mu(x) < 0$ and then

$$0 > -\lambda^{-1}p - u = -\lambda^{-1}p - u_a \Rightarrow -\lambda^{-1}p < u_a,$$

and thus $u(x) = \Pi^{\perp}_{[u_a,u_b]}(-\lambda^{-1}p(x))$. We can repeat the procedure for the other two cases.
We can conclude that $u + \mu$ indicates when the inequality restrictions are active or not. This motivates the *primal-dual active set strategy*, which is described now.
The optimality system is: find $(y, p, u) \in H^1_0(\Omega) \times H^1_0(\Omega) \times U_{ad}$ such that y, p solve 2.7, 3.9 respectively and

$$u = \Pi^{\perp}_{[u_a,u_b]}\left(-\lambda^{-1}p(x)\right) = \begin{cases} u_a & \text{if} \quad u(x) + \mu(x) < u_a \\ -\lambda^{-1}p(x) & \text{if} \quad u(x) + \mu(x) \in [u_a, u_b] \\ u_b & \text{if} \quad u(x) + \mu(x) < u_b, \end{cases} \tag{5.3}$$

with $\mu(x) = -(-\lambda^{-1}p(x) + \bar{u}(x))$. Now, we rewrite the last projection, using the following characteristic functions:

$$\chi^a(x) := \begin{cases} 1 & \text{if } x : u(x) + \mu(x) < u_a, \\ 0 & \text{otherwise,} \end{cases} \qquad \chi^b(x) := \begin{cases} 1 & \text{if } x : u(x) + \mu(x) > u_b, \\ 0 & \text{otherwise,} \end{cases} \tag{5.4}$$

which allows to write the projection as

$$(1 - \chi^a(x) - \chi^b(x))\lambda^{-1}p(x) + u(x) = \chi^a(x)u_a + \chi^b(x)u_b. \tag{5.5}$$

This relation should clearly understand in the weak form. Then, in what follows we will be concerned with the following, and final, optimality system:

$$\boxed{\begin{aligned} -\varepsilon\Delta y + \underline{b}\cdot\nabla y + \kappa y &= u + g, \quad \text{in } \Omega \\ -\varepsilon\Delta p - \underline{b}\cdot\nabla p + \kappa p &= y - y_\Omega, \quad \text{in } \Omega, \\ (1 - \chi^a(x) - \chi^b(x))\lambda^{-1}p(x) + u(x) &= \chi^a(x)u_a + \chi^b(x)u_b. \end{aligned}} \tag{5.6}$$

5.2. Mixed formulation of Optimality System

In order to simplify the computational implementation of our approach we will consider a mixed formulation of the latter system 5.6. To this end, we will use test functions $(v, q, w) \in H^1_0(\Omega) \times H^1_0(\Omega) \times L^2(\Omega)$. As usual, we multiply each equation by v, q and w respectively and then integrate on Ω to obtain

$$\begin{aligned} -\varepsilon(\nabla y, \nabla v) + (\underline{b}\cdot\nabla y, v) + \kappa(y, v) - (u, v) &= (g, v), \\ -\varepsilon(\nabla p, \nabla q) - (\underline{b}\cdot\nabla p, q) + \kappa(p, q) - (y, q) &= -(y_\Omega, q), \\ (1 - \chi^a(x) - \chi^b(x))\lambda^{-1}(p, w) + (u, w) &= (\chi^a(x)u_a + \chi^b(x)u_b, w). \end{aligned} \tag{5.7}$$

This can be simplified using the bilinear forms B and B^* defined in section 4.1 and defining the bilinear form $e(u, v) = -(u, v)$, simply the L^2 product of functions u, v. Also we define

$$\begin{aligned} \beta &= (1 - \chi^a(x) - \chi^b(x)), \\ \eta &= \chi^a(x)u_a + \chi^b(x)u_b, \end{aligned}$$

Then

$$\begin{aligned} B(y,v) + e(u,v) &= (g,v), \\ e(y,q) + B^*(p,q) &= -(y_\Omega, q), \\ -e(\lambda^{-1}\beta p + u, w) - e(u,w) &= (\eta, w). \end{aligned} \tag{5.8}$$

To state the mixed formulation we will add the three left hand sides of the previous system to define

$$\begin{aligned} a\left((y,p,u);(v,q,w)\right) = &\left(B(y,v) + e(u,v)\right) + \left(e(y,q) + B^*(p,q)\right) \\ &- \left(e(\lambda^{-1}\beta p + u, w) + e(u,w)\right). \end{aligned} \tag{5.9}$$

In the same way we define the sum of each right hand side in 5.6:

$$R\left((g, y_\Omega, \eta);(v,q,w)\right) = (g,v) + (y_\Omega, q) + (\eta, w). \tag{5.10}$$

Then, our problem is formulated as follows: Find a triple $(y,p,u) \in H_0^1(\Omega) \times H_0^1(\Omega) \times L^2(\Omega)$ such that, for any test functions $(v,q,w) \in H_0^1(\Omega) \times H_0^1(\Omega) \times L^2(\Omega)$:

$$a\left((y,p,u);(v,q,w)\right) = R\left((g, y_\Omega, \eta);(v,q,w)\right) \tag{5.11}$$

We need to show that this formulation preserves the convergence orders obtained in theorems 4.1 and 4.2. To see this, we note that coercivity and continuity of a are an inmediate consequence of theorem 2.1 applied to B and B^* and the properties of the inner product in $L^2(\Omega)$. To recover the convergence orders for each variable, we refer to the article of Leykekhman [17] and the Lecture Notes on Stokes System from the Numerics of EDP 2 curse from Lars Diening [8]. In the same direction, we recommend [23] or [12] for an introductory approach.
With all these elements, we can formulate the following algorithm to solve our OCP:

Algorithm 1 PDAS algorithm for ADR control problem, mixed formulation

Input: Initial guess u_0, μ_0, max. iteration number K
Set $k = 0$
Determine the indicator functions χ_0^a, χ_0^b of the active sets

$$\begin{aligned}\mathcal{A}_0^a =& \{x \in \Omega : \mu_0(x) + u_0(x) < u_a\} \\ \mathcal{A}_0^b =& \{x \in \Omega : \mu_0(x) + u_0(x) > u_b\}.\end{aligned}$$

while $1 \leq k \leq K$ **do**
  Define

$$\begin{aligned}\beta_{k-1} =& (1 - \chi_{k-1}^a - \chi_{k-1}^b) \\ \eta_{k-1} =& u_a \chi_{k-1}^a + u_b \chi_{k-1}^b\end{aligned}$$

