FIRST ORDER NUMERICAL ALGORITHMS FOR SOME OPTIMAL CONTROL PROBLEMS WITH PDE CONSTRAINTS

by

SONG Yongcun

B.Sc. *J.L.U*; M.Phil. *J.L.U and H.K.B.U*

A thesis submitted in partial fulfillment of the requirements for
the Degree of Doctor of Philosophy
at The University of Hong Kong.

August 2021

Abstract of thesis entitled

FIRST ORDER NUMERICAL ALGORITHMS FOR SOME OPTIMAL CONTROL PROBLEMS WITH PDE CONSTRAINTS

Submitted by

SONG Yongcun

for the degree of Doctor of Philosophy
at The University of Hong Kong
in August 2021

In this thesis, we focus on designing first order numerical algorithms for some optimal control problems with partial differential equation (PDE) constraints. The first part is focused on some PDE-constrained optimal control problems with additional box or sparsity control constraints. We design some operator splitting type algorithms for these problems, and their common feature is that the PDE constraints and the additional box or sparsity control constraints are treated separately in numerical implementation. In particular, we develop an inexact Uzawa method and an inexact alternating direction method of multipliers for elliptic and parabolic optimal control problems with box control constraints, respectively; and a primal-dual hybrid gradient algorithm for a sparse optimal control problem with diffusion-advection equation constraint. The second part is focused on the bilinear optimal control of an advection-reaction-diffusion system, where the control variable arises as the velocity field in the advection term. For this problem, we prove the existence of optimal controls, derive the first-order optimality conditions in general settings, and design a nested conjugate gradient method. These new algorithms are designed in accordance with the structures of the problems under consideration, and they can be implemented easily. Their efficiency is promisingly validated by the results of some preliminary numerical experiments and convergence properties are also studied.

FIRST ORDER NUMERICAL ALGORITHMS FOR SOME OPTIMAL CONTROL PROBLEMS WITH PDE CONSTRAINTS

by

SONG Yongcun

B.Sc. *J.L.U*; M.Phil. *J.L.U and H.K.B.U*

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy
at The University of Hong Kong.

August 2021

Declaration

I declare that this thesis represents my own work. All the works are done under the supervision of Prof. YUAN Xiaoming during the period 2018–2021 for the degree of Doctor of Philosophy at The University of Hong Kong. The work submitted has not been previously included in a thesis, dissertation or report submitted to any institution for a degree, diploma or other qualification.

SONG Yongcun

Acknowledgements

First of all, I would like to express my deep gratitude to my supervisor, Prof. YUAN Xiaoming, for his continuous support, patience, motivation, enthusiasm, and research encouragement during my study at The University of Hong Kong. It is my great honor that can be supervised by such a knowledgeable, inspiring, and supportive professor. His guidance helped me in all the time of research and preparation for this thesis. His tireless enthusiasm for research and rigorous academic attitude have greatly impressed me.

Besides my supervisor, my sincere thanks also go to Prof. Roland GLOWINSKI from University of Houston. His profound mathematical expertise extends my research perspectives greatly. I am grateful to him for all the help, insights, and encouragement he has given to me over the past few years.

Furthermore, I would like to thank all faculty members and staff in the department of mathematics and my friends at The University of Hong Kong. It is their suggestions and encouragement that help me complete the graduate study successfully.

Last but not the least, I thank my beloved family, whose support and love over the years has helped me immeasurably along the way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we first introduce some optimal control problems with partial differential equation (PDE) constraints considered in this thesis. Then, some first order numerical algorithms are introduced, which are closely related to the methods we will use for solving the optimal control problems. Finally, the outline of this thesis is presented.

1.1 Optimal control problems with PDE constraints

Generally, an optimal control problem with PDE constraints can be abstractly represented as

$$\min_{y\in Y, u\in U} J(y,u), \quad \text{s.t.} \quad e(y,u)=0, u\in U_{ad}, y\in Y_{ad}, \tag{1.1}$$

where U and Y are Banach spaces, $U_{ad}\subset U$ and $Y_{ad}\subset Y$ are closed convex sets; $J: Y\times U\to\mathbb{R}$ is the objective functional which usually consists of a data fidelity term and a regularization term; the operator $e: Y\times U\to Z$ with Z a Banach space, and $e(y,u)=0$ represents a PDE or a system of coupled PDEs. The state variable $y\in Y$ describes the state (e.g., temperature distribution) of the considered system modeled by $e(y,u)=0$; the control variable $u\in U$ is a parameter (e.g., source term) that shall be adapted in an optimal way; the

control constraint $u \in U_{ad}$ and the state constraint $y \in Y_{ad}$ describe some physical restrictions and realistic requirements. We assume that for each $u \in U_{ad}$ there exists a unique solution $y(u)$ such that $e(y(u), u) = 0$. Then, the optimal control problem (1.1) can be written as a reduced form:

$$\min_{u \in U} \hat{J}(u) := J(y(u), u), \quad u \in U_{ad},\ y(u) \in Y_{ad},$$

which plays an important role in theoretical analysis and algorithmic design for optimal control problems with PDE constraints, see e.g., [49, 81, 83, 103, 171].

Optimal control problems with PDE constraints capture important applications in various scientific areas, such as physics, chemistry, engineering, medicine and financial engineering. We refer to, e.g. [81, 82, 83, 103, 125, 171], for a few references. These problems have received tremendous attentions in the past decades mainly since the pioneering work of J. L. Lions [125], see, e.g., [81, 82, 83, 126, 194, 196]. Solving these problems is usually very challenging, from both theoretical analysis and algorithmic design perspectives. For instance, state equations are coupled with additional control or state constraints, the dimensionality after proper discretization is extremely high, and coefficient matrices after discretization are possibly extremely ill-conditioned. Because of these difficulties, there are not too many efficient algorithms in the literature, especially for some optimal control problems with time-dependent PDE constraints.

Despite the fact that there exist many different types of PDEs, to expose our main ideas clearly, we will focus on (bi-)linear elliptic and parabolic type optimal control problems, which have a wide range of applications in e.g., diffusion, heat flow, elastic deformation, and thermal treatment in cancer therapy, we refer to e.g., [83, 103, 171] for more discussions.

1.1.1 Elliptic optimal control problems with control constraints

We consider the following elliptic optimal control problem with control constraints

$$\min_{y \in Y, u \in U_{ad}} J(y, u) := \frac{1}{2}\|y - y_d\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\|u\|^2_{L^2(\Omega)}, \tag{1.2}$$

where $y \in Y := H_0^1(\Omega)$ and $u \in U := L^2(\Omega)$ satisfy the following elliptic equation:

$$\begin{cases} \mathcal{K}y = u \quad \text{in} \quad \Omega, \\ \quad y = 0 \quad \text{on} \quad \Gamma. \end{cases} \tag{1.3}$$

In (1.2)-(1.3), $\Omega \subset \mathbb{R}^d$ $(d \geq 1)$ is a convex polyhedral domain with boundary $\Gamma := \partial\Omega$, and the desired state $y_d \in L^2(\Omega)$ is given. The admissible set U_{ad} is defined by

$$U_{ad} = \{u \in L^\infty(\Omega) | a \leq u(x) \leq b, \text{ a.e. in } \Omega\} \subset L^2(\Omega),$$

where $-\infty < a < b < +\infty$ are two given constants.

In the equation (1.3), the linear second-order elliptic operator $\mathcal{K}$ is defined by

$$\mathcal{K}y = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \sum_{j=1}^{d} a_{ij} \frac{\partial y}{\partial x_j} + c_0 y,$$

where $0 \leq c_0 \in L^\infty(\Omega), 0 < a_{ij} \in L^\infty(\Omega), \forall 1 \leq i, j \leq d$ are given coefficients. In addition, we assume that the matrix-valued function $(a_{ij})_{1 \leq i,j \leq d}$ satisfies $a_{ij} = a_{ji}$ and

$$\sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij}(x) \xi_i \xi_j \geq \gamma \|\boldsymbol{\xi}\|^2, \quad \forall \boldsymbol{\xi} = \{\xi_i\}_{i=1}^d \in \mathbb{R}^d \quad \text{a.e. in } \Omega, \tag{1.4}$$

with $\gamma \geq 0$ and $\|\cdot\|$ the canonical Euclidean norm of $\mathbb{R}^d$.

The problem (1.2)-(1.3) has a wide range of applications in various areas, such as deformation of elastic membranes, heat conduction, and electrostatics; we refer to [83, 171] for further discussions. Existence and uniqueness of the solution to the problem (1.2)-(1.3) have been proved in [125].

Let $I_{U_{ad}}(\cdot)$ be the indicator function of the admissible set U_{ad} and suppose that (y^*, u^*) is the unique solution of the problem (1.2)-(1.3). Then, following the standard arguments as those in [171], it is easy to show that the optimality condition of the problem (1.2)-(1.3) reads as

$$\begin{cases} (\alpha I + \partial I_{U_{ad}})(u^*) + p^* \ni 0, \\ \mathcal{K}^* p^* = y^* - y_d, \\ \mathcal{K} y^* = u^*, \end{cases} \tag{1.5}$$

where $\partial I_{U_{ad}}$ is the subdifferential of $I_{U_{ad}}$, p^* is the corresponding adjoint variable, and $\mathcal{K}^*$ is the adjoint operator of $\mathcal{K}$. Clearly, the optimality condition (1.5) can be equivalently written as the following nonlinear saddle point problem

$$\begin{cases} 0 \in (A + \mathcal{G})(w) + B^* v - c, \\ 0 = Bw - d, \end{cases} \tag{1.6}$$

with

$$A = \begin{pmatrix} \alpha I & 0 \\ 0 & I \end{pmatrix}, \ B = \begin{pmatrix} I & -\mathcal{K} \end{pmatrix}, \ w = \begin{pmatrix} u^* \\ y^* \end{pmatrix}, \ v = p^*, \ c = \begin{pmatrix} 0 \\ y_d \end{pmatrix} \tag{1.7}$$

$$d = 0 \quad \text{and} \quad \mathcal{G}(w) = \Big\{ \begin{pmatrix} \nu \\ 0 \end{pmatrix} | \nu \in \partial I_{U_{ad}}(u) \Big\}. \tag{1.8}$$

Since the problem (1.2)-(1.3) is convex, the optimality condition (1.5) is also sufficient. Accordingly, we can solve the nonlinear saddle point problem (1.6) to obtain the optimal solution of (1.2)-(1.3).

1.1.2 Parabolic optimal control problems with control constraints

Typically, an optimal control problem with a parabolic PDE constraint and a box constraint on the control variable reads:

$$\min_{y \in Y, u \in U_{ad}} \quad J(y,u) := \frac{1}{2} \iint_Q |y - y_d|^2 dxdt + \frac{\alpha}{2} \iint_{\mathcal{O}} |u|^2 dxdt \tag{1.9}$$

and the state equation $e(y,u) = 0$ is specified as

$$\begin{cases} \dfrac{\partial y}{\partial t} - \nu \Delta y + a_0 y = u \chi_{\mathcal{O}}, & \text{in} \quad \Omega \times (0,T), \\ y = 0, & \text{on} \quad \Gamma \times (0,T), \\ y(0) = \varphi, \end{cases} \tag{1.10}$$

where Ω is a bounded domain in $\mathbb{R}^d$ ($d \geq 1$) and $\Gamma = \partial\Omega$ is the boundary of Ω; ω is an open subset of Ω and $0 < T < +\infty$; the domain $Q = \Omega \times (0,T)$ and $\mathcal{O} = \omega \times (0,T)$. The target function y_d is given in $L^2(Q)$ and the admissible set U_{ad} is defined by

$$U_{ad} = \{v | v \in L^\infty(\mathcal{O}), a \leq v(x;t) \leq b \text{ a.e. in } \mathcal{O}\} \subset L^2(\mathcal{O}).$$

In addition, we denote by $\chi_{\mathcal{O}}$ the characteristic function of the set $\mathcal{O}$. The constant $\alpha > 0$ is a regularization parameter; a and b are given constants; the initial value φ is given in $L^2(\Omega)$. The coefficients a_0 $(\geq 0) \in L^\infty(Q)$ and ν is a positive constant. The problem (1.9)–(1.10) plays an important role in e.g., physics, chemistry, and engineering, see [81, 82, 83, 194, 196]. Existence and uniqueness of the solution to the problem (1.9)–(1.10) can be proved in a standard argument as studied in [125]; we refer to [171] for the details.

1.1.3 Sparse optimal control problems with PDE constraints

In optimal control problems with PDE constraints, usually we can only put the controllers in some small regions instead of the whole domain under investigation. As a consequence, a natural question arises: how to determine the optimal locations and the intensities of the controllers? This concern inspires a class of optimal control problems where the controls are sparse i.e., they are only non-zero in a small region of the domain; and the so-called sparse optimal control problems are obtained. Sparse optimal control problems usually appear in a variety of applications including optimal actuator placement [164], polution source identification [34, 56, 124, 123], and impulse control [44]; and they have been intensively studied in the literature, see e.g., [31, 32, 33, 36, 116, 164, 161], and references therein.

To the best our knowledge, the first work dedicated to the sparse optimal control of PDEs is [164], where the following elliptic optimal control problem with L^1-regularized objective functional is considered.

$$\min_{y\in H_0^1(\Omega), u\in U_{ad}} J(y,u) = \frac{1}{2}\|y - y_d\|^2_{L^2(\Omega)} + \frac{\alpha}{2}\|u\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|u\|_{L^1(\Omega)}, \tag{1.11}$$

where y and u satisfy the following state equation:

$$\begin{cases} -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \sum_{j=1}^{d} a_{ij} \frac{\partial y}{\partial x_j} + c_0 y = u \quad \text{in} \quad \Omega, \\ \qquad\qquad\qquad\qquad\qquad\qquad y = 0 \quad \text{on} \quad \Gamma. \end{cases} \tag{1.12}$$

In (1.11)-(1.12), $\Omega \subset \mathbb{R}^d$ $(d \geq 1)$ is a convex polyhedral domain with boundary $\Gamma := \partial\Omega$, and the desired state $y_d \in L^2(\Omega)$ is given. The constants $\alpha > 0$ and $\beta > 0$ are regularization parameters. The coefficients $0 \leq c_0 \in L^\infty(\Omega), 0 < a_{ij} \in L^\infty(\Omega), \forall 1 \leq i, j \leq d$. In addition, we assume that the matrix-valued function $(a_{ij})_{1\leq i,j\leq d}$ satisfies (1.4). The admissible set U_{ad} is defined by

$$U_{ad} = \{u \in L^\infty(\Omega) | a \leq u(x) \leq b, \text{ a.e. in } \Omega\} \subset L^2(\Omega),$$

where $a, b \in L^2(\Omega)$ with $a < 0 < b$ almost everywhere.

Uniqueness and existence of the solution to the problem (1.11)-(1.12) has been proved in [164]. We note that if $\alpha = 0$ and $U_{ad} = L^2(\Omega)$, the problem (1.11)-(1.12) is not well-posed because the existence of a solution cannot be guaranteed due to the non-reflexivity of $L^1(\Omega)$. Due to the presence of the nonsmooth L^1-regularization term, the structure of the optimal control of (1.11)-(1.12) differs significantly from what one obtains for the usual smooth L^2- regularization like (1.2)-(1.3). Precisely, as analyzed in [164, 180], the optimal control of (1.11)-(1.12) has small support, and the support is adjustable in terms of the tuning of the regularization parameter β in (1.11). Indeed, it has been shown in [164] that the optimal control is zero on the whole domain Ω when the parameter β is sufficiently large. More study on optimal control problems with L^1-regularized objective functional can be found in e.g., [31, 161, 180].

In many applications, it is desirable to place the controllers only in finitely many points of the domain, or along a line (in two-dimensional space), or on a surface (in three-dimension space). This motivates us to use controllers that are localized in a set of Lebesgue measure zero, which can be achieved by modeling the controls in measure spaces. In particular, one can use a measure norm of the controls as the regularization term in the objective functional, then the resulting optimal controls have the desired sparsity property. For instance, elliptic optimal control problems in measure spaces have been considered in e.g., [32, 46], in which the following optimal control problem is considered:

$$\min J(y, u) = \frac{1}{2}\|y - y_d\|^2_{L^2(\Omega)} + \alpha\|u\|_{\mathcal{M}(\Omega)}, \tag{1.13}$$

subject to the elliptic equation (1.12). Above, the parameter $\alpha > 0$ and the target $y_d \in L^2(\Omega)$ are given; we denote by $\mathcal{M}(\Omega)$ the space of all bounded Borel

measures on Ω and the corresponding norm is given by

$$\|u\|_{\mathcal{M}(\Omega)} = \sup_{\phi \in C_0(\Omega), \|\phi\|_{C_0(\Omega)} \leq 1} \int_\Omega \phi du,$$

where $C_0(\Omega)$ is the space of all continuous functions with compact support in Ω, endowed with the norm $\|\phi\|_{C_0} = \sup_{x\in\Omega} |\phi(x)|_\infty$. The problem (1.13) is well-posed and has a unique optimal control in $\mathcal{M}(\Omega)$, see e.g., [46] for the details. Additionally, it has been shown in [46] that the optimal control of (1.13) is nonzero only on the sets where the constraint on the adjoint variable is active; and the larger the regularization parameter α, the smaller the support of the optimal control. Other types of optimal control problems in measure spaces have also been studied in the literature, we refer to [32, 33, 36, 116] for a few references.

In addition, it is worth mentioning that the initial conditions of some diffusion systems can also be considered as the control variables in sparse optimal control problems. The resulting optimal control models play a crucial role in the sparse initial source identification for diffusion systems, see e.g., [34, 35, 120, 134]. Hence, how to solve these sparse optimal control problems efficiently is one of the central concerns for solving sparse initial source identification problems. We focus on this topic in Chapter 4, where we will propose a new optimal control based numerical approach to solve the sparse initial source identification for diffusion systems and present an efficient optimization algorithm to solve the resulting sparse optimal control problem.