  Solve Find $y_k, p_k \in H_0^1(\Omega)$ and $u_k \in L^2(\Omega)$ such that

$$a\left((y_k, p_k, u_k); (v, q, w)\right) = R\left((g, y_\Omega, \eta_{k-1}); (v, q, w)\right), \tag{5.12}$$

  for test functions $v, q \in H_0^1(\Omega)$ and $w \in L^2(\Omega)$.
  Actualize $\mu_k = p_k - \lambda u_k$, χ_k^a, χ_k^b.
  if $\|\chi_k^a - \chi_{k-1}^a\| + \|\chi_k^b - \chi_{k-1}^b\| = 0$: **then** ▷ This garanties that $u_k \in U_{ad}$.
    Break
  else
    Set $k = k + 1$
  end if
end while
end

6. Numerical Results

In this section we develop numerical experiments for the ADR Optimal Control Problem 5.6. Our goal is to check the convergence orders for control, state and adjoint state given by theorems 4.1 and 4.2, as well as to establish a comparison between our method and the novel optimal control software `cashocs` [3]. First we show the parameters and exact solutions, then we provide the convergence analysis for our approach to conclude with a similar analysis using `cashocs` to determine a comparison between the two techniques. We will see that convergence orders using the mixed strategy are in concordance with the convergence results presented in section 4.2, but are strangely improved with the use of `cashocs`. We give a conjecture for this issue in section 5.3.

6.1. Parameters and exact solutions

For the parameters and data showed in Table 6.1 we set the desire state y_Ω and the external function g as

Parameter	Value
ε	1.0
$\underline{\mathrm{b}}$	$(3, 4)$
κ	1
λ	1
u_a	-2
u_b	2

Table 6.1: Parameters for example problem 5.6 in domain $\Omega = [0, 1]^2$.

$$
\begin{aligned}
y_\Omega &= -x_1^2x_2^2 + 9x_1^2x_2 - 2x_1^2 + 7x_1x_2^2 - 15x_1x_2 + 2x_1 - x_2^2 + x_2 \\
g &= -6x + 6x_1^2 - 5x_2 + 12x_1x_2 - 6x_1^2x_2 + 5x_2^2 - 4x_1x_2^2 - 2x_1^2x_2^2 \\
&\quad + \min\left(-2, \max\left(-2, -(1-x_1)x_1(1-x_2)x_2\right)\right)
\end{aligned}
$$

We consider the exact solution triple, with the control given by 3.17

$$
\begin{aligned}
y &= xy(x-1)(y-1) \\
p &= xy(x-1)(y-1), \\
u &= \min(u_b, \max(u_a, -p(x)/\lambda)).
\end{aligned}
$$

For both implementations we have chosen a tolerance $\texttt{tol} = 10^{-9}$.

6.2. Implementation in `FENICS`

We implementate the algorithm 1 in the `Python` based dedicated softwares `FENICS` [1], [20] and `Dolfin` [21]. The main advantage of these softwares relies on the hybrid resolution scheme they applied. We just need to input the domain, mesh, finite element spaces, parameters, weak formulation and boundary conditions and the software will internally resolve and return the numerical solutions. The programming was divided in four parts: a main program containing the different meshes, parameters and the PDAS

loop and four subroutines containing a solver for the mixed weak formulation 5.12, the $L^2(\Omega)$ projection implementation needed by equation 5.5 and the indicator functions χ_k^a, χ_k^b respectively. At the end of the main program, we develop the postprocessing analysis in order to provide figures of the numerical solutions as well as convergence rates of our algorithm. Details can be found in the Appendix.

6.2.1. Numerical Results

We have consider 5 different meshes with 16, 32, 64, 128 and 256 elements. Like in section 4, we choose piecewise Lagrange discontinuos triangular elements for the control function u and linear continuous Lagrange triangular elements for state and adjoint state y, p respectively. A two dimensional representation of the numerical solutions is showed in Figure 6.1

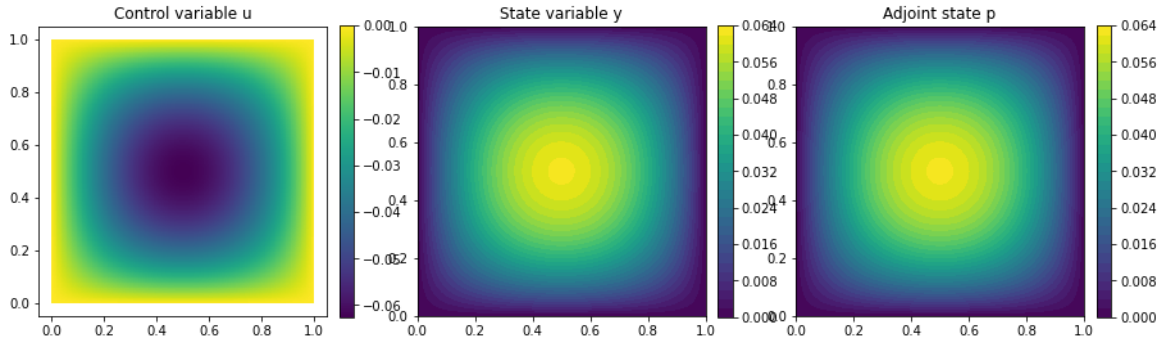

Figure 6.1: Numerical solutions for control (left), state (middle) and adjoint state for 5.12 problem and parameters given by Table 6.1. We have used 256 elements for this representation.

Next, we consider the convergence rates of our approximation technique. The numerical results essentially confirm the theoretical estimate of Theorems 4.1 and 4.2, with a convergence rate of h^1 observed for the L^2-norm for control and H^1-norm for both state and adjoint state. Another important point corresponds to the mesh-independance of the iterations. This is concordance with the literature [13]. This means that for each mesh, our method requires the same number of iterations to find the optimal solution. The low number of iterations, just 2 for each mesh, is probably because of the relative simplicity of our problem test selection and the type of our state and adjoint state equations, since both are linear PDE's. The election of $\underline{b}$ could be also relevant (see, for example [17]). The results are showed in Table 6.2 and Figure 6.2. The adjusted convergence rates were made with relative coarse meshes (10, 20, 30, 40, 50, 60 elements), but they succeed to describe correctly the behaviour of each error.