1.1.4 Bilinear optimal control of an advection-reaction-diffusion system

In a typical mathematical model of an optimal control problem with PDE constraints, either boundary or internal locally distributed controls are usually used (see e.g., (1.3) and (1.10)); these controls have localized support and are called additive controls because they arise in the model equations as additive terms. It is worth noting that additive controls describe the effect of external added sources or forces and they do not change the principal intrinsic properties

of the controlled system. Hence, they are not suitable to deal with processes whose principal intrinsic properties should be changed by some control actions. For instance, if we aim at changing the reaction rate in some chain reaction-type processes from biomedical, nuclear, and chemical applications, additive controls amount to controlling the chain reaction by adding into or withdrawing out of a certain amount of the reactants, which is not realistic. To address this issue, a natural idea is to use certain catalysts or smart materials to control the systems, which can be mathematically modeled by optimal control problems with bilinear controls. We refer to [112] for more detailed discussions.

Bilinear controls, also known as multiplicative controls, enter the model as coefficients of the corresponding partial differential equations (PDEs). These bilinear controls can change some main physical characteristics of the system under investigation, such as a natural frequency response of a beam or the rate of a chemical reaction. In the literature, bilinear controls of distributed parameter systems have become an increasingly popular topic and bilinear optimal control problems constrained by various PDEs, such as elliptic equations [115], convection-diffusion equations [17], parabolic equations [111], the Schrödinger equation [108] and the Fokker-Planck equation [66], have been widely studied both mathematically and computationally.

In particular, bilinear controls play a crucial role in optimal control problems modeled by advection-reaction-diffusion systems. On one hand, the control can be the coefficient of the diffusion or the reaction term. For instance, a system controlled by the so-called catalysts that can accelerate or slow down various chemical or biological reactions can be modeled by a bilinear optimal control problem for an advection-reaction-diffusion equation where the control arises as the coefficient of the reaction term [111]; this kind of bilinear optimal control problems have been studied in e.g., [17, 28, 111, 112]. On the other hand, the systems can also be controlled by the velocity field in the advection term, which captures important applications in e.g., bioremediation [90], environmental remediation process [121], and mixing enhancement of different fluids [128]. We note that there is a very limited research being done on the velocity field controlled bilinear optimal control problems; and only some special one-dimensional

space cases have been studied in [90, 109, 121] for the existence of an optimal control and the derivation of first-order optimality conditions. To the best of our knowledge, no work has been done yet to develop efficient numerical methods for solving multi-dimensional bilinear optimal control problems controlled by the velocity field in the advection term. This motivates us to study the following bilinear optimal control problem constrained by an advection-reaction-diffusion equation, where the control enters into the model as the velocity field in the advection term.

Let Ω be a bounded domain of $\mathbb{R}^d$ with $d \geq 1$ and let Γ be its boundary. We consider the following bilinear optimal control problem:

$$\begin{cases} \boldsymbol{u} \in \mathcal{U}, \\ J(\boldsymbol{u}) \leq J(\boldsymbol{v}), \forall \boldsymbol{v} \in \mathcal{U}, \end{cases} \tag{BCP}$$

with the objective functional J defined by

$$J(\boldsymbol{v}) = \frac{1}{2} \iint_Q |\boldsymbol{v}|^2 dxdt + \frac{\alpha_1}{2} \iint_Q |y - y_d|^2 dxdt + \frac{\alpha_2}{2} \int_\Omega |y(T) - y_T|^2 dx, \tag{1.14}$$

and $y = y(t; \boldsymbol{v})$ the solution of the following advection-reaction-diffusion equation

$$\begin{cases} \dfrac{\partial y}{\partial t} - \nu \nabla^2 y + \boldsymbol{v} \cdot \nabla y + a_0 y = f \quad \text{in} \quad Q, \\ y = g \quad \text{on} \quad \Sigma, \\ y(0) = \phi. \end{cases} \tag{1.15}$$

Above and below, $Q = \Omega \times (0, T)$ and $\Sigma = \Gamma \times (0, T)$ with $0 < T < +\infty$; $\alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_1 + \alpha_2 > 0$; the target functions y_d and y_T are given in $L^2(Q)$ and $L^2(\Omega)$, respectively; the diffusion coefficient $\nu > 0$ and the reaction coefficient a_0 are assumed to be constants; the functions $f \in L^2(Q)$, $g \in L^2(0, T; H^{1/2}(\Gamma))$ and $\phi \in L^2(\Omega)$. The set $\mathcal{U}$ of the admissible controls is defined by

$$\mathcal{U} := \{\boldsymbol{v} | \boldsymbol{v} \in [L^2(Q)]^d, \nabla \cdot \boldsymbol{v} = 0\}.$$

Clearly, the control variable $\boldsymbol{v}$ arises in (BCP) as a flow velocity field in the advection term of (1.15), and the divergence-free constraint $\nabla \cdot \boldsymbol{v} = 0$ implies that the flow is incompressible. One can control the system by changing the flow velocity $\boldsymbol{v}$ in order that y and $y(T)$ are good approximations to y_d and y_T, respectively. Note that the objective functional J in (BCP) is nonconvex due to the nonlinear relationship between the state y and the control $\boldsymbol{v}$.

1.2 Some first order numerical algorithms

1.2.1 Conjugate gradient methods for optimization problems in Hilbert spaces

Since their invention in the 1950s, conjugate gradient (CG) methods have been proved to be easy to implement, low memory requirements, quite robust, and fast convergent. Consequently, CG methods are popular and very efficient for solving various linear and nonlinear problems, see e.g., [78, 138] and references therein. To introduce CG methods briefly, we follow [78] and discuss their applications to the following prototypical optimization problem in Hilbert spaces.

$$\begin{cases} u \in V, \\ J(u) \leq J(v), \forall v \in V, \end{cases} \tag{1.16}$$

where V is a real Hilbert space equipped with the inner product $(\cdot, \cdot)$ and the corresponding norm $\|\cdot\|$, $J : V \to \mathbb{R}$ is a differentiable functional. To guarantee the existence of a solution to (1.16), we further assume that J is coercive and weakly lower semi continuous over V, i.e.,

$$\lim_{\|v\| \to +\infty} J(v) = +\infty$$

and

$$\text{if } \lim_{n \to +\infty} v_n = v \text{ weakly in } V, \text{ then } \liminf_{n \to +\infty} J(v_n) \geq J(v).$$

Concerning the differentiability of J, we assume that J is either Fréchet-differentiable or Gâteaux-differentiable, and denote by $DJ(v) \in V'$ the differential of J at $v \in V$. Here and in what follows, V' is the dual space of V and we denote by $\langle \cdot, \cdot \rangle$ the duality pairing between V' and V.

Following [78], CG algorithms for solving problem (1.16) read as follows:

Initialize $u^0 \in V$, and solve

$$\begin{cases} g^0 \in V, \\ (g^0, v) = \langle DJ(u^0), v \rangle, \forall v \in V. \end{cases}$$

If $g^0 = 0$, then $u = u^0$; otherwise set $w^0 = g^0$.

For $k \geq 0$, u^k, g^k and w^k being known, the last two different from 0, one computes u^{k+1}, g^{k+1} and w^{k+1} as follows:

Compute the stepsize ρ_k by solving the following optimization problem

$$\begin{cases} \rho_k \in \mathbb{R}, \\ J(u^k - \rho_k w^k) \leq J(u^k - \rho w^k), \forall \rho \in \mathbb{R}. \end{cases} \tag{1.17}$$

Update u^{k+1} and g^{k+1}, respectively, by

$$u^{k+1} = u^k - \rho_k w^k,$$

and solving

$$\begin{cases} g^{k+1} \in V, \\ (g^{k+1}, v) = \langle DJ(u^{k+1}), v \rangle, \forall v \in V. \end{cases} \tag{1.18}$$

If $g^{k+1} = 0$, take $u = u^{k+1}$; otherwise,

Compute either

$$\beta_k = \frac{\|g^{k+1}\|^2}{\|g^k\|^2}, \text{ (Fletcher-Reeves update)}$$

or

$$\beta_k = \frac{(g^{k+1}, g^{k+1} - g^k)}{\|g^k\|^2}, \text{ (Polak-Ribière update)}$$

and then update

$$w^{k+1} = g^{k+1} + \beta_k w^k.$$

Do $k + 1 \to k$ and return to (1.17).

In practice, the implementation of CG algorithms requires the solutions of the linear variational problem (1.18) and the one-dimensional minimization problem (1.17) to update the descent direction and to compute the optimal step size, respectively. Suppose that the objective functional J is quadratic, namely, the functional $J(v)$ in (1.16) is given by

$$J(v) = \frac{1}{2} a(v, v) - L(v),$$

where the bilinear functional $a : V \times V \to \mathbb{R}$ is continuous, V-elliptic, and symmetric; $L : V \to \mathbb{R}$ is linear and continuous. In this case, it is easy to show that

$$\langle DJ(w), v\rangle = a(w, v) - L(v), \forall v, w \in V,$$

and the solution of the minimization problem (1.17) verifies

$$\rho_k = \frac{(g^k, w^k)}{a(w^k, w^k)}.$$

For a generic objective functional J, problem (1.17) usually has no closed-form solution. Hence, the step size ρ_k can not be computed exactly and it can only be determined by some line search strategies (see e.g., [138]) or solving the problem (1.17) iteratively. For instance, the Newton method is suggested in [78] for solving the problem (1.17) to compute the step size. To be concrete, it is easy to see that problem (1.17) is a particular case of the following minimization problem:

$$\begin{cases} \hat{\rho} \in \mathbb{R}, \\ J(u - \hat{\rho}w) \leq J(u - \rho w), \forall \rho \in \mathbb{R}. \end{cases} \tag{1.19}$$

Let $j(\rho) = J(u - \rho w)$, we then have

$$j'(\rho) = -\langle DJ(u - \rho w), w\rangle, \text{ and } j''(\rho) = \langle D^2 J(u - \rho w)w, w\rangle.$$

Applying the Newton method to the solution of (1.19), we obtain the following iterative scheme

An initial value ρ^0 is given in $\mathbb{R}$;

For $k \geq 0$,

$$\rho^{k+1} = \rho^k + \frac{\langle DJ(u - \rho^k w), w\rangle}{\langle D^2 J(u - \rho^k w)w, w\rangle}.$$

The convergence properties of CG methods in both of finite and infinite dimensional spaces have been widely studied in the literature, we refer to [75, 78, 138] and references therein for the details. Some other variants of CG methods and more discussions on the implementation of CG methods including preconditioning and restart strategies have also been discussed in the above references.

1.2.2 Inexact Uzawa methods for saddle point problems

Inexact Uzawa methods for linear saddle point problems

A typical linear saddle point problem appearing in discretized PDEs can be written as finding $w \in H_1$ and $v \in H_2$ such that

$$\begin{cases} Aw + B^\top v - f = 0, \\ \qquad Bw - g = 0, \end{cases} \tag{1.20}$$

where H_1 and H_2 are two finite dimensional Hilbert spaces, $f \in H_1$ and $g \in H_2$ are given, $A : H_1 \to H_1$ is a linear, symmetric, and positive definite operator, $B : H_1 \to H_2$ is a linear operator and $B^\top : H_2 \to H_1$ is the adjoint operator of B. For concrete applications of (1.20), see, e.g., [25, 41, 65] for the mixed finite element discretization of elasticity problems, Stokes equations and Maxwell equations; and [49, 103, 171] for the application of Lagrangian multiplier type methods to optimal control problems; and [42] for some parameter identification problems. Throughout, the Ladyzhenskaya-Babuška-Brezzi (LBB) condition in [15] is assumed. That is, for some positive number c_0, it holds that

$$\langle BA^{-1}B^\top v, v\rangle = \sup_{u\in H_1} \frac{\langle v, Bu\rangle^2}{\langle Au, u\rangle} \geq c_0\|v\|^2. \tag{1.21}$$

Here, $\|\cdot\|$ denotes the norm in the space of H_1 or H_2 corresponding to its respective inner product $\langle\cdot,\cdot\rangle$, whichever is clear according to the function type in the specific context under discussion despite of the same notation. As analyzed in [15], this condition ensures that the problem (1.20) is well defined with a unique solution point.

For iterative methods solving (1.20) in the literature, we refer to the Arrow-Hurwicz and Uzawa methods in [2, 5, 6, 21, 150], penalty and multiplier methods in [2, 50, 100, 139, 148], Krylov subspace methods in [20, 29, 61, 83, 141, 156, 179]. We also refer to the survey paper [10] and references therein for a thorough discussion. In particular, Uzawa-type methods have been widely used for various applications, especially for some large-scale problems arising in scientific computing areas, because of their economical requirement of memory and simplicity of implementation. Let us recall the classic Uzawa method in [2, 150] for solving

(1.20):

$$\begin{cases} w^{k+1} = A^{-1}(-B^\top v^k + f), \\ v^{k+1} = v^k + \omega(Bw^{k+1} - g), \end{cases} \tag{1.22}$$

where $\omega > 0$ is a relaxation parameter. As analyzed in [158], $\omega \in (0, 2/\rho(BA^{-1}B^\top))$ ensures the convergence of (1.22), where $\rho(\cdot)$ denotes the spectrum radius of an operator. As pointed out in [10], if we use the first equation in (1.22) to eliminate w^{k+1} from the second one, it is easy to see that (1.22) can be written as a stationary Richardson iteration applied to the following Schur complement system:

$$Sv = BA^{-1}f - g,$$

where the Schur complement S is defined as $S := BA^{-1}B^\top$. Despite the simplicity of the iterative scheme, the exact Uzawa method (1.22) requires computing A^{-1}, which might be computationally expensive and thus the first equation in (1.22) might need to be solved iteratively, as discussed in [43, 78]. It is also analyzed in, e.g. [21, 57, 105], that the exact Uzawa method (1.22) may converge slowly if the Schur complement is not well conditioned.

These concerns have then motivated many authors to consider inexact or preconditioned variants of the Uzawa method; we refer to [21, 57, 105, 150, 152] for a few references. Following [21], a class of inexact Uzawa type methods (1.22) can be unified as:

$$\begin{cases} w^{k+1} = Q_A^{-1}(Q_A w^k - Aw^k - B^\top v^k + f), \\ v^{k+1} = v^k + Q_B^{-1}(Bw^{k+1} - g), \end{cases} \tag{1.23}$$

where $Q_A : H_1 \to H_1$ and $Q_B : H_2 \to H_2$ are two symmetric positive definite operators playing the role of preconditioners of A and S, respectively. Obviously, if $Q_A = A$ and $Q_B = \frac{1}{\omega}I$, the framework of inexact Uzawa methods (1.23) reduces to the exact version (1.22). In practice, Q_A and Q_B are chosen such that Q_A^{-1} and Q_B^{-1} can be easily computed and that $Q_B^{-1}S$ is well conditioned; see [21, 57, 105] for insightful discussions. It is shown in [21] that the framework of inexact Uzawa methods (1.23) converges linearly as long as the preconditioners defining the framework are properly scaled.

Inexact Uzawa methods for nonlinear saddle point problems

The linear saddle point problem (1.20) can be generalized in different perspectives. For instance, the operator A in (1.20) can be replaced with a generic nonlinear operator $F : H_1 \to H_1$ and consequently the following generic nonlinear saddle point problem can be obtained:

$$\begin{cases} F(w) + B^\top v - f = 0, \\ Bw - g = 0. \end{cases} \tag{1.24}$$

Such nonlinear saddle point problems arise in, e.g., augmented Lagrangian formulations of inverse problems [42], electromagnetic Maxwell equations [41], and the nonlinear optimization problem [173]. The nonlinear operator F arising in these applications is indeed strongly monotone. It is generally difficult to solve the first nonlinear equation in (1.24) exactly, and thus various inexact Uzawa methods targeting inexact solutions of the nonlinear equation have been extensively studied in the literature. For example, setting $H_1 = \mathbb{R}^n$ and $H_2 = \mathbb{R}^m$ with $m \leq n$, the following inexact Uzawa method is proposed in [40]:

$$\begin{cases} F(w^{k+1}) = -B^\top v^k + f + \delta^k, \\ v^{k+1} = v^k + Q_k^{-1}(Bw^{k+1} - g), \end{cases} \tag{1.25}$$

where the vector $\delta^k \in \mathbb{R}^n$ represents an allowable error for solving the nonlinear equation at the k-th iteration, and Q_k are symmetric and positive definite matrices that should be adjusted iteratively. The global convergence and local superlinear convergence rate of the inexact Uzawa method (1.25) are proved in [40] under the conditions that $F(w)$ is Lipschitz continuous, $\{\|\delta^k\|\}$ converges to zero and $\|Q_k^{-1}\|$ are uniformly bounded for all k. In [106], it is suggested that the equations in (1.20) and the second equation in (1.24) are solved iteratively (e.g., by the preconditioned nonlinear CG method) so as to avoid computing the inverses of Q_A, Q_B and Q_k. The linear convergence rate in an energy-norm is established therein for (1.25), provided that the exact generalized Jacobian of $F(w)$ in the sense of Clarke (see [45]) is approximated by a well-chosen preconditioner for solving the Schur complement problem related to the second equation in (1.24). Note that the Lipschitz continuity of $F(w)$ is also required in [106] for its convergence analysis.

1.2.3 Alternating direction method of multipliers

We consider the following optimization problem:

$$\min_{v\in V} F(Bv) + G(v), \tag{1.26}$$

where V is a Hilbert space, the operator $B \in \mathcal{L}(V, H)$ with H a Hilbert space, the functionals $F : H \to \mathbb{R} \cup \{+\infty\}$ and $G : V \to \mathbb{R} \cup \{+\infty\}$ are proper, convex, and lower semicontinuous, and satisfy $dom(F \circ B) \cap dom(G) \neq \emptyset$, which guarantees the existence of a solution to the problem (1.26). A wide range of problems in mathematics, physics, statistics, signal processing, imaging, etc., can be modeled in the form of (1.26), such as flow of a viscous plastic fluid in a pipe [67], elasto-plastic problems [78], PDE-constrained optimization problems [125], obstacle problems [73], simplified friction problems, and image restoration problems [76]. Numerical methods for solving the problem (1.26) can be found in e.g., [67, 73, 76].