NDOF	$h_{\max}$	Iterations	$\|u - u_h\|_{L^2(\Omega)}$	$\|y - y_h\|_{H^1(\Omega)}$	$\|p - p_h\|_{H^1(\Omega)}$
16	3.125×10^{-2}	2	$2.231\,98 \times 10^{-3}$	$1.519\,16 \times 10^{-2}$	$1.519\,46 \times 10^{-2}$
32	1.5625×10^{-2}	2	$1.102\,57 \times 10^{-3}$	7.6044×10^{-3}	$7.604\,78 \times 10^{-3}$
64	7.8125×10^{-3}	2	$5.495\,79 \times 10^{-4}$	$3.803\,27 \times 10^{-3}$	$3.803\,32 \times 10^{-3}$
128	$3.906\,25 \times 10^{-3}$	2	$2.745\,75 \times 10^{-4}$	$1.901\,77 \times 10^{-3}$	$1.901\,78 \times 10^{-3}$
256	$1.953\,12 \times 10^{-3}$	2	$1.372\,61 \times 10^{-4}$	$9.509\,02 \times 10^{-4}$	$9.509\,02 \times 10^{-4}$

Table 6.2: Error approximations for test problem 5.6 in domain $\Omega = [0,1]^2$. Although we showed that norms $\|\cdot\|_\Omega$ and $\|\cdot\|_{H^1(\Omega)}$ are equivalent in Section 1, the election of the parameters sets this equivalence as an equality.

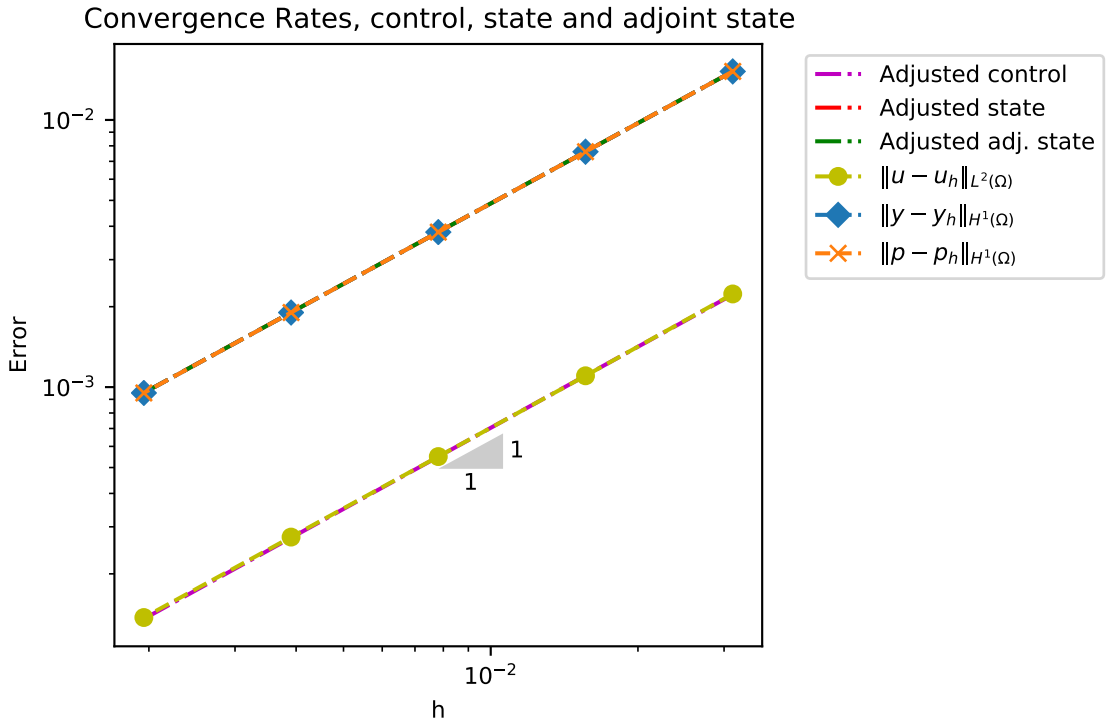

Figure 6.2: Convergence Rates for control u in $L^2(\Omega)$ error, state y and adjoint state p in $H^1(\Omega)$ error. The adjusted convergence rates were with least squares method calculated and are 1.008 38 for control variable and 9.9923×10^{-1}, $9.993\,15 \times 10^{-1}$ for state and adjoint state variables respectively.

6.3. Implementation in `cashocs`

We repeat our analysis for the problem under consideration using this time the dedicated optimal control software `cashocs` [3]. This software uses the framework given by `FENICS` and `Dolfin` to implementate its internal solvers. We have chosen the PDAS Strategy as solver in order to compare with our results. All the settings are the same as in previous section with a single difference: by sotfware restriction, we must choose the same finite element space for the three variables. That is, linear continuous Lagrange triangular elements for control, state and adjoint state. This extra regularity in the election of the finite element space for the control variable has implied an extra regularity for the numerical control solution as we can see in Figure 6.4. In this case

we can conjecture that the optimal control lies in a subset of $U_{ad} \subset L^2(\Omega)$. On the other hand, convergence rates for both state and adjoint state were preserved and were not afected by the election of the finite element space for the control variable. We can see this behaviour in Figure 6.4.

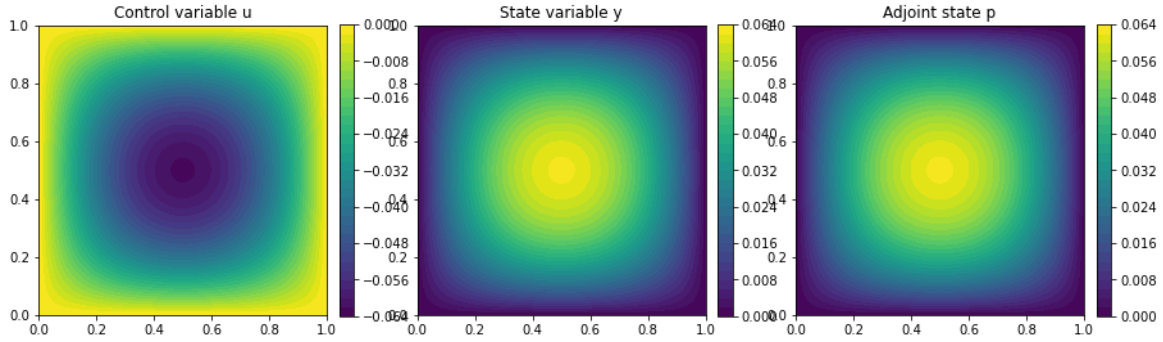

Figure 6.3: Numerical solutions for control (left), state (middle) and adjoint state for 5.12 problem and parameters given by Table 6.1 using `cashocs`. It is important to remark that the scheme to solve the adjoint equation used by this software follows the approach of [14] and therefore will solve the optimality system for $-p$. We have used 256 elements for this representation.

NDOF	$h_{\max}$	Iterations	$\|u - u_h\|_{L^2(\Omega)}$	$\|y - y_h\|_{H^1(\Omega)}$	$\|p - p_h\|_{H^1(\Omega)}$
16	3.125×10^{-2}	2	$4.388\,45 \times 10^{-4}$	$1.519\,37 \times 10^{-2}$	$1.519\,59 \times 10^{-2}$
32	1.5625×10^{-2}	2	$1.100\,54 \times 10^{-4}$	$7.604\,66 \times 10^{-3}$	$7.604\,94 \times 10^{-3}$
64	7.8125×10^{-3}	2	2.7535×10^{-5}	3.8033×10^{-3}	$3.803\,34 \times 10^{-3}$
128	$3.906\,25 \times 10^{-3}$	2	$6.885\,08 \times 10^{-6}$	$1.901\,77 \times 10^{-3}$	$1.901\,78 \times 10^{-3}$
256	$1.953\,12 \times 10^{-3}$	2	$1.721\,35 \times 10^{-6}$	$9.509\,02 \times 10^{-4}$	$9.509\,03 \times 10^{-4}$

Table 6.3: Error approximations for test problem 5.6 in domain $\Omega = [0, 1]^2$ using `cashocs` PDAS solver. As we can see, the error $L^2(\Omega)$ is less than the approximation obtained in our algorithm, but is almost equal for the $H^1(\Omega)$ error for state and adjoint variables.

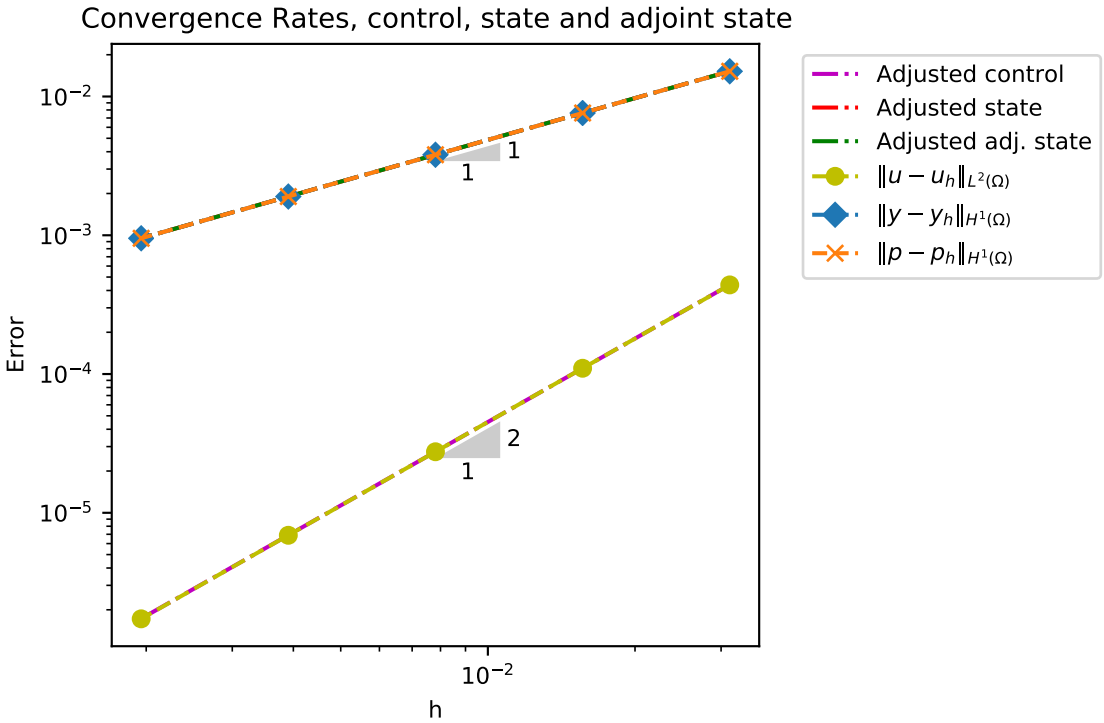

Figure 6.4: Convergence Rates for control u in $L^2(\Omega)$ error, state y and adjoint state p in $H^1(\Omega)$ error. The adjusted convergence rates were with least squares method calculated and are 1.997 87 for control variable and 9.993×10^{-1}, $9.993\,75 \times 10^{-1}$ for state and adjoint state variables respectively.

7. Conclusions and Outlook

Summary of the work

Following the guideline given by the literature [24], [18], we have managed to prove the existence and uniqueness of the optimal state-control pair. In addition, the active primal-dual set strategy has been successfully implemented using a mixed formulation approach of the optimal system. Numerical results of our approach coincide with the theory proposed and demonstrated in this work.
The fact of using L^2-Projection in the implementation to coincide with the theory shows good results, since the convergence orders stipulated by the theoretical framework are respected and the approximation errors are kept under control. On the other hand, the results provided by `cashocs`, despite having a better order of convergence for the control, do not respect the theory and must be rigorously examined. Interesting is the fact that the convergence orders for state and adjoint state remain as suggested by our theoretical framework.

Important Results

Without a doubt, after providing the theoretical framework of the work in Sections 2 and 3, results for the approximation have been obtained through the finite element method that coincide with the related literature [17]. When we have our results proved, we did not have such reference, so the fourth section constitutes the soul of our work. The numerical results only confirm the correctness of our analysis.
The mixed formulation of the optimal system has the advantage of a notorious simplification of the programming task and it allows us to parallelize the routines and minimise the computational calculations.