Here, we focus on the application of the alternating direction method of multipliers (ADMM) to the solution of (1.26). For this purpose, we introduce an auxiliary variable $q \in H$ satisfying $q = Bv$, then it is easy to see that problem (1.26) is equivalent to the following optimization problem with linear constraints and a separable objective functional:

$$\begin{cases} \min\limits_{(v,q)\in V\times H} & F(q) + G(v) \\ \text{s.t.} & q = Bv. \end{cases} \tag{1.27}$$

Let $\beta > 0$ be a penalty parameter, then the augmented Lagrangian associated with (1.27) can be defined by

$$L_\beta(v, q; \mu) = F(q) + G(v) + (\mu, Bv - q) + \frac{\beta}{2}\|Bv - q\|^2,$$

where $(\cdot, \cdot)$ and $\|\cdot\|$ are the inner product and norm defined on the space H, respectively. The corresponding saddle point problem reads: find $\{u, p; \lambda\} \in V \times H \times H$ such that

$$L_\beta(u, p; \mu) \leq L_\beta(u, p; \lambda) \leq L_\beta(v, q; \lambda),\ \forall\{v, q; \mu\} \in V \times H \times H. \tag{1.28}$$

Generally, the existence of a solution for problem (1.27) does not imply that the saddle point problem (1.28) has a solution and such an existence result has to be

verified in each specific case. Despite of this fact, one can show that the converse holds, see e.g., [80].

Theorem 1.1. *Suppose that* $\{u, p; \lambda\} \in V \times H \times H$ *is a saddle point of the augmented Lagrangian functional* L_β*, then* $\{u, p\}$ *is a solution to the problem (1.27) and* $u = Bp$*.*

To compute such saddle points, the augmented Lagrangian method (ALM) (proposed by H. Hestenes [100] and M. Powell [148] individually) reads as:

$$\begin{cases} \{u^{n+1}, p^{n+1}\} = \arg \min\limits_{v \in V, q \in H} L_\beta(v, q; \lambda^n), & \text{(1.29a)} \\ \lambda^{n+1} = \lambda^n + \beta(Bu^{n+1} - p^{n+1}). & \text{(1.29b)} \end{cases}$$

The convergence of ALM in infinite dimensional settings has been proved in e.g., [67, 73, 76, 80]. Concerning the implementation of ALM, the main difficulty is usually the solution of the minimization problem (1.29a), mainly because the variables v and q are coupled together. To address this issue, a natural idea is to utilize the separable structure of the problem (1.27) and treat the variables v and q individually. As a consequence, we obtain the following ADMM algorithm, where the minimization problem (1.29a) is decomposed into two parts and they are solved in the Gauss-Seidel manner.

$$\begin{cases} p^{n+1} = \arg \min\limits_{q \in H} L_\beta(u^n, q; \lambda^n), & \text{(1.30a)} \\ u^{n+1} = \arg \min\limits_{v \in V} L_\beta(v, p^{n+1}; \lambda^n), & \text{(1.30b)} \\ \lambda^{n+1} = \lambda^n + \beta(Bu^{n+1} - p^{n+1}). & \text{(1.30c)} \end{cases}$$

The ADMM was first proposed by R. Glowinski and A. Marrocco [72] for solving nonlinear elliptic equations; its convergence and convergence rate have been intensively studied in the literature, see e.g.,[19, 67, 73, 76, 80, 95, 97, 130]. A key feature of the ADMM is that the decomposed subproblems (1.30a) and (1.30b) are usually much easier than the ALM subproblem (1.29a), which makes the ADMM a benchmark algorithm in various areas such as image processing, statistical learning, data mining, and so on; we refer to [19, 53, 77] for some review papers on the ADMM.

We note that for some specific cases, the subproblems (1.30a) and (1.30b) are simple enough to have closed-form solutions. However, generally, the subproblems (1.30a) and/or (1.30b) can only be solved iteratively and inexactly when

implementing the ADMM. In this regard, various inexact versions of the ADMM in different settings can be found in the literature. For example, inexact versions of the ADMM for the generic case have been discussed in [52, 54, 137, 189]. These works require summable conditions on the sequence of accuracy (represented in terms of either the absolute or relative errors). The proximal ADMM, which adds appropriate quadratic terms to regularize the subproblems and may alleviate these subproblems for some cases by specifying the proximal terms appropriately, has been studied in, e.g., [22, 92].

Finally, we remark that the ADMM is closely related to the Douglas-Rachford splitting method which was first studied by Douglas and Rachford in [51] for heat conduction equations and then generalized by Lions and Mercier in [127] to nonlinear cases. In particular, the ADMM applied to the problem (1.27) is equivalent to the Douglas-Rachford splitting method applied to the dual problem of (1.27) with a proper time step size and an initial value; we refer to e.g., [38, 80] for the detailed discussions.

1.2.4 Primal-dual hybrid gradient methods

Introducing now an auxiliary variable $\mu \in H$ and applying the standard Fenchel-Rockafellar duality (see e.g., [55, Chapter VII]), we can show that (1.27) is equivalent to the following saddle point problem:

$$\min_{v\in V}\max_{\mu\in H} G(v) + (\mu, Bv) - F^*(\mu), \tag{1.31}$$

where $F^*(\mu) := \sup_{q\in H}(q,\mu) - F(q)$ is the convex conjugate of $F(q)$. Clearly, the problem (1.31) is a primal-dual formulation of (1.27). To solve such a saddle point problem numerically, the following primal-dual hybrid gradient (PDHG) method was proposed in [37].

$$\begin{cases} u^{n+1} = \arg\min_{v\in V}\{G(v) + (\lambda^n, Bv) + \dfrac{1}{2r}\|v-u^n\|^2\}, & \text{(1.32a)}\\ \bar{u}^n = u^{n+1} + \tau(u^{n+1}-u^n), & \text{(1.32b)}\\ \lambda^{n+1} = \arg\max_{\mu\in H}\{(\mu, B\bar{u}^n) - F^*(\mu) - \dfrac{1}{2s}\|\mu-\lambda^n\|^2\}, & \text{(1.32c)} \end{cases}$$

where $\tau \in [0,1]$ is the combination parameter and $r, s > 0$ are step sizes of the primal and dual step, respectively. The PDHG method does not require specific

initial iterates; its subproblems are usually much easier than the original model for some concrete applications; and its implementation is very easy. All these advantages make the PDHG very competitive with other kind of methods; and it has been widely used in various areas such as image processing [37, 60, 192], statistical learning [85], and inverse problems [8, 47, 168, 169, 170, 178].

The convergence and a worst-case $O(1/k)$ convergence rate measured by the iteration complexity of the PDHG method (1.32) with $\tau = 1$ have been analyzed in [37]. For the case of $\tau = 0$, its convergence was established in [60] under some asymptotic conditions on the step size sequences. When the step sizes are fixed, it has been shown in [94] by a simple counterexample that the PDHG method (1.32) with $\tau = 0$ is not necessarily convergent even with tiny constant step sizes. Hence, some additional conditions are required to guarantee the convergence of this special case. For instance, it was proved that the PDHG (1.32) with $\tau = 0$ and constant step sizes is convergent if one function in the model (1.31) is strongly convex. With this additional condition, a worst-case $O(1/k)$ convergence rate measured by the iteration complexity was also established therein.

It is worth mentioning that, as discussed in [37], the PDHG method (1.32) is closely related to some well-known numerical methods, such as the extrapolational gradient method [113], the Douglas-Rachford splitting method [51], and the alternating direction method of multipliers [72]. In particular, when $\tau = 0$ in (1.32b), the the PDHG method (1.32) corresponds to the Arrow-Hurwicz algorithm [2], which has been studied in [193] for total variation image restoration problems. Additionally, it has been shown in [95] that the PDHG method (1.32) with $\tau = 1$ is essentially an application of the proximal point algorithm [153].

Additionally, we note that some variants of (1.32) have been proposed in the literature to improve the numerical efficiency and alleviate the restrictions on the parameters in (1.32), see e.g., [85, 93, 94, 95, 168, 170]. In particular, for the case of $\tau \in [0, 1]$ in (1.32b), an algorithmic framework of generalized PDHG schemes was proposed in [93], which allows the output of the PDHG subroutine to be further updated by correction steps with constant step sizes. With different choices of parameters, some generalized PDHG schemes can be specified from the algorithmic framework and they are usually more efficient than the classical one

(1.32). In [95], some PDHG-based prediction-correction schemes are proposed and the combination parameter τ can be relaxed to $[-1, 1]$ which is broader than that in (1.32).

1.3 Outline of the thesis

The remaining part of this thesis is organized as follows.

In Chapter 2, we consider a class of nonlinear saddle point problems in the form of (1.6) and propose an algorithmic framework based on some inexact Uzawa methods in the literature. Under mild conditions, the convergence of this algorithmic framework is uniformly proved and the linear convergence rate is estimated. By choosing application-tailored preconditioners, we specify an efficient algorithm by the algorithmic framework for solving the elliptic optimal control problem with control constraints (1.2)-(1.3). The resulting algorithm does not need to solve any optimization subproblems or systems of linear equations in its iteration; each of its iterations only requires the projection onto a simple admissible set, four algebraic multi-grid V-cycles and a few matrix-vector multiplications. Its numerical efficiency is then demonstrated by some preliminary numerical results.

In Chapter 3, we focus on the implementation of the well-known alternating direction method of multipliers (ADMM) to the parabolic optimal control problem with control constraints (1.9)–(1.10). The ADMM decouples the control constraint and the parabolic state equation at each iteration. As a result, the main computation of each ADMM iteration is for solving an unconstrained parabolic optimal control subproblem. Because of its inevitably high dimensionality after the space-time discretization, the parabolic optimal control subproblems have to be solved iteratively and inexactly. Hence, the implementation of the ADMM must be embedded by an internal iterative process for solving these parabolic optimal control subproblems. We propose an easily implementable inexactness criterion for these subproblems; and obtain an inexact version of the ADMM whose execution consists of two-layer nested iterations. The strong global convergence

of the resulting inexact ADMM is proved rigorously in an infinite-dimensional Hilbert space; and the worst-case convergence rate measured by the iteration complexity is also established. We illustrate by the CG method how to execute the inexactness criterion, and show the efficiency of the resulting ADMM–CG iterative scheme numerically. Additionally, we consider an optimal control problem constrained by the wave equation with control constraints to show that our philosophy in algorithmic design can be easily extended to other optimal control problems and hence the proposed inexact ADMM can be deliberately specified as various algorithms for a wide range of optimal control problems.

In Chapter 4, we consider the sparse initial source identification for diffusion systems. It is well-known that such a problem is exponentially ill-posed because of the strong smoothing property of diffusion systems. Computationally, we propose an efficient optimal control based two-stage numerical approach. First, we formulate the sparse initial source identification problem as an optimal control problem with $L^2 + L^1$-regularized functional, where the L^1-term detects the sparsity of the initial sources and the L^2-term guarantees the well-posedness of the problem while avoiding numerical ill-conditioning problems. Then, we consider a structure enhancement stage, which consists of solving two simple and low-dimensional optimization problems in terms of the spatial variable and the intensities, to identify the locations and intensities of the sources, respectively. To solve the resulting optimal control problem, we advocate the well-known Primal-Dual Hybrid Gradient (PDHG) method. The PDHG method is cheap and easy to implement, as it decouples the optimal control problem into two simpler subproblems and only requires solving two PDEs at each iteration. To further improve the numerical efficiency of the PDHG method, we introduce a generalized PDHG-based prediction-correction algorithmic framework and prove its convergence rigorously. The efficiency of the proposed approach is compared with other optimization procedures based on the Gradient Descent methodology, and validated through several numerical results.

In Chapter 5, we consider the bilinear optimal control problem (BCP). Such a problem is generally challenging from both theoretical analysis and algorithmic design perspectives mainly because the state variable depends nonlinearly on

the control variable and an additional divergence-free constraint on the control is coupled together with the state equation. Mathematically, the proof of the existence of optimal solutions is delicate, and up to now, only some results are known for a few special cases where additional restrictions are imposed on the space dimension and the regularity of the control. We prove the existence of optimal controls and derive the first-order optimality conditions in general settings without any extra assumption. Computationally, the well-known conjugate gradient (CG) method can be applied conceptually. However, due to the additional divergence-free constraint on the control variable and the nonlinear relation between the state and control variables, it is challenging to compute the gradient and the optimal stepsize at each CG iteration, and thus nontrivial to implement the CG method. To address these issues, we advocate a fast inner preconditioned CG method to ensure the divergence-free constraint and an efficient inexactness strategy to determine an appropriate stepsize. An easily implementable nested CG method is thus proposed for solving such a complicated problem. Efficiency of the proposed nested CG method is promisingly validated by the results of some preliminary numerical experiments

Chapter 2

An Inexact Uzawa Algorithmic Framework for Nonlinear Saddle Point Problems with Applications to Elliptic Optimal Control Problem

Motivated by (1.6), we consider a class of nonlinear saddle point problems in the form of

$$\begin{cases} 0 \in (A + \mathcal{G})(w) + B^\top v - f, \\ 0 = Bw - g, \end{cases} \tag{2.1}$$

where H_1 and H_2 are two finite dimensional Hilbert spaces, $f \in H_1$ and $g \in H_2$ are given, $A : H_1 \to H_1$ is a linear, symmetric, and positive definite operator, $B : H_1 \to H_2$ is a linear operator and $B^\top : H_2 \to H_1$ is the adjoint operator of B, the operator $\mathcal{G} : H_1 \to 2^{H_1}$ is maximal monotone and we use the notation $(A + \mathcal{G})(w) := \{Aw + \nu | \nu \in \mathcal{G}(w)\}$. Throughout, we also assume the LBB condition (1.21) which ensures that the solution set of (2.1) is nonempty yet not unique, as to be shown in Lemma 2.1.

Besides the elliptic optimal control problem with control constraints (1.2)-(1.3), the consideration of (2.1) is also strongly motivated by a number of appli-

cations, such as the obstacle problem [68], the flow of a viscous plastic fluid in a pipe [78], the elasto-plastic problem [73] and the Cahn-Hilliard equation with an obstacle potential [86]. For these applications, $\mathcal{G}$ is usually the subdifferential of a nonsmooth convex function (e.g., the indicator function of a convex set) which is not required to be Lipschitz continuous. We refer to [118, 119] for more such applications.

2.1 An inexact Uzawa algorithmic framework

2.1.1 Algorithmic framework

To solve the nonlinear saddle point problem (2.1), it is natural to consider extending the framework of inexact Uzawa methods (1.23) which is applicable to the linear saddle point problem (1.20). The aforementioned advantages of (1.23) are important for tackling large-scale problems arising in the mentioned application areas. To discern the detail of this extension, let us introduce two symmetric positive definite operators $Q_A : H_1 \to H_1$ and $Q_B : H_2 \to H_2$ such that $Q_A - A$ and $Q_B - S$ are symmetric and positive semi-definite, and rewrite (1.23) as

$$\begin{pmatrix} Q_A - A & -B^\top \\ 0 & -Q_B \end{pmatrix} \begin{pmatrix} w^k \\ v^k \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} Q_A w^{k+1} \\ Bw^{k+1} - Q_B v^{k+1} \end{pmatrix}. \tag{2.2}$$

It is obvious that the linear saddle point problem (1.20) can be represented as

$$\begin{pmatrix} Q_A - A & -B^\top \\ 0 & -Q_B \end{pmatrix} \begin{pmatrix} w \\ v \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} Q_A w \\ Bw - Q_B v \end{pmatrix}. \tag{2.3}$$

In addition, we rewrite (2.1) as

$$\begin{pmatrix} Q_A - A & -B^\top \\ 0 & -Q_B \end{pmatrix} \begin{pmatrix} w \\ v \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix} \in \begin{pmatrix} (\mathcal{G} + Q_A)(w) \\ Bw - Q_B v \end{pmatrix}. \tag{2.4}$$

Then, comparing the model's extension from (2.3) to (2.4), it is natural to think of replacing Q_A in the right-hand side of (2.2) with $(\mathcal{G} + Q_A)$ and to use the

resulting iterative scheme for (2.4):

$$\begin{pmatrix} Q_A - A & -B^\top \\ 0 & -Q_B \end{pmatrix} \begin{pmatrix} w^k \\ v^k \end{pmatrix} + \begin{pmatrix} f \\ g \end{pmatrix} \in \begin{pmatrix} (\mathcal{G} + Q_A)(w^{k+1}) \\ Bw^{k+1} - Q_B v^{k+1} \end{pmatrix}. \tag{2.5}$$

We rewrite (2.5) as

$$\begin{cases} w^{k+1} = (Q_A + \mathcal{G})^{-1}(Q_A w^k - Aw^k - B^\top v^k + f), \\ v^{k+1} = v^k + Q_B^{-1}(Bw^{k+1} - g), \end{cases} \tag{2.6}$$

and this is our new inexact Uzawa algorithmic framework for the nonlinear saddle point problem (2.1).

Note that the operator $(Q_A+\mathcal{G})^{-1}$ in (2.6) is single-valued because Q_A is positive definite and $\mathcal{G}$ is maximal monotone. Obviously, the algorithmic framework (2.6) reduces to the framework of inexact Uzawa methods (1.23) if the linear saddle point problem (1.20) is considered, i.e., when $\mathcal{G} \equiv \mathbf{0}$. Similar as (1.23), Q_A and Q_B play the role of preconditioners of A and S, respectively. All advantageous features of (1.23) for better numerical performance are also inherited in (2.6).