State of Research and Outlook

Notwithstanding the foregoing, the equivalence between the optimal system and the mixed formulation remains to be shown more clearly. Especially what concerns the projection relationship between adjoint state and control 5.5. This is one of the points where to persevere in our investigation.
The programs developed allow to consider different solvers for other control problems. Remains to study the limitations of this approach and obviously to look the perspective for further applications. For now, we have already found a second test problem in section 5.1 of [17] which seems to offer more difficulties than the one studied in our work. It will be interesting to compare both schemes in the near future.

References

[1] Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

[2] M. Bebendorf. A note on the Poincaré inequality for convex domains. *Z. Anal. Anwendungen*, 22(4):751–756, 2003.

[3] Sebastian Blauth. cashocs: A computational, adjoint-based shape optimization and optimal control software. *SoftwareX*, 13:100646, Jan 2021.

[4] Susanne C. Brenner and L. Ridgway Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008.

[5] Eduardo Casas. Optimal control of pde theory and numerical analysis. page 136, Aug 2006. Lecture.

[6] Eduardo Casas and Fredi Tröltzsch. Error estimates for linear-quadratic elliptic control problems. In *Analysis and optimization of differential systems (Constanta, 2002)*, pages 89–100. Kluwer Acad. Publ., Boston, MA, 2003.

[7] Juan Carlos de los Reyes and Fredi Tröltzsch. Optimal control of the stationary navier–stokes equations with mixed control-state constraints. *SIAM J. Control. Optim.*, 46(2):604–629, 2007.

[8] Lars Diening. *Numerics of PDE II*. Universität Bielefeld, 2020.

[9] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements*, volume 159 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2004.

[10] Funke, S.W. and Nordass, M. PDE constrained optimisation in Hilbert Spaces. `https://fenicsproject.org/pub/presentations/fenics14-paris/SimonFunke.pdf`, 2014.

[11] Carlos Galeano Urueña. Técnicas de solución numérica de la ecuación difusión-advección-reacción para el estudio de dispersión de contaminantes. Master's thesis, 2009.

[12] Gabriel Gatica. *A Simple Introduction to the Mixed Finite Element Method. Theory and Applications*. 01 2014.

[13] Ulbrich M. Hintermüller, M. A mesh-independence result for semismooth newton methods. *Mathematical Programming*, 101:151–184, 08 2004.

[14] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE constraints*, volume 23 of *Mathematical Modelling: Theory and Applications*. Springer, New York, 2009.

[15] Kazufumi Ito and Karl Kunisch. *Lagrange multiplier approach to variational problems and applications*, volume 15 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.

[16] Günter Leugering, Peter Benner, Sebastian Engell, Andreas Griewank, Helmut Harbrecht, Michael Hinze, Rolf Rannacher, and Stefan Ulbrich. *Trends in PDE Constrained Optimization*, volume 165. 01 2014.

[17] Dmitriy Leykekhman. Investigation of commutative properties of discontinuous galerkin methods in pde constrained optimal control problems. *J. Sci. Comput.*, 53(3):483–511, 12 2012.

[18] J.-L. Lions. *Optimal control of systems governed by partial differential equations*. Die Grundlehren der mathematischen Wissenschaften, Band 170. Springer-Verlag, New York-Berlin, 1971. Translated from the French by S. K. Mitter.

[19] W. Liu, Pekka Neittaanmäki, and Dan Tiba. Existence for shape optimization problems in arbitrary dimension. *SIAM J. Control and Optimization*, 41:1440–1454, 01 2002.

[20] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012.

[21] Anders Logg and Garth N. Wells. Dolfin: Automated finite element computing. *ACM Trans. Math. Softw.*, 37(2), April 2010.

[22] Pekka Neittaanmaki and Sergey Repin. Two-sided guaranteed estimates of the cost functional for optimal control problems with elliptic state equations. *Lecture Notes in Computational Science and Engineering*, 101, 2014.

[23] Endre Süli. A brief excursion into the mathematical theory of mixed finite element methods.

[24] Fredi Tröltzsch. *Optimal control of partial differential equations*, volume 112 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2010. Theory, methods and applications, Translated from the 2005 German original by Jürgen Sprekels.

[25] Pieter Wesseling. *The stationary convection-diffusion equation*, pages 111–161. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

Appendix A Implementation in FENICS

Here the interested reader can find the routines used to implementate the PDAS to the ADR optimal Control Problem in FENICS

A.1 Main Program

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu Jul 15 19:47:55 2021

\@author: Juan Carlos Chavarr\'ia

This script will return the solution of an Optimal Control Problem
with restrictions given by an PDE and a box condition using the PDAS.
The script is general. The user provide the PDE, used as an input (solver),
and run this routine to obtain the solution as well as the desired state used
to compute de values of the objective functional J.