In addition, as for the linear saddle point problem (1.20), if $Q_A = A$ and $Q_B = \frac{1}{\omega} I (\omega > 0)$ in (2.6), then the resulting algorithm reduces to

$$\begin{cases} w^{k+1} = (A + \mathcal{G})^{-1}(-B^\top v^k + f), \\ v^{k+1} = v^k + \omega(Bw^{k+1} - g). \end{cases} \tag{2.7}$$

As (1.22), we also call (2.7) the exact Uzawa method for the nonlinear saddle point problem (2.1). In fact, it can be shown that some applications of the classic augmented Lagrangian method proposed in [100, 148] to, e.g., a class of elliptic variational inequalities of the second kind in [73], are special cases of the exact Uzawa method (2.7). Finally, for the special case where $\mathcal{G}$ is the sum of the subdifferential of a nonsmooth convex function and a skew symmetric matrix or a positive definite operator, the algorithmic framework (2.6) is reduced to the Arrow-Hurwitz-type method in [118] (subject to a difference of constants associated with the preconditioners Q_A and Q_B) and the preconditioned Uzawa method in [119] (with $Q_A = A$ in (2.6)), respectively. Note that the discussion on how to choose the preconditioners Q_A and Q_B in [118] is mainly theoretical,

while we focus on finding appropriate application-tailored preconditioners so as to specify the inexact Uzawa algorithmic framework (2.6) as implementable and efficient algorithms for given specific applications.

2.1.2 Our objectives

The first purpose is to study the convergence for the inexact Uzawa algorithmic framework (2.6) in the generic setting of the nonlinear saddle point problem (2.1). We shall prove the convergence and estimate the linear convergence rate for the sequence generate by (2.6) under the same condition for the easier case (1.23) in [21], i.e.,

$$\text{(Convergence Condition)} \qquad Q_A \succeq A \quad \text{and} \quad Q_B \succeq S. \tag{2.8}$$

Hereafter, by $A_1 \succeq A_2$ (*resp.*, $A_1 \succ A_2$), we mean $A_1 - A_2$ is symmetric and positive semidefinite (*resp.*, definite), for two given symmetric positive definite operators A_1 and A_2.

Another purpose of this chapter is to investigate how to specify the inexact Uzawa algorithmic framework (2.6) as an efficient algorithm when it is applied to the elliptic optimal control problem with control constraints (1.2)-(1.3). The key is choosing appropriate preconditioners Q_A and Q_B; we shall show how to employ the algebraic multi-grid (AMG) V-cycle techniques in [63, 185] to find high-quality preconditioners for this specific application. As a result, an efficient algorithm is derived for solving the elliptic optimal control problem with control constraints (1.2)-(1.3).

2.2 Convergence

In this section, we prove the convergence of the sequence $\{\eta^k\}$ generated by the inexact Uzawa algorithmic framework (2.6) under the convergence condition (2.8).

Despite that the algorithmic framework (2.6) is intrinsically inspired by (1.23), theoretical analysis for the convergence and linear convergence rate of (2.6) is generally more difficult because of the presence of the maximal monotone operator $\mathcal{G}$. More specifically, convergence analysis for the linear case (1.23) is critically based on estimating the bound of the spectral radius of the corresponding iterative matrix (after representing the iterative scheme as a fixed point form), while this technique is not available for the nonlinear case (2.1). In addition, convergence results in the mentioned literature [40, 106] for the generic nonlinear saddle point problem (1.24) are not applicable either, because $\mathcal{G}$ in (2.1) is not necessarily Lipschitz continuous. On the other hand, it is not difficult to extend the analysis in [118] to the general case (2.1) and establish the convergence for the algorithmic framework (2.6). Here, we give a new proof for the convergence by analyzing the strict contraction property of the sequence generated by (2.6). This new perspective is also preparatory for proving the linear convergence rate in Section 2.3.

First, we summarize some useful results and present some assumptions for further analysis. We denote by $\eta \in H_1 \times H_2$, $Q : H_1 \times H_2 \to H_1 \times H_2$ and $\varphi : H_1 \times H_2 \to 2^{H_1 \times H_2}$, respectively, as the following:

$$\eta := \begin{pmatrix} w \\ v \end{pmatrix}, \; Q := \begin{pmatrix} Q_A - A & 0 \\ 0 & Q_B \end{pmatrix} \text{ and } \varphi(\eta) := \begin{pmatrix} \mathcal{G}(w) + Aw + B^\top v - f \\ -Bw + g \end{pmatrix}. \tag{2.9}$$

Obvious, $\varphi^{-1}(0)$ gives the solution set of (2.1) and we denote $\mathcal{S} := \varphi^{-1}(0)$.

Lemma 2.1. *The solution set $\mathcal{S}$ of (2.1) is nonempty.*

Proof. According to the LBB condition (1.21), the operator B is surjective. Thus, there exists a $w_0 \in H_1$ such that $Bw_0 = g$. If B is injective, it is obvious that we have $w_0 = B^{-1}g$ and there exists a $v^* \in H_2$ such that

$$0 \in \mathcal{G}(B^{-1}g) + AB^{-1}g + B^\top v^* - f.$$

If B is not injective, then the kernel space $Ker(B)$ of B is nonempty.

Let $\{\psi_1, \psi_2, \ldots, \psi_{n_B}\} \in H_1$ be a basis of $Ker(B)$ and $\Psi := \begin{pmatrix} \psi_1, \psi_2, \ldots, \psi_{n_B} \end{pmatrix}$ the operator from $\mathbb{R}^{n_B}$ to H_1. Then, any $w \in H_1$ satisfying $Bw = g$ can be

represented as $\Psi\zeta + w_0$ for some $\zeta \in \mathbb{R}^{n_B}$. Since A is positive definite and $\mathcal{G}$ is maximal monotone, it is easy to verify that the operator $\Psi^\top A\Psi$ is positive definite and the operator $\Psi^\top\mathcal{G}(\Psi\cdot + w_0)$ is maximal monotone. This means there must exist one and only one $\zeta^* \in \mathbb{R}^{n_B}$ such that

$$0 \in \Psi^\top\mathcal{G}(\Psi\zeta^* + w_0) + \Psi^\top A(\Psi\zeta^* + w_0) - \Psi^\top f.$$

Therefore, there exists $h^* \in \mathcal{G}(\Psi\zeta^* + w_0)$ such that $h^* + A(\Psi\zeta^* + w_0) - f$ is in the orthogonal complement space of $Ker(B)$, *i.e.*, the image space of $B^\top$ (see, e.g. [157]). Hence, there exists a $v^* \in H_2$ satisfying

$$0 = h^* + A(\Psi\zeta^* + w_0) + B^\top v^* - f.$$

We denote $w^* := \Psi\zeta^* + w_0$. Then, it follows from $Bw^* = g$ and $h^* \in \mathcal{G}(w^*)$ that

$$0 \in \begin{pmatrix} \mathcal{G}(w^*) + Aw^* + B^\top v^* - f \\ -Bw^* + g \end{pmatrix},$$

which means $(w^*, v^*)^\top$ is a solution point of (2.1). Hence, the solution set $\mathcal{S}$ of (2.1) is nonempty. □

Remark 2.1. *According to the proof above, we notice that the components w of all solution points of (2.1) are indeed identical because of the uniqueness of ζ^*, despite that the other component v is not necessarily unique (which is due to the presence of $\mathcal{G}$).*

Throughout, we require the preconditioners Q_A and Q_B in (2.6) to satisfy the convergence condition (2.8). With this condition, we know $Q \succeq 0$. Thus, we denote by $\|\eta\|_Q = \sqrt{\eta^\top Q\eta}$ the semi norm and it holds that

$$\|\eta\|_Q \leq \sqrt{\rho(Q)}\|\eta\|. \tag{2.10}$$

In addition, we define $dist(\iota, \mathcal{D})$ and $dist_Q(\iota, \mathcal{D})$, respectively, as

$$dist(\iota, \mathcal{D}) := \inf_{\xi\in\mathcal{D}} \|\iota - \xi\| \quad \text{and} \quad dist_Q(\iota, \mathcal{D}) := \inf_{\xi\in\mathcal{D}} \|\iota - \xi\|_Q,$$

for a given subset $\mathcal{D}$ and ι in the same space. Then it follows from (2.10) that

$$dist_Q(\iota, \mathcal{D}) \leq \sqrt{\rho(Q)} \cdot dist(\iota, \mathcal{D}). \tag{2.11}$$

Next, we prove two useful lemmas.

Lemma 2.2. *For A and B in (2.1), if the preconditioners Q_A and Q_B satisfy the convergence condition (2.8), then it holds that*

$$A - B^\top Q_B^{-1} B \succeq 0.$$

Proof. It follows from the LBB condition (1.21) that the Schur complement $S = BA^{-1}B^\top$ is positive definite and thus it is invertible. Since we have $Q_B \succeq S$ from the convergence condition (2.8), it holds that

$$B^\top S^{-1} B \succeq B^\top Q_B^{-1} B. \tag{2.12}$$

Therefore, it suffices to show $A - B^\top S^{-1} B \succeq 0$. For this purpose, we introduce $\Pi := I - A^{-1}B^\top S^{-1}B$. It is easy to show that $\Pi^2 = \Pi$ and $A\Pi = \Pi^\top A$. Therefore, we have

$$\langle A\Pi x, x\rangle = \langle A\Pi^2 x, x\rangle = \langle \Pi^\top A\Pi x, x\rangle = \langle A\Pi x, \Pi x\rangle \geq 0, \quad \forall x \in H_1.$$

We thus obtain that $A - B^\top S^{-1}B = A\Pi \succeq 0$ and the desired result follows from (2.12) directly. □

Lemma 2.3. *There exists a constant $L_1 > 0$ such that*

$$\begin{aligned} &\|(Q_A + \mathcal{G})^{-1}(Q_A w_1 - Aw_1 - B^\top v_1 + f) \\ &\qquad - (Q_A + \mathcal{G})^{-1}(Q_A w_2 - Aw_2 - B^\top v_2 + f)\|_{Q_A} \\ \leq & L_1 \|\eta_1 - \eta_2\|_Q, \forall \eta_1, \eta_2 \in H_1 \times H_2. \end{aligned} \tag{2.13}$$

Proof. To verify this inequality, we first need to prove the following inequality:

$$\|(Q_A + \mathcal{G})^{-1} h_1 - (Q_A + \mathcal{G})^{-1} h_2\|_{Q_A} \leq \|Q_A^{-1}(h_1 - h_2)\|_{Q_A}, \forall h_1, h_2 \in H_1. \tag{2.14}$$

For this purpose, let

$$u_1 = (Q_A + \mathcal{G})^{-1} h_1 \quad \text{and} \quad u_2 = (Q_A + \mathcal{G})^{-1} h_2, \forall h_1, h_2 \in H_1.$$

Then, we have

$$-Q_A u_1 + h_1 \in \mathcal{G}(u_1) \quad \text{and} \quad -Q_A u_2 + h_2 \in \mathcal{G}(u_2).$$

Since the operator $\mathcal{G}$ is monotone, we have

$$\langle -Q_A(u_1 - u_2) + (h_1 - h_2), u_1 - u_2\rangle \geq 0.$$

We then obtain

$$\|u_1-u_2\|_{Q_A}^2 \le \langle Q_A^{-1}(h_1-h_2), u_1-u_2\rangle_{Q_A} \le \|Q_A^{-1}(h_1-h_2)\|_{Q_A}\|u_1-u_2\|_{Q_A},$$

which implies (2.14) directly.

Moreover, let $h_i = Q_A w_i - Aw_i - B^\top v_i + f \in H_1$, for $i = 1, 2$. Then, it follows from (2.14) that

$$\begin{aligned}
&\|(Q_A+\mathcal{G})^{-1}(Q_A w_1 - Aw_1 - B^\top v_1 + f) - (Q_A+\mathcal{G})^{-1}(Q_A w_2 - Aw_2 - B^\top v_2 + f)\|_{Q_A}\\
\le&\|Q_A^{-1}((Q_A - A)(w_1-w_2) - B^\top(v_1-v_2))\|_{Q_A}\\
\le&\|Q_A^{-1}(Q_A - A)(w_1-w_2)\|_{Q_A} + \|Q_A^{-1}B^\top(v_1-v_2))\|_{Q_A}\\
\le&\Big(\sqrt{\|(Q_A-A)^{\top/2}Q_A^{-1}(Q_A-A)^{1/2}\|} + \sqrt{\|Q_B^{-\top/2}BQ_A^{-1}B^\top Q_B^{-1/2}\|}\Big)\|\eta_1-\eta_2\|_Q.
\end{aligned}$$

If we set

$$L_1 = \sqrt{\|(Q_A-A)^{\top/2}Q_A^{-1}(Q_A-A)^{1/2}\|} + \sqrt{\|Q_B^{-\top/2}BQ_A^{-1}B^\top Q_B^{-1/2}\|},$$

the assertion (2.13) is proved. □

Next, we prove that the sequence $\{\eta^k\}$ generated by the inexact Uzawa algorithmic framework (2.6) is contractive with respect to the solution set $\mathcal{S}$. The contraction property is crucial for establishing the convergence of the sequence.

Theorem 2.1. *Let $\eta^* = (w^*, v^*)^\top \in \mathcal{S}$, that is,*

$$\begin{cases} 0 \in (A+\mathcal{G})(w^*) + B^\top v^* - f, \\ 0 = Bw^* - g, \end{cases} \tag{2.15}$$

and $\{\eta^k = (w^k, v^k)^\top\}$ be the sequence generated by the inexact Uzawa algorithmic framework (2.6). With the convergence condition (2.8), we have

$$\|\eta^{k+1} - \eta^*\|_Q^2 \le \|\eta^k - \eta^*\|_Q^2 - \|\eta^{k+1} - \eta^k\|_Q^2. \tag{2.16}$$

Proof. From (2.6), it is easy to verify that

$$\begin{cases} Q_A(w^k - w^{k+1}) - Aw^k - B^\top v^k + f \in \mathcal{G}(w^{k+1}), & (2.17)\\ Q_B(v^k - v^{k+1}) + Bw^{k+1} - g = 0, & (2.18) \end{cases}$$

and (2.15) can be written as

$$\begin{cases} -Aw^* - B^\top v^* + f \in \mathcal{G}(w^*), & (2.19)\\ Bw^* - g = 0. & (2.20) \end{cases}$$

Subtracting (2.19) and (2.20) from (2.17) and (2.18) respectively, and using the maximal monotonicity of $\mathcal{G}$, we obtain

$$\langle (Q_A - A)(w^k - w^{k+1}) - A(w^{k+1} - w^*) - B^\top (v^k - v^*), w^{k+1} - w^* \rangle \geq 0, \quad (2.21)$$

$$Q_B(v^k - v^{k+1}) + B(w^{k+1} - w^*) = 0. \quad (2.22)$$

Then, we derive that

$$\begin{aligned}
&\langle Q(\eta^k - \eta^{k+1}), \eta^{k+1} - \eta^* \rangle \\
&= \langle (Q_A - A)(w^k - w^{k+1}), w^{k+1} - w^* \rangle + \langle Q_B(v^k - v^{k+1}), v^{k+1} - v^* \rangle \\
&\overset{(2.21)}{\geq} \langle A(w^{k+1} - w^*), w^{k+1} - w^* \rangle + \langle B^\top (v^k - v^*), w^{k+1} - w^* \rangle \\
&\quad + \langle Q_B(v^k - v^{k+1}), v^{k+1} - v^* \rangle \\
&\overset{(2.22)}{=} \langle A(w^{k+1} - w^*), w^{k+1} - w^* \rangle - \langle v^k - v^*, Q_B(v^k - v^{k+1}) \rangle \\
&\quad + \langle Q_B(v^k - v^{k+1}), v^{k+1} - v^* \rangle \\
&= \langle A(w^{k+1} - w^*), w^{k+1} - w^* \rangle - \langle v^k - v^{k+1}, Q_B(v^k - v^{k+1}) \rangle \\
&\overset{(2.22)}{=} \langle A(w^{k+1} - w^*), w^{k+1} - w^* \rangle - \langle B^\top Q_B^{-1} B(w^{k+1} - w^*), w^{k+1} - w^* \rangle \\
&= \langle (A - B^\top Q_B^{-1} B)(w^{k+1} - w^*), w^{k+1} - w^* \rangle.
\end{aligned} \quad (2.23)$$

It follows from Lemmas 2.2 and (2.23) that

$$\langle Q(\eta^k - \eta^{k+1}), \eta^{k+1} - \eta^* \rangle \geq 0,$$

which implies

$$\|\eta^{k+1} - \eta^*\|_Q^2 \leq \|\eta^k - \eta^*\|_Q^2 - \|\eta^{k+1} - \eta^k\|_Q^2,$$

and we complete the proof. □

With these preparations, we are now able to prove the convergence of the inexact Uzawa algorithmic framework (2.6).

Theorem 2.2. *Let $\{\eta^k = (w^k, v^k)^\top\}$ be the sequence generated by the inexact Uzawa algorithmic framework (2.6). Then, we have*

$$\|w^k - w^*\| \to 0 \quad \textit{and} \quad \textit{dist}(v^k, \mathcal{S}_v) \to 0,$$

where $\mathcal{S}_v := \{v^* | (w^*, v^*) \in \mathcal{S}\}$.