-- In this case we consider the ADR equation studied in the Bachellorarbeit
"""
from fenics import *
from math import sqrt, inf
from dolfin import *
from SolverADR import *                         # Solver of the ADR eq.
from project_up import *                        # Indicatrix function for X_a
from project_low import *                       # The same for X_b
from L2projection import *                      # L2 projector
import numpy as np
import matplotlib.pyplot as plt
from plotslopes import *
#from mpltools import annotation
# Set some algorithmic parameters
atol = 1e-9                                     # Tolerance
maxiter = 100                                   # Max. number of iterations
# Set the lambda parameter
lam = Constant(1.0)
# Define polynomial degrees for control and state discretization
degree_con = 0
degree_state = 1
# Exact Solutions
ye=Expression('x[0]*x[1]*(1-x[0])*(1-x[1])',degree=4)
pe=Expression('x[0]*x[1]*(1-x[0])*(1-x[1])',degree=4)
ue=Expression('min(2.0, max(-2.0, -((1-x[0])*x[0]*(1-x[1])*x[1])/lam))',lam=lam,degree=3)
#Define Mesh
# number of uniform refinements
Nref = 5
### Store information ###
# a counter
array=np.arange(Nref)
# NDOF of each mesh
l=np.empty(Nref, dtype=int)
# Maximum diameter
h=np.empty(Nref)
# Errors
L2error = np.empty(Nref)
State_error_y = np.empty(Nref)
Adjointerror_p = np.empty(Nref)
### Loops ###
# exterior loop: set each mesh and calculate the approx. errors
for j in array:
    ## NDOF ##
    l[j]=2**(j+4)
    ## mesh size
    h[j]=np.divide(1,(np.multiply(2,l[j])))
    ## mesh
    mesh = UnitSquareMesh(l[j],l[j])
    ##### Spaces ####
    ## Finite Element Spaces
    PU = FiniteElement('DG',mesh.ufl_cell(),degree_con)
    PY = FiniteElement('Lagrange',mesh.ufl_cell(),degree_state)
    ## Function Spaces
    U = FunctionSpace(mesh,PU)
    Y = FunctionSpace(mesh,PY)
    # Desired State
    yom=Expression('(2*x[0] - 2*pow(x[0],2) + x[1] -14*x[0]*x[1] + 8*x[1]*pow(x[0],2) \
                   -pow(x[1],2) + 6*x[0]*pow(x[1],2)) ', degree=degree_state)
    yd=interpolate(yom,Y)
    # RHS state equation
    exf=Expression('(6*x[0]-6*pow(x[0],2)+5*x[1]-13*x[0]*x[1]\
              +7*pow(x[0],2)*x[1]-5*pow(x[1],2)+5*x[0]*pow(x[1],2)\
                  +pow(x[0],2)*pow(x[1],2)\
                      -min(2.0, max(-2.0, -(1-x[0])*x[0]*(1-x[1])*x[1])))'\
                          ,degree=degree_state)
    g=interpolate(exf,Y)
    # Set the bounds of box condition
```

```
    ua = interpolate(Expression('-2.0', degree=degree_con), U)
    ub = interpolate(Expression('2.0', degree=degree_con), U)
    # Prepare for the PDAS loop
    # Set the initial guesses
    u0 = interpolate(Expression('-100.0', degree=degree_con), U)
    mu0 = interpolate(Expression('-100.0', degree=degree_con), U)
    # Create functions holding the iterations
    u = Function(U)
    mu= Function(U)
    toproj=Function(U)
    # # Build the constant 1 to use in beta function
    uno = interpolate(Expression('1.0', degree=degree_con), U)
    # Set the initial iterate
    u.assign(u0)
    mu.assign(mu0)
    toproj.assign(u0+mu0)
    ## Build the initial indicator functions
    Xa_old = project_up(toproj, U, ua)
    Xb_old = project_low(toproj, U, ub)
    # counter & condition
    iter = 0
    done = False
    # inner loop: PDAS
    while (iter < maxiter) and (done == False):
        beta = (uno-Xa_old-Xb_old)/lam
        phi = ua*Xa_old+ub*Xb_old
        u, y, p = SolverADR(mesh, PU, PY, beta, phi, yd, g, lam)
        Ph=L2Projection(mesh, p, degree_con) # We need to project, otherwise mu mix different
            function spaces
        tomu = np.add(np.multiply(1/lam,Ph.vector()[:]), np.multiply(-1, u.vector()[:]))
        mu.vector()[:] = tomu
        coefftoproj = np.add(u.vector()[:], mu.vector()[:])
        toproj.vector()[:] = coefftoproj
        Xa_new = project_up(toproj, U, ua)
        Xb_new = project_low(toproj, U, ub)
        error_a = sqrt(assemble((Xa_old-Xa_new)*(Xa_old-Xa_new)*dx))
        error_b = sqrt(assemble((Xb_old-Xb_new)*(Xb_old-Xb_new)*dx))
        print("Error is= ", error_a, error_b)
        if error_a+error_b < atol:
            done = True
        Xa_old = Xa_new
        Xb_old = Xb_new
        iter = iter+1
    # end of pdas loop
    print('number of iterations',iter)
    ## Errors: L2 for u, H1 for y and p
    L2error[j] = errornorm(ue,u,norm_type='L2')
    State_error_y[j]=errornorm(ye, y, norm_type='H1')
    Adjointerror_p[j]=errornorm(pe, p, norm_type='H1')
# end of loops
# print some information
print("maximal diameter of mesh is =", h )
print("L2 error",L2error )
print("H1 error-state", State_error_y )
print("H1 error-adjoint", Adjointerror_p)
### Post Processing
# Plot the solutions
# and export to paraview
plt.figure(figsize=(15,5))
# Control
plt.subplot(1, 3, 1)
fig = plot(u)
plt.colorbar(fig, fraction=0.046, pad=0.04)
plt.title('Control variable u')
vtkfile = File('control.pvd')
vtkfile << u
# State
plt.subplot(1,3,2)
fig = plot(y)
plt.colorbar(fig, fraction=0.046, pad=0.04)
plt.title('State variable y')
vtkfile = File('state.pvd')
vtkfile << y
# Adjoint State
plt.subplot(1,3,3)
fig = plot(p)
plt.colorbar(fig, fraction=0.046, pad=0.04)
plt.title('Adjoint state p')
vtkfile = File('adjoint.pvd')
vtkfile << p
# Saving
plt.savefig('solution.pdf')

# error plots
plt.figure(2)
plt.rcParams.update({'font.size': 8.5})
# convergence rate (adapted by hand)
# adjusted control
const_shift = -1.13571                      # shift of line in y-direction
const_rate = 1.00838                          # slope for line in loglog Plot
rate = pow(10,const_shift)*np.exp((const_rate*np.log(h)))    # function to plot
p1,=plt.loglog(h,rate,'m-.', label='Adjusted control')

# adjusted state
```

C

```
const_shift = -0.314198                    # shift of line in y-direction
const_rate = 0.99923                       # slope for line in loglog Plot
ratey = pow(10,const_shift)*np.exp((const_rate*np.log(h)))    # function to plot
p2,=plt.loglog(h,ratey,'r-.', label='Adjusted state')