Proof. First, we need to show that the sequence $\{\eta^k\}$ is bounded. It follows from (2.16) that, for any integer $K > 0$ and $\eta^* \in \mathcal{S}$, we have

$$\|\eta^K - \eta^*\|_Q^2 + \sum_{k=0}^{K-1} \|\eta^{k+1} - \eta^k\|_Q^2 \leq \|\eta^0 - \eta^*\|_Q^2,$$

which implies that $\|\eta^{k+1} - \eta^*\|_Q^2$ is bounded and that

$$\lim_{k\to\infty} \|\eta^{k+1} - \eta^k\|_Q^2 = 0.$$

As a result, $\|v^k - v^*\|_{Q_B}$ is bounded. It is clear that $\|w^{k+1} - w^*\|_{Q_A}$ is also bounded. Indeed, we have

$$\begin{aligned} \|w^{k+1} - w^*\|_{Q_A} &= \|(Q_A + \mathcal{G})^{-1}(Q_A w^k - Aw^k - B^\top v^k + f) \\ &\quad - (Q_A + \mathcal{G})^{-1}(Q_A w^* - Aw^* - B^\top v^* + f)\|_{Q_A} \qquad (2.24) \\ &\overset{(2.13)}{\leq} L_1 \|\eta^k - \eta^*\|_Q. \end{aligned}$$

In addition, the boundedness of $\|\eta^k - \eta^*\|$ is guaranteed by the positive definiteness of both Q_A and Q_B. Therefore, the sequence $\{\eta^k\}$ is bounded.

To show that any cluster point of $\{\eta^k\}$ is a solution point of the problem (2.1), let η^∞ be a cluster point of $\{\eta^k\}$ with $\lim_{l\to\infty} \eta^{k_l} = \eta^\infty$. We first prove that $\eta^\infty \in \mathcal{S}$. Recall $\lim_{k\to\infty} \|\eta^{k+1} - \eta^k\|_Q^2 = 0$. We can derive that

$$\begin{aligned} \lim_{k\to\infty} \|w^{k+1} - w^k\|_{Q_A} &= \lim_{k\to\infty} \|(Q_A + \mathcal{G})^{-1}(Q_A w^k - Aw^k - B^\top v^k + f) \\ &\quad - (Q_A + \mathcal{G})^{-1}(Q_A w^{k-1} - Aw^{k-1} - B^\top v^{k-1} + f)\|_{Q_A} \\ &\overset{(2.13)}{\leq} \lim_{k\to\infty} L_1 \|\eta^k - \eta^{k-1}\|_Q = 0, \end{aligned}$$

and

$$\lim_{k\to\infty} \|v^{k+1} - v^k\|_{Q_B} = 0.$$

We thus have $\lim_{k\to\infty} \|\eta^{k+1} - \eta^k\| = 0$, which implies that

$$\lim_{l\to\infty} \|w^{k_l+1} - w^{k_l}\| = 0, \quad \text{and} \quad \lim_{l\to\infty} \|v^{k_l+1} - v^{k_l}\| = 0.$$

Then, one can derive that

$$
\begin{aligned}
&\|(Q_A+\mathcal{G})^{-1}(Q_Aw^\infty - Aw^\infty - B^\top v^\infty + f) - w^\infty\|_{Q_A} \\
&= \|(Q_A+\mathcal{G})^{-1}(Q_Aw^\infty - Aw^\infty - B^\top v^\infty + f) - \lim_{l\to\infty} w^{k_l}\|_{Q_A} \\
&\le \|(Q_A+\mathcal{G})^{-1}(Q_Aw^\infty - Aw^\infty - B^\top v^\infty + f) \\
&\qquad\qquad - \lim_{l\to\infty}(Q_A+\mathcal{G})^{-1}(Q_Aw^{k_l} - Aw^{k_l} - B^\top v^{k_l} + f)\|_{Q_A} \\
&\qquad + \|\lim_{l\to\infty}(Q_A+\mathcal{G})^{-1}(Q_Aw^{k_l} - Aw^{k_l} - B^\top v^{k_l} + f) - \lim_{l\to\infty} w^{k_l+1}\|_{Q_A} \\
&\qquad + \|\lim_{l\to\infty} w^{k_l+1} - \lim_{l\to\infty} w^{k_l}\|_{Q_A} \\
&= \lim_{l\to\infty}\|(Q_A+\mathcal{G})^{-1}(Q_Aw^\infty - Aw^\infty - B^\top v^\infty + f) \\
&\qquad\qquad - (Q_A+\mathcal{G})^{-1}(Q_Aw^{k_l} - Aw^{k_l} - B^\top v^{k_l} + f)\|_{Q_A} \\
&\overset{(2.13)}{\le} \lim_{l\to\infty} L_1\|\eta^\infty - \eta^{k_l}\|_Q = 0,
\end{aligned}
$$

and

$$
\lim_{l\to\infty}\|Q_B^{-1}(Q_Bv^{k_l} + Bw^{k_l+1} - g) - v^{k_l+1}\| = \|Q_B^{-1}(Bw^\infty - g)\| = 0.
$$

Therefore, we conclude that:

$$
\begin{cases}
0 \in (A+\mathcal{G})w^\infty + B^\top v^\infty - f, \\
0 = Bw^\infty - g,
\end{cases}
$$

which implies that η^∞ is a solution point of the problem (2.1).

As mentioned in Remark 2.1, the components w of solution points of the nonlinear saddle point problem (2.1) are identical. We thus have $\|w^k - w^*\| \to 0$. It is easy to verify that $dist(v^k, \mathcal{S}_v) \to 0$. Otherwise, there should exist a subsequence $\{v^{k_l}\}$ of $\{v^k\}$ such that $dist(\eta^{k_l}, \mathcal{S}) \ge dist(v^{k_l}, \mathcal{S}_v) \ge \epsilon_0$ for some $\epsilon_0 > 0$. Let η^* be a cluster point of $\{\eta^{k_l}\}$. Then we have $dist(\eta^*, \mathcal{S}_v) \ge \epsilon_0$ which contradicts with the above conclusion of any cluster point of $\{\eta^k\}$ being a solution point of the problem (2.1). We thus complete the proof. □

Remark 2.2. *Compared with (2.17), (2.18) and (2.19), (2.20), it is easy to verify that η^{k+1} is a solution point of the problem (2.1) if and only if $\eta^{k+1} = \eta^k$. It also holds that $\lim_{k\to\infty}\|\eta^{k+1} - \eta^k\| = 0$. Therefore, given a tolerance $\epsilon > 0$, we can use $\|\eta^{k+1} - \eta^k\| \le \epsilon$ as a stopping criterion to numerically implement the inexact Uzawa algorithmic framework (2.6).*

2.3 Linear convergence rate

In this section, we further consider the linear convergence rate for the inexact Uzawa algorithmic framework (2.6). It turns out that some techniques in variational analysis, especially some sophisticated techniques recently initiated in [130, 182, 188] for convergence analysis of various algorithms in optimization, are useful for our analysis. It seems to be the first time to consider such variational analysis techniques to derive the linear convergence rates for inexact Uzawa type methods in the context of saddle point problems. More specifically, we shall derive the linear convergence rate of (2.6) under mild conditions which can be easily satisfied by various concrete applications of the nonlinear saddle point problem (2.1).

To discuss the linear convergence rate for (2.6), we make the following assumption.

Assumption 2.1. *The set-valued mapping φ defined in (2.9) is metrically subregular around $(\eta^*, 0)$, $\forall \eta^* \in \mathcal{S}$. That is, $\forall \eta^* \in \mathcal{S}$, there exist a neighborhood $\mathbb{B}_\epsilon(\eta^*)$ of η^* and $\kappa \geq 0$ such that*

$$dist(\eta, \mathcal{S}) \leq \kappa \cdot dist(0, \varphi(\eta)), \quad \forall \eta \in \mathbb{B}_\epsilon(\eta^*).$$

More details about the metric subregularity can be found in, e.g. [187].

We first prove a useful lemma which plays a key role in analyzing the linear convergence rate for (2.6).

Lemma 2.4. *Let $\{\eta^k\}$ be the sequence generated by the inexact Uzawa algorithmic framework (2.6). If Assumption 2.1 is satisfied, then there exist constants $K_0 > 0$ and $\bar{\kappa} \geq 0$ such that, for any $k \geq K_0$, it holds that*

$$dist(\eta^{k+1}, \mathcal{S}) \leq \bar{\kappa} \|\eta^{k+1} - \eta^k\|_Q.$$

Proof. Because of Theorem 2.2, we know that any cluster point of $\{\eta^k\}_{k=1}^\infty$ is a solution point of the problem (2.1). Moreover, the proof of Theorem 2.2 indicates that the set of the cluster points $\mathcal{S}_c$ is bounded. Let $\{\mathbb{B}_{\frac{\epsilon}{2}}(\eta^*)\}, \eta^* \in \mathcal{S}, \epsilon > 0$ be

an open covering of the bounded closed set $\bar{\mathcal{S}}_c$, i.e., the closure of $\mathcal{S}_c$. There exist finite open sets $\{\mathbb{B}_{\frac{\epsilon_i}{2}}(\eta_i^*)\}, \eta_i^* \in \mathcal{S}, i = 1, \ldots, i_0$ such that $\bar{\mathcal{S}}_c \subseteq \bigcup_{i=1,\ldots,i_0}\{\mathbb{B}_{\frac{\epsilon_i}{2}}(\eta_i^*)\}$. Hence, there exists $K_0 > 0$ such that, for any $k > K_0$, there is always one solution point $\eta_{j_k}^* \in \mathcal{S}, j_k \in \{i = 1, \ldots, i_0\}$ such that $\eta^{k+1} \in \mathbb{B}_{\epsilon_{j_k}}(\eta_{j_k}^*)$. Then, Assumption 2.1 shows that, for some $\kappa_{j_k} \geq 0$, we have

$$dist(\eta^{k+1}, \mathcal{S}) \leq \kappa_{j_k} \cdot dist(0, \varphi(\eta^{k+1})).$$

It follows from (2.6) and the definition of $\varphi(\eta)$ in (2.9) that

$$\begin{pmatrix} (Q_A - A)(w^k - w^{k+1}) \\ Q_B(v^k - v^{k+1}) \end{pmatrix} \in \varphi(\eta^{k+1}).$$

Then, we obtain

$$\begin{aligned} dist(\eta^{k+1}, \mathcal{S}) &\leq \kappa_{j_k} \cdot dist(0, \varphi(\eta^{k+1})) \\ &\leq \kappa_{j_k}(\|(Q_A - A)(w^k - w^{k+1})\| + \|Q_B(v^k - v^{k+1})\|) \\ &\leq \kappa_{j_k}(\sqrt{\rho(Q_A - A)} + \sqrt{\rho(Q_B)})\|\eta^{k+1} - \eta^k\|_Q \\ &\leq \left(\max_{i=1,\ldots,i_0}\{\kappa_i\}\right) \cdot (\sqrt{\rho(Q_A - A)} + \sqrt{\rho(Q_B)})\|\eta^{k+1} - \eta^k\|_Q. \end{aligned}$$

Therefore, with $\bar{\kappa} = (\max_{i=1,\ldots,i_0}\{\kappa_i\}) \cdot (\sqrt{\rho(Q_A - A)} + \sqrt{\rho(Q_B)}) \geq 0$, the result is proved. □

Next, we prove a local convergence property of the sequence $\{dist_Q^2(\eta^{k+1})\}$.

Theorem 2.3. *Let $\{\eta^k\}$ be the sequence generated by the inexact Uzawa algorithmic framework (2.6). If Assumption 2.1 is satisfied, then there exists $K_0 > 0$ such that, for any $k \geq K_0$, it holds that*

$$dist_Q^2(\eta^{k+1}, \mathcal{S}) \leq (1 + \frac{1}{\rho(Q)\bar{\kappa}^2})^{-1} dist_Q^2(\eta^k, \mathcal{S}).$$

Proof. From (2.16), we have

$$dist_Q^2(\eta^{k+1}, \mathcal{S}) \leq dist_Q^2(\eta^k, \mathcal{S}) - \|\eta^{k+1} - \eta^k\|_Q^2, \forall k \geq 0.$$

By virtue of the inequality (2.11) and Lemma 2.4, there exist $K_0 > 0$ and $\bar{\kappa} \geq 0$ such that

$$\frac{1}{\sqrt{\rho(Q)}} dist_Q(\eta^{k+1}, \mathcal{S}) \leq dist(\eta^{k+1}, \mathcal{S}) \leq \bar{\kappa}\|\eta^{k+1} - \eta^k\|_Q, \quad \forall k \geq K_0. \tag{2.25}$$

Consequently, we have

$$dist_Q^2(\eta^{k+1}, \mathcal{S}) \leq dist_Q^2(\eta^k, \mathcal{S}) - \frac{1}{\rho(Q)\bar{\kappa}^2} dist_Q^2(\eta^{k+1}, \mathcal{S}), \quad \forall k \geq K_0,$$

and the proof is complete. □

Now, we can prove a similar theorem concerning the same property but globally. Techniques in [62, 130] are useful for proving the theorem.

Theorem 2.4. *Let $\{\eta^k\}$ be the sequence generated by the inexact Uzawa algorithmic framework (2.6). If Assumption 2.1 is satisfied, then for all $k \geq 0$, there exists $\tilde{\kappa} > 0$ such that*

$$dist_Q^2(\eta^{k+1}, \mathcal{S}) \leq (1 + \frac{1}{\tilde{\kappa}^2})^{-1} dist_Q^2(\eta^k, \mathcal{S}). \tag{2.26}$$

Proof. First, it follows from (2.25) that

$$dist_Q(\eta^{k+1}, \mathcal{S}) \leq \sqrt{\rho(Q)}\bar{\kappa}\|\eta^{k+1} - \eta^k\|_Q, \quad \forall k \geq K_0.$$

Therefore, we only need to consider the case $k < K_0$. Let

$$\varepsilon := \min_{0 \leq k < K_0} \{\|\eta^{k+1} - \eta^k\|_Q\},$$

then we have

$$\|\eta^{k+1} - \eta^k\|_Q \geq \varepsilon, \quad \forall k < K_0.$$

Recall the contraction property of $\{\eta^k\}$ in (2.16). There exists a constant $C > 0$ such that $\|\eta^{k+1} - \eta^*\|_Q \leq C$. We thus have

$$dist_Q(\eta^{k+1}, \mathcal{S}) \leq \|\eta^{k+1} - \eta^*\|_Q \leq \frac{C}{\varepsilon}\|\eta^{k+1} - \eta^k\|_Q, \quad \forall k < K_0.$$

Let $\tilde{\kappa} := \max\{\sqrt{\rho(Q)}\bar{\kappa}, \frac{C}{\varepsilon}\}$. We have

$$dist_Q(\eta^{k+1}, \mathcal{S}) \leq \tilde{\kappa}\|\eta^{k+1} - \eta^k\|_Q, \quad \forall k \geq 0.$$

Taking (2.16) into account, we immediately obtain

$$dist_Q^2(\eta^{k+1}, \mathcal{S}) \leq (1 + \frac{1}{\tilde{\kappa}^2})^{-1} dist_Q^2(\eta^k, \mathcal{S}), \quad \forall k \geq 0,$$

and the proof is complete. □

Based on Theorem 2.4, the global linear convergence rate of the sequences $\{w^k\}$ and $\{v^k\}$ can be derived directly. We present these results in the following theorem.

Theorem 2.5. *Let $\{\eta^k\}$ be the sequence generated by the inexact Uzawa algorithmic framework (2.6). If Assumption 2.1 is satisfied, then for all $k \geq 0$, there exist $\tilde{\kappa} > 0$, $L_2 > 0$ and $L_3 > 0$ such that*

$$\|w^{k+1} - w^*\| \leq L_2(1 + \frac{1}{\tilde{\kappa}^2})^{-\frac{k}{2}} dist_Q(\eta^0, \mathcal{S}), \tag{2.27}$$

and

$$dist(v^{k+1}, \mathcal{S}_v) \leq L_3(1 + \frac{1}{\tilde{\kappa}^2})^{-\frac{k+1}{2}} dist_Q(\eta^0, \mathcal{S}), \tag{2.28}$$

where $\mathcal{S}_v = \{v^* | (w^*, v^*) \in \mathcal{S}\}$.

Proof. By setting $L_3 = \sqrt{\rho(Q_B^{-1})}$, the second inequality (2.28) follows from (2.26) directly. We then focus on the first inequality (2.27) in the following discussion.

According to (2.24), we have

$$\|w^{k+1} - w^*\| \leq \rho(Q_A^{-1})\|w^{k+1} - w^*\|_{Q_A} \leq \rho(Q_A^{-1})L_1\|\eta^k - \eta^*\|_Q.$$

Since it is shown in Lemma 2.1 that the components w of all solution points of (2.6) are identical, taking infimum with respect to η^* on the inequality above yields

$$\|w^{k+1} - w^*\| \leq \rho(Q_A^{-1})L_1 dist_Q(\eta^k, \mathcal{S}).$$

It follows from Theorem 2.4 that

$$dist_Q(\eta^k, \mathcal{S}) \leq (1 + \frac{1}{\tilde{\kappa}^2})^{-\frac{k}{2}} dist_Q(\eta^0, \mathcal{S}).$$

Thus we have

$$\|w^{k+1} - w^*\| \leq \rho(Q_A^{-1})L_1(1 + \frac{1}{\tilde{\kappa}^2})^{-\frac{k}{2}} dist_Q(\eta^0, \mathcal{S}).$$

The proof is complete with $L_2 = \rho(Q_A^{-1})L_1$. □

Note that Assumption 2.1 can be easily satisfied by many popular choices of the abstract operator $\mathcal{G}$. Examples include the subdifferential of a convex function such as the indicator function of a box constraint and the L^1-norm

function; see, e.g., [154] for more details. As a result, according to Theorem 2.5, the linear convergence rate holds for the resulting algorithms generated via the inexact Uzawa algorithmic framework (2.6) for some concrete applications including the elliptic optimal control problem with control constraints which will be discussed in the next section.

2.4 Application to elliptic optimal control problems

With the proved convergence and linear convergence rate, it is promising to consider the inexact Uzawa algorithmic framework (2.6) for various applications. Meanwhile, the highly abstract and general algorithmic framework (2.6) becomes practical only when the preconditioners Q_A and Q_B are chosen appropriately for a specific application of the nonlinear saddle point problem (2.1) in abstract form. To illustrate how to choose application-tailored preconditioners in (2.6) for a given application and thus derive an efficient algorithm via the framework (2.6), we consider an elliptic optimal control problem with control constraints, which is a fundamental problem in various areas (see, e.g., [18, 30, 49, 103, 125, 145, 171]).