# adjusted adjoint state
const_shift = -0.314022                    # shift of line in y-direction
const_rate = 0.999315                      # slope for line in loglog Plot
ratep = pow(10,const_shift)*np.exp((const_rate*np.log(h)))    # function to plot
p3,=plt.loglog(h,ratep,'g-.', label='Adjusted adj. state')
p4,=plt.loglog(h, L2error, 'y--o', label=r'$\Vert u-u_h\Vert_{L^2(\Omega)}$')
p5,=plt.loglog(h, State_error_y, '--D', label=r'$\Vert y-y_h\Vert_{H^1(\Omega)}$')
p6,=plt.loglog(h, Adjointerror_p, '--x', label=r'$\Vert p-p_h\Vert_{H^1(\Omega)}$')
slope_marker((h[2],L2error[2]-0.1*L2error[2]), (1,1))
#slope_marker((h[1],State_error_y[1]-0.1*State_error_y[1]), (1,1))
plt.xlabel('h')
plt.ylabel('Error')
plt.title("Convergence Rates, control, state and adjoint state")

plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
plt.tight_layout()
plt.savefig('convergence_ich.pdf')
plt.show()
```

A.2 Optimality system Solver

```
from dolfin import *

import matplotlib.pyplot as plt
import sys
import numpy as np

def SolverADR(mesh,PU,PY,beta,phi,yd,g,lam):
    ### Parameters
    epsi =Constant(1.0)
    kappa=Constant(1.0)
    be=Constant((3.0, 4.0))
    ### BC parameter for mixed formulation
    Dbc= Constant(0.0)
    ##### function spaces:  mixed formulation #####
    U = FunctionSpace(mesh,PU)
    Y = FunctionSpace(mesh,PY)
    # ## Interpolation of data
    # beta=interpolate(beta,U)
    # phi=interpolate(phi,U)
    # yd=interpolate(yd,Y)
    # g=interpolate(g,U)
    # (u,y,p) \in (U,Y,Y) but we need the element spaces
    X = FunctionSpace(mesh,MixedElement([PU,PY,PY]))
    ## Boundary Conditions for State and Dual eqs.
    bcy = DirichletBC(X.sub(1),Dbc,'on_boundary')
    bcp = DirichletBC(X.sub(2),Dbc,'on_boundary')
    BC=[bcy, bcp]
    ##### solve the ADR equations #####
    (u,y,p) = TrialFunctions(X)
    (w,v,q) = TestFunctions(X)
    ## Bilinear Forms
    Co= (beta*p*w+u*w)*dx
    Ay= (inner(epsi*grad(y),grad(v))+(inner(be,grad(y)))*v+kappa*y*v-u*v)*dx
    Ap= (inner(epsi*grad(p),grad(q))-(inner(be,grad(p)))*q+kappa*p*q-y*q)*dx
    a = Co+Ay+Ap
    L = (phi*w+g*v-yd*q)*dx
    ## Assign the solution
    uh_yh_ph = Function(X)
    ## Solve
    solve(a == L,uh_yh_ph,BC)
    #### Split components of the solution #####
    uh = Function(U)
    yh = Function(Y)
    ph = Function(Y)
    assign(uh,uh_yh_ph.sub(0))
    assign(yh,uh_yh_ph.sub(1))
    assign(ph,uh_yh_ph.sub(2))

    return(uh,yh,ph)
```

A.3 L^2-Projection

```python
 #!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Jul 24 20:02:38 2021

@author: juancarlos
"""
from dolfin import *
import numpy as np

def L2Projection(mesh, f, polydegree):
    V = FunctionSpace(mesh,'DG',polydegree)
    u = TrialFunction(V)
    v = TestFunction(V)
    r = u*v*dx
    rhs = f*v*dx
    Pu = Function(V)
    solve(r == rhs, Pu)

    return(Pu)
```

A.4 Indicator Functions

A.4.1 χ_a

```python
 #!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 14 21:43:57 2021
This script creates the indicatrix function of the Active set A^a.
It needs three inputs: the functions to compare, f and ua as well as the
function space U.
@author: juancarlos
"""

from dolfin import *
import numpy as np
import matplotlib.pyplot as plt

def project_up(f,U,ua):
    # Build the vectors to compare
    fv = f.vector()[:]
    upperv = ua.vector()[:]
    # Indicatrix function
    cond = np.where(fv<upperv,1,0)
    # Build the vector with the binary values
    proj_up = Function(U)
    proj_up.vector()[:] = cond

    return proj_up
```

A.4.2 χ_b

```python
 #!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Jul 14 21:43:57 2021
This script creates the indicatrix function of the Active set A^b.
It needs three inputs: the functions to compare, f and ua as well as the
function space U.
@author: juancarlos
"""

from dolfin import *
import numpy as np
import matplotlib.pyplot as plt

def project_low(f,U,ub):
    # Build the vectors to compare
    fv = f.vector()[:]
    upperv = ub.vector()[:]
    # Indicatrix function
    cond = np.where(fv>upperv,1,0)
    # Build the vector with the binary values
    proj_low = Function(U)
    proj_low.vector()[:] = cond

    return proj_low
```

Appendix B Implementation in cashocs

Here the interested reader can find the code used to implementate the PDAS to the ADR optimal Control Problem in cashocs

B.1 Main Program

```
 # Copyright (C) 2020-21 Sebastian Blauth
#
# This file is part of CASHOCS.
#
# CASHOCS is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# CASHOCS is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with CASHOCS.  If not, see <https://www.gnu.org/licenses/>.

## """For the documentation of this demo see
## https://cashocs.readthedocs.io/en/latest/demos/optimal_control/doc_box_constraints.html.
## """