2.4.1 Problem statement

We consider the elliptic optimal control problem with control constraints

$$\min_{y\in H_0^1(\Omega),u\in\mathcal{C}} J(y,u)=\frac{1}{2}\|y-y_d\|^2_{L^2(\Omega)}+\frac{\alpha}{2}\|u\|^2_{L^2(\Omega)}, \tag{2.29}$$

where y and u satisfy the following state equation:

$$\begin{cases} \mathcal{K}y=u \quad \text{in} \quad \Omega, \\ \quad y=0 \quad \text{on} \quad \Gamma. \end{cases} \tag{2.30}$$

In (2.30), $\Omega\subset\mathbb{R}^d (d\geq 1)$ is a convex polyhedral domain with boundary $\Gamma:=\partial\Omega$, and the desired state $y_d\in L^2(\Omega)$ is given. The admissible set $\mathcal{C}$ is defined by

$$\mathcal{C}=\{u\in L^\infty(\Omega)|a\leq u(x)\leq b, \text{ a.e. in } \Omega\}\subset L^2(\Omega),$$

where $-\infty < a < b < +\infty$ are two given constants. If $a = -\infty$ and $b = +\infty$, then the problem (2.29) reduces to the unconstrained case.

In the equation (2.30), the linear second-order elliptic operator $\mathcal{K}$ is defined by

$$\mathcal{K}y = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \sum_{j=1}^{d} a_{ij} \frac{\partial y}{\partial x_j} + c_0 y,$$

where $0 \leq c_0 \in L^\infty(\Omega), 0 < a_{ij} \in L^\infty(\Omega), \forall 1 \leq i, j \leq d$ are coefficients. In addition, we assume that the matrix-valued function $(a_{ij})_{1 \leq i,j \leq d}$ satisfies $a_{ij} = a_{ji}$ and

$$\sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij}(x) \xi_i \xi_j \geq \gamma \|\boldsymbol{\xi}\|^2, \quad \forall \boldsymbol{\xi} = \{\xi_i\}_{i=1}^{d} \in \mathbb{R}^d \quad \text{a.e. in } \Omega,$$

with $\gamma \geq 0$ and $\|\cdot\|$ the canonical Euclidean norm of $\mathbb{R}^d$. Under these assumptions, the bilinear form $a(\cdot, \cdot) : H_0^1 \times H_0^1 \to \mathbb{R}$ associated with $\mathcal{K}$ can be defined by

$$a(y, v) = \sum_{i=1}^{d} \sum_{j=1}^{d} \int_\Omega a_{ij}(x) \frac{\partial y}{\partial x_j} \frac{\partial v}{\partial x_i} dx + \int_\Omega c_0 y v dx. \tag{2.31}$$

The existence and uniqueness of the solution point to the problem (2.29) can be proved in a similar way as discussed in [125]; more details can be found in [103].

To characterize the solution point u^* of the problem (2.29), we introduce an adjoint variable p. It is then well known (e.g., see [103]) that the first-order optimality conditions of (2.29) can be stated as below.

Theorem 2.6. *Suppose that $u^* \in \mathcal{C}$ is the unique solution of the problem (2.29). Then the following first-order optimality conditions hold:*

$$0 \in \partial I_{\mathcal{C}}(u^*) + \alpha u^* + p^*, \tag{2.32}$$

$$\begin{cases} \mathcal{K}y^* = u^* \quad in \quad \Omega, \\ \quad y^* = 0 \quad on \quad \Gamma, \end{cases} \tag{2.33}$$

$$\begin{cases} \mathcal{K}^* p^* = y^* - y_d \quad in \quad \Omega, \\ \qquad\quad p^* = 0 \quad on \quad \Gamma, \end{cases} \tag{2.34}$$

where y^* and p^* are the state and adjoint variables associated with u^*, respectively, $I_{\mathcal{C}}(u)$ the indicator function of the admissible set $\mathcal{C}$ and $\partial I_{\mathcal{C}}(u)$ the subdifferential of $I_{\mathcal{C}}$ and $\mathcal{K}^*$ is the adjoint of the operator $\mathcal{K}$.

Since the problem (2.29) is convex, the optimality conditions (2.32)-(2.34) are also sufficient. Accordingly, we can solve (2.32)-(2.34) to obtain the optimal solution of (2.29).

2.4.2 Finite element discretization

In this subsection, we discuss the finite element discretization of the problem (2.29) in order to solve it numerically.

Let $\mathcal{T}_h$ be a quasi-uniform triangulation of $\bar{\Omega}$, and T be an element of $\mathcal{T}_h$ satisfying $\bar{\Omega}_h = \bigcup_{T\in\mathcal{T}_h} T$. In the case that Ω is a convex polyhedral domain, we have $\Omega = \Omega_h$. Let h_T denote the diameter of the element T in $\mathcal{T}_h$ and $h = \max_{T\in\mathcal{T}_h} \{h_T\}$. Let P_1 denote the space of polynomials with degree ≤ 1 and define the finitely dimensional spaces V_h and V_h^0, respectively, as,

$$V_h := \{v_h | v_h \in C(\bar{\Omega}); v_h\mid_T \in P_1, \forall T \in \mathcal{T}_h\},$$
$$V_h^0 := \{v_h | v_h \in V_h, v_h\mid_{\partial\Omega} = 0\}.$$

We use V_h to approximate $H^1(\Omega)$ and $L^2(\Omega)$; and V_h^0 to approximate $H_0^1(\Omega)$. Moreover, let $\{\phi_i(x)\}_{i=1}^n$ be the piecewise linear basis functions of V_h satisfying

$$\phi_i(x) \geq 0, \forall i = 1 \cdots n, \quad \text{and} \quad v_h = \sum_{i=1}^{n} v_i\phi_i(x), \forall v_h \in V_h. \tag{2.35}$$

Then, we can approximate the problem (2.29) by the following finitely dimensional optimal control problem:

$$\begin{aligned} \min_{y\in V_h^0, u\in V_h} \ & \frac{1}{2}\|y_h - y_{dh}\|^2_{L^2(\Omega_h)} + \frac{\alpha}{2}\|u_h\|^2_{L^2(\Omega_h)} + I_{\mathcal{C}_h}(u_h) \\ \text{s.t.} \ & a(y_h, v_h) = (u_h, v_h), \forall v_h \in V_h. \end{aligned} \tag{2.36}$$

In (2.36), $y_{dh} \in V_h$ is an approximation of y_d, and $\mathcal{C}_h$ denotes the discrete admissible set defined by

$$\mathcal{C}_h := V_h \cap \mathcal{C} = \{v_h = \sum_{i=1}^{n} v_i\phi_i(x) | a \leq v_i \leq b, 1 \leq i \leq n\}.$$

In [30], the convergence of the finite element discretization is discussed. We cite this result below.

Theorem 2.7. *(cf. [30]) Let u^* and u_h^* be the optimal solutions of the problem (2.29) and (2.36), respectively. Then, there holds:*

$$\lim_{h\to 0}\frac{1}{h}\|u^*-u_h^*\|_{L^2(\Omega)}=0.$$

This theorem implies that the convergence rate of the finite element discretization is of order $o(h)$, which will be validated by numerical results to be presented in Section 2.5.

Moreover, for numerical implementation, we define the mass matrix M and stiffness matrix K, respectively, as follows

$$M_{ij}=\int_{\Omega_h}\phi_i(x)\phi_j(x)dx,\quad K_{ij}=a(\phi_i(x),\phi_j(x)),\quad \forall 1\le i,j\le n.$$

Using these notations, the discrete optimal control problem originating from (2.29) in matrix-vector form is given by

$$\begin{aligned}\min_{\boldsymbol{y}\in\mathbb{R}^n,\boldsymbol{u}\in\mathbb{R}^n}\ & \frac{1}{2}\|\boldsymbol{y}-\boldsymbol{y}_d\|_M^2+\frac{\alpha}{2}\|\boldsymbol{u}\|_M^2+I_{\boldsymbol{\mathcal{C}}}(\boldsymbol{u})\\ \text{s.t.}\ & K\boldsymbol{y}=M\boldsymbol{u},\end{aligned}\tag{2.37}$$

where $\boldsymbol{y}=\{y_i\}_{i=1}^n$, $\boldsymbol{u}=\{u_i\}_{i=1}^n$ and $\boldsymbol{y}_d=\{y_{di}\}_{i=1}^n$, and the discrete admissible set $\boldsymbol{\mathcal{C}}$ is given by

$$\boldsymbol{\mathcal{C}}=\{\boldsymbol{u}\in\mathbb{R}^n|a\le u_i\le b,\ 1\le i\le n\}.$$

By introducing the adjoint variable $\boldsymbol{p}\in\mathbb{R}^n$, the optimality conditions of (2.37) read as

$$\begin{pmatrix}0\\ M\boldsymbol{y}_d\\ 0\end{pmatrix}\in\begin{pmatrix}\alpha M+\partial I_{\boldsymbol{\mathcal{C}}} & 0 & M^\top\\ 0 & M & -K^\top\\ M & -K & 0\end{pmatrix}\begin{pmatrix}\boldsymbol{u}\\ \boldsymbol{y}\\ \boldsymbol{p}\end{pmatrix}.\tag{2.38}$$

2.4.3 (2.38) is a special case of (2.1)

In this subsection, we illustrate that the nonlinear saddle point system (2.1) includes the discrete optimality conditions (2.38) as a special case. Hence, the inexact Uzawa algorithmic framework (2.6) is implementable for (2.38).

To proceed the discussion, we introduce

$$A = \begin{pmatrix} \alpha M & 0 \\ 0 & M \end{pmatrix}, B = \begin{pmatrix} M & -K \end{pmatrix}, w = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{y} \end{pmatrix}, \tag{2.39}$$

$$v = \boldsymbol{p}, \quad \Theta(w) = I_{\boldsymbol{\mathcal{C}}}(\boldsymbol{u}), \ f = \begin{pmatrix} 0 \\ M\boldsymbol{y}_d \end{pmatrix}, \ g = 0. \tag{2.40}$$

Then, the optimality conditions (2.38) can be reformulated as

$$\begin{pmatrix} f \\ g \end{pmatrix} \in \begin{pmatrix} A + \partial\Theta & B^\top \\ B & 0 \end{pmatrix} \begin{pmatrix} w \\ v \end{pmatrix}, \tag{2.41}$$

which is a special case of (2.1) with $\mathcal{G} = \partial\Theta$. Clearly, for the unconstrained case, there is no $\partial\Theta$, and the nonlinear saddle point (2.41) reduces to the linear saddle point problem (1.20).

2.4.4 Difficulties of implementing (2.6) for (2.38)

In this subsection, we look into details for the implementation of (2.6) to (2.38). In particular, as mentioned, it is important to choose appropriate preconditioners Q_A and Q_B in accordance with the specific structure of the model (2.38). First, according to (2.39), we know that the Schur complement $S = BA^{-1}B^\top$ is reduced to $\frac{1}{\alpha}M + KM^{-1}K^\top$ for the specific nonlinear saddle point problem (2.41), and as well studied in, e.g. [58], $\rho(S)$ is of order $O(h^{-2})$.

Let us now observe the application of the exact Uzawa method (2.7) to (2.38), which reads as

$$\begin{cases} 0 \in \partial I_{\boldsymbol{\mathcal{C}}}(\boldsymbol{u}^{k+1}) + \alpha M\boldsymbol{u}^{k+1} + M^\top \boldsymbol{p}^k, & \text{(2.42a)} \\ 0 = M\boldsymbol{y}^{k+1} - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d, & \text{(2.42b)} \\ 0 = \boldsymbol{p}^{k+1} - \boldsymbol{p}^k - \omega(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}). & \text{(2.42c)} \end{cases}$$

As discussed in Section 1.1 for linear saddle point problems, the implementation of (2.42a)-(2.42c) has some numerical difficulties as listed below.

1. Since M is symmetric and positive definite, the $\boldsymbol{u}$-subproblem (2.42a) is equivalent to the following optimization problem:

$$\min_{\boldsymbol{u}\in\mathbb{R}^n} I_{\boldsymbol{\mathcal{C}}}(\boldsymbol{u}) + \frac{\alpha}{2}\|\boldsymbol{u} + \frac{\boldsymbol{p}^k}{\alpha}\|_M^2. \tag{2.43}$$

The problem (2.43) usually has no closed-form solution because M is not diagonal, despite the simplicity of $I_{\mathcal{C}}(\boldsymbol{u})$. Thus, inner iterations should be embedded into the implementation of (2.42a); hierarchically nested iterations and hence the lack of rigorous analysis for the convergence of the overall scheme (2.42) are thus caused.

2. For the $\boldsymbol{y}$-subproblem (2.42b), it is usually a huge-scale system of linear equations especially for fine discretization cases.

3. For the subproblem (2.42c), the main difficulty is that ω is forced to be tiny for fine discretization cases and thus slow convergence is inevitable. Recall that the scheme (2.42) is derived from the implementation of the exact Uzawa method (2.7) to the problem (2.38). Hence, it can be regarded as taking $Q_B = \frac{1}{\omega}I$ in (2.6) and the convergence condition (2.8) requires that $\omega \leq \frac{1}{\rho(S)}$. As just mentioned, $\rho(S)$ is of order $O(h^{-2})$; thus ω is of order $O(h^2)$ which is very small even for medium values of h. This may easily lead to extremely slow convergence, see, e.g., the analysis in [21] for linear saddle point problems.

2.4.5 Strategies

Now, we elaborate on our ideas for tackling the difficulties listed above. To tackle the first difficulty, a natural idea is to add a proximal term $\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{u}^{\boldsymbol{k}}\|_{c_0 I-\alpha M}$ to the objective function in (2.43). With some special choice of the constant $c_0 > 0$ such that $c_0 I - \alpha M \succeq 0$, the proximally regularized problem may have a closed-form solution. Such a strategy has been widely used in areas such as optimization and image processing, see e.g., [183, 186, 191]. For the problem under our discussion, however, this proximal regularization technique seems not to be useful because the convergence condition $c_0 I - \alpha M \succeq 0$ means that c_0 depends on h and it can rarely be sufficiently large. Here, we consider the lump mass matrix

$$W := \mathrm{diag}(\int_{\Omega_h} \phi_i(x)dx)_{i=1}^n,$$

where $\phi_i(x), i = 1, \cdots, n$ are basis functions defined in (2.35). It has been shown in [184] that

$$W \succeq M, \quad \forall h > 0. \tag{2.44}$$

Adding the proximal term $\frac{\alpha}{2}\|\boldsymbol{u}-\boldsymbol{u^k}\|_{W-M}$ to the objective function, the problem (2.43) is transformed to

$$\min_{\boldsymbol{u}\in\mathbb{R}^n} I_{\boldsymbol{\mathcal{C}}}(\boldsymbol{u}) + \frac{\alpha}{2}\|\boldsymbol{u} + W^{-1}(\frac{1}{\alpha}M^\top \boldsymbol{p}^k - (W-M)\boldsymbol{u}^k)\|_W^2,$$

whose optimality condition reads as

$$0 \in \partial I_{\boldsymbol{\mathcal{C}}}(\boldsymbol{u}^{k+1}) + \alpha W\boldsymbol{u}^{k+1} - \alpha(W-M)\boldsymbol{u}^k + M^\top \boldsymbol{p}^k. \tag{2.45}$$

Since W is diagonal, we obtain

$$\boldsymbol{u}^{k+1} = P_{\boldsymbol{\mathcal{C}}}(W^{-1}((W-M)\boldsymbol{u}^k - \frac{1}{\alpha}M^\top \boldsymbol{p}^k)),$$

where $P_{\boldsymbol{\mathcal{C}}}$ is the projection onto the admissible set $\boldsymbol{\mathcal{C}}$ and it usually can be easily computed because of the simplicity of the set $\boldsymbol{\mathcal{C}}$.

For the second difficulty, we consider some iterative schemes that are tailored for the system of linear equations. Since M is diagonally dominant, we can choose $D = 2\,\mathrm{diag}(M)$ which implies immediately that

$$D \succ M. \tag{2.46}$$

With this choice, the computation of $\boldsymbol{y}^{k+1}$ is essentially the application of the damped Jacobi iteration method to the $\boldsymbol{y}$-subproblem (2.42b), which reads as:

$$\boldsymbol{y}^{k+1} = \boldsymbol{y}^k - D^{-1}(M\boldsymbol{y}^k - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d). \tag{2.47}$$

As a result, we can update $\boldsymbol{y}^{k+1}$ element-wisely, which is computationally inexpensive and easily implementable.

Last, we discuss how to choose appropriate preconditioners Q_A and Q_B in accordance with the specific structure of the model (2.38) to tackle the third difficulty. First, taking (2.6) into account, we know that (2.45) and (2.47) essentially imply that

$$Q_A := \begin{pmatrix} \alpha W & 0 \\ 0 & D \end{pmatrix},$$

and it follows from (2.44) and (2.46) that $Q_A \succeq A$.

For the preconditioner Q_B, inspired by [21] for linear saddle point problems, an ideal choice is $Q_B = S$. But it is generally too difficult to compute Q_B^{-1} because the condition number of S is of order $O(h^{-4})$ which means S is extremely ill-conditioned even for not very small values of h. Therefore, we need to choose Q_B to balance the efficiency of the resulting algorithm and the computation of Q_B^{-1}. For this purpose, we note that the Schur complement S can be written as

$$S = (K + \frac{1}{\sqrt{\alpha}}M)M^{-1}(K + \frac{1}{\sqrt{\alpha}}M)^\top - \frac{2}{\sqrt{\alpha}}K.$$

According to the work [145, 146], the Schur complement S can be approximated by

$$P_B = (K + \frac{1}{\sqrt{\alpha}}M)M^{-1}(K + \frac{1}{\sqrt{\alpha}}M)^\top, \tag{2.48}$$

which drops off the term $\frac{2}{\sqrt{\alpha}}K$ in S. Let us recall a known result below.