from fenics import *
import numpy as np
import cashocs
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
from plotslopes import *
# config file: see https://cashocs.readthedocs.io/en/latest/demos/optimal_control/
    doc_box_constraints.html
config = cashocs.load_config('config_box.ini')
# Maximum number of uniforment refinements, we start with 16 elements
Nref = 5
### Parameters
epsi =Constant(1.0)
kappa=Constant(1.0)
be=Constant((3.0, 4.0))
g=Expression('(6*x[0]-6*pow(x[0],2)+5*x[1]-13*x[0]*x[1]\
              +7*pow(x[0],2)*x[1]-5*pow(x[1],2)+5*x[0]*pow(x[1],2)\
                  +pow(x[0],2)*pow(x[1],2)\
                      -min(2.0, max(-2.0, -(1-x[0])*x[0]*(1-x[1])*x[1])))'\
                          ,degree=1)
yde = Expression('(2*x[0] - 2*pow(x[0],2) + x[1] -14*x[0]*x[1] + 8*x[1]*pow(x[0],2) \
                  -pow(x[1],2) + 6*x[0]*pow(x[1],2)) ', degree=1)
### exact solutions
ye=Expression('x[0]*x[1]*(1-x[0])*(1-x[1])',degree=4)
pe=Expression('-x[0]*x[1]*(1-x[0])*(1-x[1])',degree=4)
ue=Expression('min(2.0, max(-2.0, -((1-x[0])*x[0]*(1-x[1])*x[1])/lam))',lam=1,degree=3)
### Store information ###
# a counter
array=np.arange(Nref)
# NDOF of each mesh
l=np.empty(Nref, dtype=int)
# Maximum diameter
h=np.empty(Nref)
# Errors
L2error = np.empty(Nref)
State_error_y = np.empty(Nref)
Adjointerror_p = np.empty(Nref)
### Loops ###
# exterior loop: set each mesh and calculate the approx. errors
for j in array:
    ## NDOF ##
    l[j]=2**(j+4)
    ## mesh size
    h[j]=np.divide(1,(np.multiply(2,l[j])))
    ## mesh
    mesh, subdomains, boundaries, dx, ds, dS = cashocs.regular_mesh(l[j])
    V = FunctionSpace(mesh, 'CG', 1)
    #W = FunctionSpace(mesh, 'DG', 0)
    y = Function(V)
    p = Function(V)
    u = Function(V)
    ## Box condition limits
    u_a = interpolate(Expression('-2.0', degree=1), V)

    u_b = interpolate(Expression('2.0', degree=1), V)

    ### Primal  (forward) weak equation ###

    e = epsi*inner(grad(y), grad(p))*dx +inner(be,grad(y))*p*dx+kappa*y*p*dx- u*p*dx-g*p*dx
```

```
80
81      bcs = cashocs.create_bcs_list(V, Constant(0), boundaries, [1, 2, 3, 4])
82
83      alpha = Constant(1.0)
84      J = Constant(0.5)*(y - yde)*(y - yde)*dx + Constant(0.5*alpha)*u*u*dx
85
86      ## Box condition
87      cc = [u_a, u_b]
88      ## Solver
89      ocp = cashocs.OptimalControlProblem(e, bcs, J, y, u, p, config, control_constraints=cc)
90      ocp.solve('pdas', 1e-9, 0.0, 100)
91      ## primal dual Active set strategy, rel. tol=1e-9, abs tol.=0, max it.=100 --^
92      ## Check point
93      assert np.alltrue(u_a.vector()[:] <= u.vector()[:]) and np.alltrue(u.vector()[:] <= u_b.vector
            ()[:])
94      L2error[j] = errornorm(ue, u, norm_type='L2', degree_rise=2)
95      State_error_y[j] = errornorm(ye, y, norm_type='H1', degree_rise=2)
96      Adjointerror_p[j] = errornorm(pe, p, norm_type='H1', degree_rise=2)
97      h[j]=1/(2*l[j])
98  # L2errorList.append(L2error)
99  # State_errorList_y.append(H1errorstate)
100 # AdjointerrorList_p.append(H1erroradjoint)}
101
102 print("maximal diameter of the mesh is =", h)
103 print("L2 error", L2error)
104 print("H1 error-state", State_error_y)
105 print("H1 error-adjoint", Adjointerror_p)
106
107 ## Post processing
108 # Plot the solutions
109 plt.figure(figsize=(15,5))
110 # control
111 plt.subplot(1, 3, 1)
112 fig = plot(u)
113 plt.colorbar(fig, fraction=0.046, pad=0.04)
114 plt.title('Control variable u')
115 vtkfile = File('control.pvd')
116 vtkfile << u
117 # state
118 plt.subplot(1,3,2)
119 fig = plot(y)
120 plt.colorbar(fig, fraction=0.046, pad=0.04)
121 plt.title('State variable y')
122 # adjoint state
123 plt.subplot(1,3,3)
124 fig = plot(-p) # We plot -p because cashocs use the scheme of Hinze to solve
125 plt.colorbar(fig, fraction=0.046, pad=0.04)
126 plt.title('Adjoint state p')
127 plt.show()
128 plt.savefig('solution_cashocs.png')
129 plt.tight_layout()
130
131 # error plots
132 plt.figure(2)
133 plt.rcParams.update({'font.size': 8.5})
134 # convergence rate (adapted by hand)
135 # adjusted control
136 const_shift = -0.350016                    # shift of line in y-direction
137 const_rate = 1.99787                       # slope for line in loglog Plot
138 rate = pow(10,const_shift)*np.exp((const_rate*np.log(h)))    # function to plot
139 pl,=plt.loglog(h,rate,'m-.', label='Adjusted control')
140 # adjusted state
141 const_shift = -0.314052                    # shift of line in y-direction
142 const_rate = 0.9993                        # slope for line in loglog Plot
143 ratey = pow(10,const_shift)*np.exp((const_rate*np.log(h)))    # function to plot
144 p2,=plt.loglog(h,ratey,'r-.', label='Adjusted state')
145 # adjusted adjoint state
146 const_shift = -0.313897                    # shift of line in y-direction
147 const_rate = 0.999375                      # slope for line in loglog Plot
148 ratep = pow(10,const_shift)*np.exp((const_rate*np.log(h)))    # function to plot
149 p3,=plt.loglog(h,ratep,'g-.', label='Adjusted adj. state')
150 p4,=plt.loglog(h, L2error, 'y--o', label=r'$\Vert u-u_h\Vert_{L^2(\Omega)}$')
151 p5,=plt.loglog(h, State_error_y, '--D', label=r'$\Vert y-y_h\Vert_{\Omega}$')
152 p6,=plt.loglog(h, Adjointerror_p, '--x', label=r'$\Vert p-p_h\Vert_{\Omega}$')
153 slope_marker((h[2],L2error[2]-0.1*L2error[2]), (2,1))
154 slope_marker((h[2],State_error_y[2]-0.1*State_error_y[2]), (1,1))
155 plt.xlabel('h')
156 plt.ylabel('Error')
157 plt.title("Convergence History Plot, un. refinement")
158 plt.legend(bbox_to_anchor=(1.05, 1.0), loc='upper left')
159 plt.tight_layout()
160 plt.savefig('convergence_cashocs.pdf')
161 plt.show()
```