Theorem 2.8. *(cf [145].) Suppose that we approximate S by P_B. Then, we can bound the eigenvalues of $P_B^{-1}S$ as follows:*

$$\lambda(P_B^{-1}S) \in [\frac{1}{2},\ 1],$$

which is independent of α and h.

Therefore, the positive definiteness of K and the theorem above ensure that

$$P_B \succ S. \tag{2.49}$$

This theorem also implies that we can use P_B defined in (2.48) as a surrogate of the preconditioner Q_B for the Schur complement S if we can approximate the matrices $(K + \frac{1}{\sqrt{\alpha}}M)$ and $(K + \frac{1}{\sqrt{\alpha}}M)^\top$ efficiently. Thus, implementing the inexact Uzawa algorithmic framework (2.6) for (2.38) with the surrogate P_B essentially requires to compute $(K + \frac{1}{\sqrt{\alpha}}M)^{-1}$ and $(K + \frac{1}{\sqrt{\alpha}}M)^{-\top}$. The matrices $(K + \frac{1}{\sqrt{\alpha}}M)$ and $(K + \frac{1}{\sqrt{\alpha}}M)^\top$, however, are still ill-conditioned due to the presence of the stiffness matrix K whose condition number is of order $O(h^{-2})$ (see, e.g. [58]). Hence, it is not practical to directly compute these two inverses. Since K is the discretization of the linear second-order elliptic operator $\mathcal{K}$, as discussed in, e.g. [26, 172], a spectrally equivalent approximation of K can be

obtained by performing one or more multi-grid sweeps. We thus can follow some well-studied literatures such as [181, 185] to execute two algebraic multi-grid (AMG) V-circles to approximate the computations of $(K+\frac{1}{\sqrt{\alpha}}M)^{-1}v$ and $(K+\frac{1}{\sqrt{\alpha}}M)^{-\top}v$, respectively, for a given vector v. Note that, via the implementation of AMG V-circles, the matrix $(K+\frac{1}{\sqrt{\alpha}}M)$ is implicitly approximated by a matrix, denoted by G. Hence, the expression of G is unknown. As we shall show in Section 6 by numerical results, this approximation is good enough to ensure very fast convergence. The implementation of AMG V-circles is based on the iFEM package developed in [39] with a Jacobian smoother.

2.4.6 A specific inexact Uzawa type algorithm for (2.29)

In summary, we choose the preconditioners Q_A and Q_B as

$$Q_A := \begin{pmatrix} \alpha W & 0 \\ 0 & D \end{pmatrix} \succeq A, \text{ and } Q_B := \tau G M^{-1} G^\top \succeq P_B \succ S, \tag{2.50}$$

where the constant $\tau > 0$ should be large enough to ensure $Q_B \succeq P_B$. Note that it follows from (2.49) that $Q_B \succ S$. Moreover, since the algebraic multi-grid process G is a spectrally equivalent approximation to $K+\frac{1}{\sqrt{\alpha}}M$ (see, e.g., [145, 146]), it is easy to determine the value of τ around 1 such that $Q_B \succ S$ is satisfied. More details will be shown in Section 2.5. We thus obtain the following specified algorithm via inexact Uzawa algorithmic framework (2.6) for the problem (2.37):

$$\begin{cases} 0 \in \partial I_{\boldsymbol{\mathcal{C}}}(\boldsymbol{u}^{k+1}) + \alpha W \boldsymbol{u}^{k+1} - \alpha(W-M)\boldsymbol{u}^k + M^\top \boldsymbol{p}^k, & \text{(2.51a)} \\ 0 = D(\boldsymbol{y}^{k+1} - \boldsymbol{y}^k) + M\boldsymbol{y}^k - K^\top \boldsymbol{p}^k - M\boldsymbol{y}_d, & \text{(2.51b)} \\ 0 = \boldsymbol{p}^{k+1} - \boldsymbol{p}^k - Q_B^{-1}(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1}). & \text{(2.51c)} \end{cases}$$

We would reiterate that in (2.50) the matrix $G \approx K+\frac{1}{\sqrt{\alpha}}M$ (hence, $G^\top \approx (K+\frac{1}{\sqrt{\alpha}}M)^\top$) and the approximation is in an implicit manner, while there is no need to know the explicit expression of G to implement (2.51). The reason is that Q_B is approximated by the P_B defined in (2.48), and thus the computation of $Q_B^{-1}(M\boldsymbol{u}^{k+1} - K\boldsymbol{y}^{k+1})$ in (2.51c) is realized via totally four AMG V-circles implemented on $(K+\frac{1}{\sqrt{\alpha}}M)$ and $(K+\frac{1}{\sqrt{\alpha}}M)^\top$, respectively, plus a matrix-vector multiplication involving the matrix M. Since $Q_A \succeq A$ and $Q_B \succ S$,

the convergence condition (2.8) is verified. As a result, the convergence of the resulting algorithm (2.51) follows from Theorem 2.2 directly. Since $I_{\boldsymbol{\mathcal{C}}}$ is the indicator function of a box constraint, Assumption 2.1 is satisfied. Therefore, according to Theorem 2.5, the algorithm (2.51) indeed converges linearly. We shall verify these theoretical results in Section 2.5.

Our explanations above indicate that, because of the well-chosen preconditioners in (2.50), the algorithm (2.51) designed specially for the elliptic optimal control problem with control constraints (2.37) is featured by the fact that each of its iteration only requires computing a projection onto the admissible set (see (2.51a)), four AMG V-cycles (see (2.51c)) and some matrix-vector multiplications. There is no any optimization problem or system of linear equations which should be solved iteratively in its iterations, and thus nested iterations are completely avoided. Therefore, it is very easy to implement the specific algorithm (2.51).

2.5 Numerical results

In this section, we test the elliptic optimal control problem with control constraints (2.29) and numerically verify the efficiency of the specific algorithm (2.51) derived from the algorithmic framework (2.6).

First, we would like to mention that there are other numerical schemes in the literature that can be applied to various elliptic optimal control problems with control constraints. For example, one can regard the corresponding first-order optimality conditions of the problem under discussion (2.32)-(2.34) as a nonsmooth equation and then apply some so-called semismooth Newton (SSN) type methods (see, e.g. [99, 103, 104, 147, 174]). In [14], the primal dual active set (PDAS) method is proposed and it can be viewed as a combination of an SSN type method with the active set strategy proposed in [101]. It is proved in [174] that SSN type methods may have locally superlinear convergence rates and they are capable of approaching high-precision solutions. Meanwhile, it is known that SSN type methods require solving a possibly large-scale and ill-conditioned

system of linear equations at each of their iterations, in order to compute the Newton step, on top of that the convergence of SNN type methods usually depends on well-chosen initial iterate. In addition, it follows from Theorem 2.7 that the discretization error is $o(h)$ and it accounts for the main part of the total error of utilizing some numerical scheme to solve the problem (2.29). In this sense, approaching an approximate solution in a very high precision may not necessarily reduce the total error order, while significantly more computation loads are caused. Therefore, it is also interesting to consider such a numerical scheme that can find approximate solutions in medium-precision but with significantly less computational time. The proposed inexact Uzawa algorithmic framework (2.6) is such an effort.

We also notice that some schemes based on the so-called alternating direction method of multipliers (ADMM) (see [72]) have been proposed in the literature for solving the elliptic optimal control problem with control constraints, see e.g. [162]. As the proposed algorithm (2.51), these ADMM type schemes do explore the separate structures of the models under investigation and thus each iteration consists of a projection onto the admissible set and a large-scale linear saddle point problem. But solving the linear saddle point problems iteratively per se may be computationally expensive, and it results in nested iterations and hence causes new difficulties in rigorously ensuring the overall convergence. By contrast, as elaborated in Section 5, the algorithm (2.51) derived from the algorithmic framework (2.6) does not require to solve any optimization subproblems or systems of linear equations numerically over the fine mesh. Therefore, because of the aforementioned very different natures, we do not numerically compare the algorithm (2.51) with SSN type and ADMM type methods in the literature.

Our codes were written in MATLAB R2016b and experiments were implemented on a Surface Pro 5 laptop with 64-bit Windows 10.0 operation system, Intel(R) Core(TM) i7-7660U CPU (2.50 GHz), and 16 GB RAM. To test our proposed algorithm (2.51), we define the primal and dual residuals, respectively, as:

$$p_s := (\|u^k - u^{k-1}\|^2_{L^2(\Omega)} + \|y^k - y^{k-1}\|^2_{L^2(\Omega)})^{\frac{1}{2}} \quad \text{and} \quad d_s := \|p^k - p^{k-1}\|_{L^2(\Omega)};$$

and the stopping criterion is set as

$$\max(p_s, d_s) \leq tol,$$

where $tol > 0$ is a prescribed tolerance. The initial values are set as $u = 0, y = 0$ and $p = 0$ for all experiments.

Example 1. We first consider the example given in [104]:

$$\min_{y \in H_0^1(\Omega), u \in \mathcal{C}} \quad J(y,u) = \tfrac{1}{2}\|y - y_d\|^2_{L^2(\Omega)} + \tfrac{\alpha}{2}\|u\|^2_{L^2(\Omega)}$$

$$\text{s.t.} \qquad \begin{cases} -\Delta y = u \quad \text{in} \quad \Omega, \\ \quad y = 0 \quad \text{on} \quad \Gamma, \end{cases}$$

where the domain $\Omega = (0,1) \times (0,1)$ and constraints on the control variable u are $a = 0.3$ and $b = 1$. In addition, we set the desired state $y_d = 4\pi^2 \alpha \sin(\pi x)\sin(\pi y) + y_r$, where y_r denotes the solution associated with the problem

$$\begin{cases} -\Delta y_r = r \quad \text{in} \quad \Omega, \\ \quad y_r = 0 \quad \text{on} \quad \Gamma, \end{cases}$$

and $r = \min\{1, \max\{0.3, 2\sin(\pi x)\sin(\pi y)\}\}$. It follows from the construction of y_d and r that $u^* = r$ is the unique solution of this example. The exact solution u^* on a grid with $h = 2^{-6}$ is presented in Figure 2.1. Note that we only consider the boundary condition $y = 0$ in order to simplify the process of constructing an example with a known exact solution. In practice, it is not necessary to assume the boundary condition $y = 0$; the scope to which our proposed method can be applied is not affected by the boundary condition. Indeed, the state equation is an elliptic problem with Dirichlet boundary condition; so we only need to operate on interior points of the domain of the problem and thus the boundary condition does not affect the coefficient matrix of the linear system after discretization. Similar assumptions can be found in many literatures such as [99, 101, 103, 145, 146].

We first set $tol = 10^{-6}$ and test the algorithm (2.51) with different discretization mesh sizes: $h = 2^{-i}, i = 3, \cdots, 10$ and different regularization parameters: $\alpha = 10^{-j}, j = 2, \cdots, 6$. For tests with $h = 2^{-9}, tol = 10^{-6}$, and $\alpha = 10^{-j}, j = 2, \cdots, 6$, the iteration numbers range between 43 and 66, which

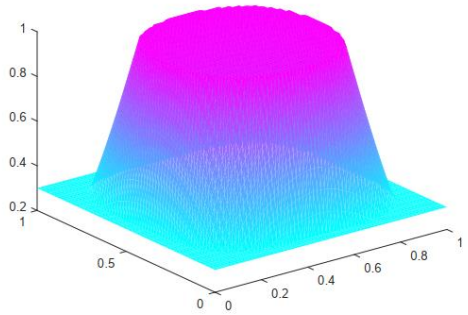


Figure 2.1: Exact solution u^* of Example 1.

implies that the numerical performance of (2.51) is robust to various values of the regularization parameter α. Hence, as [104], we just report the results for $\alpha = 10^{-4}$ for succinctness. Accordingly, the value of τ for Q_B in (2.50) is tuned to be 0.5. The numerical results with different mesh sizes h are presented in Table 2.1. The computing time in the third column show that we can obtain a medium-precision numerical solution in a short time even for fine discretization cases such as $h = 2^{-10}$. Compared with the result in [162], we observe that, although less iteration numbers are reported therein, the ADMM-based scheme in [162] requires much more computing time especially when the mesh size h is small. As mentioned, it is because a linear saddle point problem in form of (1.20) should be solved at each iteration and thus nested iterations are inevitable.

We also observe that the additional constraint in the admissible set does not increase computing time for the algorithm (2.51). We further test the unconstrained case, i.e., $a = -\infty$ and $b = +\infty$, and compare it with the constrained case for a fixed number of iterations (say, 100). The comparison is reported in Table 2.2.

In Figure 2.4, we present the errors $\|u-u^*\|_{L^2(\Omega)}$ and $\|y-y^*\|_{L^2(\Omega)}$, the primal and dual residuals with $h = 2^{-6}$ and $tol = 10^{-6}$. The results presented here verify the linear convergence rate discussed in Section 2.3. Numerical results of $y, y-y_d$ and $u, u-u^*$ with $h = 2^{-6}$ and $tol = 10^{-6}$ are reported in Figures 2.2 and 2.3, respectively.

To validate the finite element discretization error estimates, we define the

Table 2.1: Numerical results of the algorithm (2.51) for Example 1.

h	$Iter$	$CPU(s)$	$\frac{\|y-y_d\|_{L^2(\Omega)}}{\|y_d\|_{L^2(\Omega)}}$	$J(y,u)$	$\|u\|_{L^2(\Omega)}$
2^{-3}	41	0.103278	5.5599×10^{-2}	2.6848×10^{-5}	0.7055
2^{-4}	48	0.170980	5.2592×10^{-2}	2.8549×10^{-5}	0.7294
2^{-5}	47	0.284057	5.1871×10^{-2}	2.9146×10^{-5}	0.7375
2^{-6}	47	0.435089	5.1684×10^{-2}	2.9390×10^{-5}	0.7399
2^{-7}	47	0.785748	5.1640×10^{-2}	2.9496×10^{-5}	0.7423
2^{-8}	47	2.396362	5.1628×10^{-2}	2.9547×10^{-5}	0.7430
2^{-9}	47	11.192291	5.1626×10^{-2}	2.9571×10^{-5}	0.7433
2^{-10}	47	49.309952	5.1625×10^{-2}	2.9583×10^{-5}	0.7434

Table 2.2: Computing time(s) comparison between unconstrained(U) and constrained(C) cases for Example 1.

h	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}
U	0.1135	0.1815	0.2317	0.5108	1.2521	4.3241	19.1093	85.3758
C	0.1273	0.2556	0.4384	0.5899	1.2639	4.0090	19.7053	86.5423

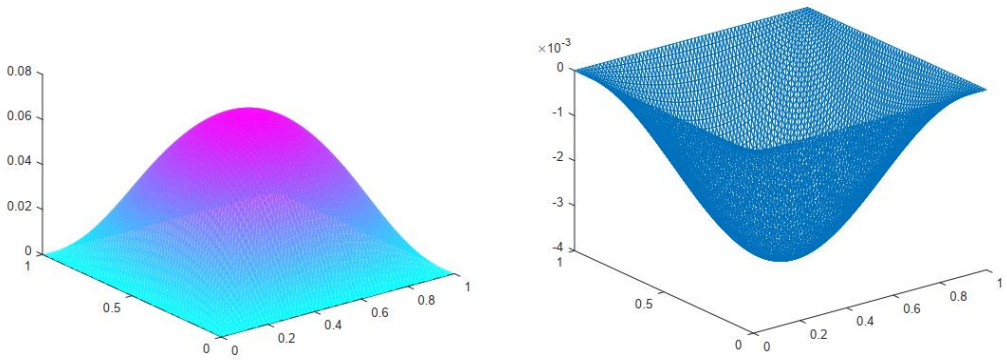


Figure 2.2: Numerical solution y and error $y - y_d$ of Example 1.

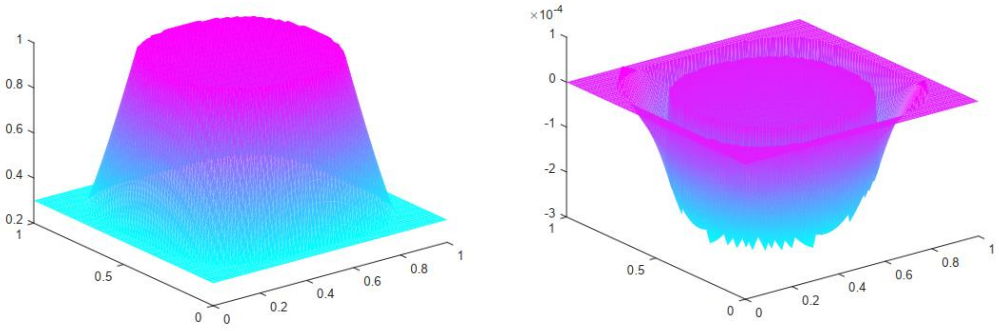


Figure 2.3: Numerical solution u and error $u - u^*$ of Example 1.

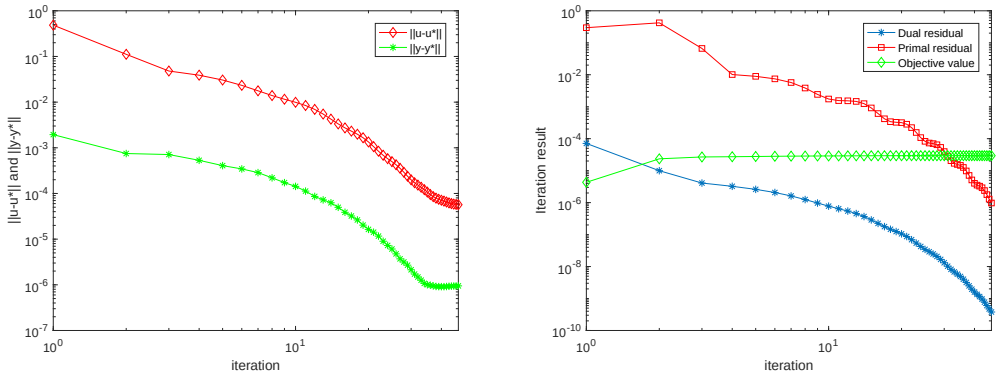


Figure 2.4: Iteration error $\|u - u^*\|$ and $\|y - y^*\|$ (left), primal residual p_s, dual residual d_s and objective function value (right) of Example 1.

experimental order of convergence (EOC) as

$$EOC := \frac{\log(e(h_1)) - \log(e(h_2))}{\log(h_1) - \log(h_2)}, \tag{2.52}$$

where $e(h) > 0$ is the error with L^2-norm, and h_1 and h_2 denote two consecutive mesh sizes. From (2.52), it is easy to show that, if $e(h) = O(h^\beta)$, then $EOC = \beta$. In addition, to eliminate the iteration error of the algorithm (2.51) , we set $tol = 10^{-9}$ in the following experiments. For this case, the discretization error dominates the total error. For this purpose, we specify $e(h)$ with respect to u and y as, respectively,

$$e_u(h) = \|u^* - u_h^*\|_{L^2(\Omega)},\ e_y(h) = \|y^* - y_h^*\|_{L^2(\Omega)}.$$

We report the numerical results in Table 2.3 for some choices of the mesh size h. It is observed that the convergence order of the finite element discretization of the control variable in our experiments is approximately $O(h^2)$. This empirically validates the theoretically derived order of $o(h)$ in Theorem 2.7. In addition, the convergence order of the state variable y is $O(h^2)$ which coincides with the theoretical result shown in [64]. Moreover, the third column in Table 2.3 shows that it only requires less than 80s in the finest discretization case ($h = 2^{-10}$) to reach highly accurate solutions. This further validates the efficiency of the algorithm (2.51) for solving the elliptic optimal control problem with control constraints (2.29).

Table 2.3: Convergence order of finite element discretization for Example 1.

h	$Iter$	$CPU(s)$	$e_u(h)$	$EOC(u)$	$e_y(h)$	$EOC(y)$
2^{-3}	77	0.124715	3.4529×10^{-3}	$\sim$	6.1846×10^{-5}	$\sim$
2^{-4}	83	0.166852	8.7410×10^{-4}	1.9819	1.5136×10^{-5}	2.0281
2^{-5}	80	0.353843	2.1415×10^{-4}	2.0291	3.8981×10^{-6}	1.9597
2^{-6}	82	0.511972	5.3773×10^{-5}	1.9937	9.6936×10^{-7}	2.0076
2^{-7}	82	1.242614	1.3438×10^{-5}	2.0005	2.4253×10^{-7}	1.9987
2^{-8}	82	3.433469	3.3609×10^{-6}	1.9994	6.6031×10^{-8}	2.0000
2^{-9}	82	16.899456	8.4238×10^{-7}	1.9963	1.5157×10^{-8}	2.0000
2^{-10}	82	74.143923	2.1271×10^{-7}	1.9856	3.7853×10^{-9}	2.0015

Example 2. We consider the following problem:

$$\min_{y\in H_0^1(\Omega),u\in\mathcal{C}} \quad J(y,u) := \tfrac{1}{2}\|y-y_d\|^2_{L^2(\Omega)} + \tfrac{\alpha}{2}\|u\|^2_{L^2(\Omega)}$$

$$\text{s.t.} \qquad \begin{cases} -\Delta y + y = u & \text{in} \quad \Omega, \\ y = 0 & \text{on} \quad \Gamma, \end{cases}$$

where the target function $y_d = \sin(2\pi x)\sin(2\pi y)$, $\Omega = (0,1)\times(0,1)$, and the admissible set $\mathcal{C}$ is given by

$$\mathcal{C} = \{v(x) \in L^\infty(\Omega) | -70 \le v(x) \le 70, \text{a.e. in } \Omega\} \subset L^2(\Omega).$$

We set the parameter $\alpha = 10^{-6}$ and the constant $\tau = 0.5$. Compared with Example 1, this example is more general and its exact solution is unknown.

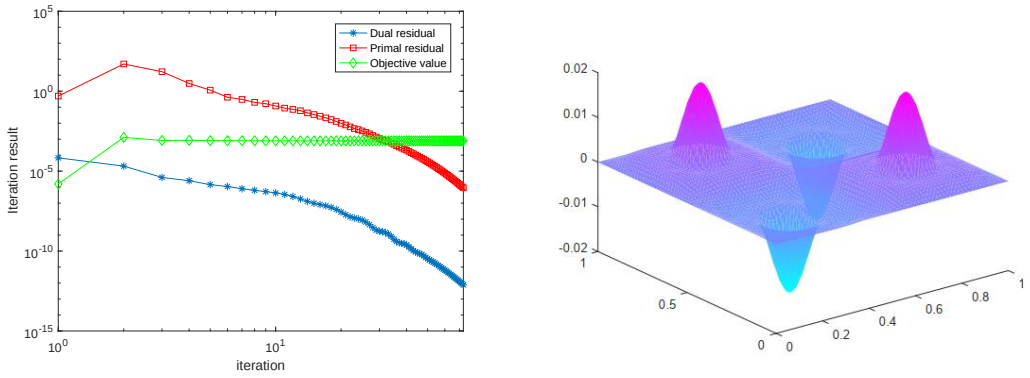
Similar as Example 1, we first test the algorithm (2.51) for finding medium-precision solutions with $tol = 10^{-6}$ for different values of the mesh size h. Numerical results are summarized in Table 2.4. It shows that the algorithm (2.51) converges very fast. It is observed that the resulting iteration number seems not to be affected by the mesh size. We fix the iteration number as 100 and compare the computing time for both the constrained and unconstrained cases with different mesh size h. The results are reported in Table 2.5. It is verified again that the additional constraint in the admissible set does not increase computing time for the algorithm (2.51). We also report some numerical results with $h = 2^{-6}$ and $tol = 10^{-6}$ in Figures 2.5 and 2.6, respectively.

Table 2.4: Numerical results of the algorithm (2.51) for Example 2.

h	$Iter$	$CPU(s)$	$\frac{\|y-y_d\|_{L^2(\Omega)}}{\|y_d\|_{L^2(\Omega)}}$	$J(y,u)$	$\|u\|_{L^2(\Omega)}$
2^{-3}	76	0.169272	1.6454×10^{-2}	8.7736×10^{-4}	41.2225
2^{-4}	75	0.175245	9.2394×10^{-2}	7.9962×10^{-4}	39.7361
2^{-5}	76	0.307826	7.6773×10^{-3}	7.8246×10^{-4}	39.3749
2^{-6}	73	0.461259	7.5190×10^{-3}	7.7863×10^{-4}	39.2832
2^{-7}	69	0.899516	7.4743×10^{-3}	7.7769×10^{-4}	39.2608
2^{-8}	65	2.749966	7.4641×10^{-3}	7.7745×10^{-4}	39.2553
2^{-9}	60	14.249536	7.4613×10^{-3}	7.7739×10^{-4}	39.2538
2^{-10}	59	49.860595	7.4606×10^{-3}	7.7738×10^{-4}	39.2535

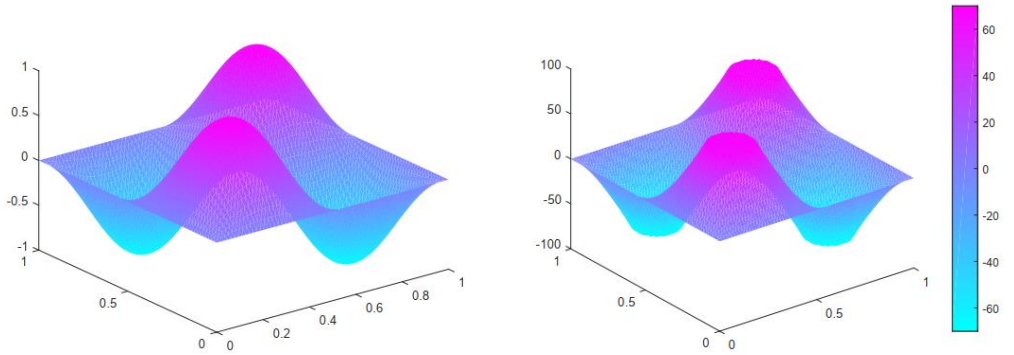
Table 2.5: Computing time (s) comparison between unconstrained (U) and constrained (C) cases for Example 2.

h	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}
U	0.1633	0.2428	0.4106	0.7522	1.2153	3.9141	17.5494	82.7913
C	0.1685	0.2624	0.3938	0.6672	1.0936	3.6811	17.3901	80.3295



(a) Iteration result (b) $y - y_d$

Figure 2.5: Iteration result(left) and error $y - y_d$(right) with $h = 1/64$ of Example 2.



(a) y (b) u

Figure 2.6: Numerical solution y (left) and u (right) with $h = 1/64$ of Example 2.

Chapter 3

Implementation of the ADMM to Parabolic Optimal Control Problems with Control Constraints and Beyond

In this chapter, we consider the following optimal control problem with a parabolic PDE constraint and a box constraint on the control variable:

$$\min_{u\in U_{ad}, y\in L^2(Q)} \quad \frac{1}{2}\iint_Q |y-y_d|^2 dxdt + \frac{\alpha}{2}\iint_{\mathcal{O}} |u|^2 dxdt \tag{3.1}$$

subject to the state equation

$$\begin{cases} \dfrac{\partial y}{\partial t} - \nu\Delta y + a_0 y = u\chi_{\mathcal{O}}, & \text{in} \quad \Omega\times(0,T), \\ y = 0, & \text{on} \quad \Gamma\times(0,T), \\ y(0) = \varphi, \end{cases} \tag{3.2}$$

where Ω is an open bounded domain in $\mathbb{R}^d$ ($d \geq 1$) and $\Gamma = \partial\Omega$ is the piecewise continuous boundary of Ω; ω is an open subset of Ω and $0 < T < +\infty$; the domain $Q = \Omega\times(0,T)$ and $\mathcal{O} = \omega\times(0,T)$. In (3.1)–(3.2), u and y are called the control variable and state variable, respectively. The target function y_d is given in $L^2(Q)$ and the admissible set U_{ad} is defined by

$$U_{ad} = \{v | v \in L^\infty(\mathcal{O}), a \leq v(x;t) \leq b \text{ a.e. in } \mathcal{O}\} \subset L^2(\mathcal{O}).$$

In addition, we denote by $\Delta := \nabla \cdot \nabla$ the Laplace operator and $\chi_{\mathcal{O}}$ the characteristic function of the set $\mathcal{O}$. The constant $\alpha > 0$ is a regularization parameter; a and b are given constants; the initial value φ is given in $L^2(\Omega)$. The coefficients a_0 (≥ 0) $\in L^\infty(Q)$ and ν is a positive constant.

3.1 Some existing algorithms

3.1.1 Parabolic optimal control problems without control constraints

For the special case of the problem (3.1)–(3.2) where $U_{ad} = L^2(\mathcal{O})$, i.e., there is no constraint on the control variable, the resulting problem is called an unconstrained parabolic optimal control problem and it has been well studied in some earlier literatures such as [125] and some more recent ones such as [171]. There is a rich set of papers discussing how to solve unconstrained parabolic optimal control problems numerically; and methods in the literature can be generally categorized as the "black-box" and "all-at-once" approaches. The "black-box" approach commonly suggests substituting the state equation into the objective functional to eliminate the state variable y, and treats an unconstrained parabolic optimal control problem as an optimization problem with respect to the control variable u. Note that each iteration of a "black-box" approach requires solving the involved state equation. We refer to [81, 83] for some efficient "black-box" type numerical schemes for unconstrained parabolic control problems with different types of control variables. On the other hand, the "all-at-once" approach keeps the state equation in the constraints, and treats both the state and control variables separately. The optimality condition of such a resulting constrained optimization problem after discretization can be represented as a linear saddle point system, which can be solved by some efficient iterative solvers such as Krylov subspace methods. We refer to [132, 144, 176] for more details. Both "black-box" and "all-at-once" approaches can be combined with standard techniques such as domain decomposition methods and multi-grid methods to further improve their numerical performance; see, e.g., [7, 16, 70, 98, 131], for some intensive study.

3.1.2 SSN methods for parabolic optimal control problems with control constraints

In the literature, semi-smooth Newton (SSN) methods are state-of-the-art for various optimal control problems with control constraints. For instance, SSN methods have been intensively studied for optimal control problems with elliptic PDE constraints; see, e.g., [101, 103, 175] and reference therein. A common feature of SSN methods is that a semismooth Newton direction is constructed by using a generalized Jacobian in sense of Clarke (see [45]) and then a Newton iteration is expressed in terms of certain active set strategy which identifies the active and inactive indices iteratively in accordance with the control constraints, see, e.g., [14, 147]. In [14], some adaptive strategies have been proposed to alleviate the computational load of the Newton iterations with the resulting iteratively varying coefficient matrices. As analyzed in [101], a SSN method with an active set strategy can be explained as the primal-dual active set (PDAS) strategy studied in [14] for certain problems such as linear-quadratic optimal control problems with box control constraints, including the problem (3.1)–(3.2). The convergence of the PDAS approach can be found in [117] while some numerical results are also reported therein for parabolic boundary control problems with $d = 1$. In [101], it has been proved that SSN methods possess locally superlinear convergence and usually can find high-precision solutions, on the condition that that some initial values can be deliberately chosen. Note that it is assumed by default that the resulting Newton systems should all be solved exactly to validate the theoretical analysis and hence the mentioned nice properties of SSN type methods. Computationally, it is notable that the Newton systems arising in SSN methods are usually ill-conditioned, and as commented in [166] that "it is never solved without the application of a preconditioner". Seeking appropriate preconditioners so as to improve the spectral properties of the Newton systems is indeed a major factor to ensure the success of implementing a SSN type method. In the literature, e.g., [102, 147, 160, 165, 177], some preconditioned iterative solvers were proposed for various SSN methods.

One motivation of considering SSN type methods for the general case of the problem (3.1)–(3.2) with $U_{ad} \subsetneq L^2(\mathcal{O})$ is that the indicator function of the ad-

ditional constraint on the control variable $u \in U_{ad}$ arising in the optimality condition of the problem (3.1)–(3.2) is nonsmooth; hence gradient type methods are not applicable, see e.g., [103, 171]. But, a particular obstacle of applying SSN type methods to the problem (3.1)–(3.2) is that the simple box constraint on the control variable is forced to be considered together with the main parabolic PDE (3.2) simultaneously. Despite that the computational load of assembling the Newton systems can be alleviated by the adaptive strategies in [14], the varying active sets require adjusting the preconditioners iteratively. Indeed, as commented in [165], "we have recomputed the preconditioner for every application involving a different active set" and that "the recomputation of the preconditioner needs to be avoided". Hence, the simple constraint on the control variable unnecessarily complicates the Newton systems because of the request of active-set-dependent preconditioning, and this feature makes it difficult to apply SSN type methods to the problem (3.1)–(3.2).

Implementation of SSN type methods to the general case of the problem (3.1)–(3.2) with $d \geq 2$ is further restrained by the inevitably high dimensionality of the resulting Newton systems. To elaborate, if we set the mesh sizes of both the time and space discretizations as $1/100$, then the dimensionality of the resulting Newton system at each iteration is order of $O(10^6)$ for $d = 2$ and $O(10^8)$ for $d = 3$. Hence, for some time-dependent problems such as (3.1)–(3.2) with $d \geq 2$, it is not practical to solve such large-scale Newton systems either exactly or up to high precisions. It is thus necessary to discern some criterion that can be implemented easily, and to investigate the convergence if these Newton systems can only be solved up to certain levels of accuracy due to the difficulty of high dimensionality. In the literature, usually some empirically perceived constant accuracy is set a prior, and certainly fixing a constant accuracy by liberty may unnecessarily result in either too accurate computation (hence slower convergence) or too loose approximation (hence possible divergence) for the internal iterations[1]. There seems still to lack of discussions on how to specify the inexactness criterion methodologically and how to prove the convergence of the resulting inexact executions rigorously in the literature of SSN methods. Also,

[1]The same concerns also apply to the interior point methods in, e.g., [143], for different types of optimal control problems.

as mentioned in, e.g., [147], some SSN methods require the accuracy for internal iterations to be increased when the mesh size for discretization becomes smaller. This essentially increases the computational load for solving the Newton systems and may significantly slow down the overall convergence if fine meshes are used to discretize the problem (3.1)–(3.2).

3.2 Conceptual application of ADMM

Inspired by the aforementioned difficulties in the consideration of implementing the well-studied SSN methods to the problem (3.1)–(3.2), our first motivation is to design an algorithm that can treat the parabolic PDE constraint (difficult one) and the box constraint on the control variable (easy one) separately in its execution. A particular goal is that the subproblems associated with the parabolic PDE constraint arising in different iterations should have invariant coefficient matrices so that certain numerical strategy such as preconditioning can be uniformly applied. To this end, it suffices to consider the well-studied alternating direction method of multipliers (ADMM) which was first introduced by Glowinski and Marroco in [72] for nonlinear elliptic problems.

Let us see how the ADMM can be applied to the problem (3.1)–(3.2) and a prototype algorithm can be obtained immediately. For this purpose, we let $S : L^2(\mathcal{O}) \longrightarrow L^2(Q)$ be an affine solution operator associated with the state equation (3.2); and it is defined as

$$S(u) := y. \tag{3.3}$$

It is clear that S is bounded, continuous and compact. More properties of the operator S can be referred to [171]. With $y = S(u)$, the problem (3.1)–(3.2) can be rewritten as

$$\min_{u \in U_{ad}} \quad \frac{1}{2\alpha} \iint_Q |S(u) - y_d|^2 dxdt + \frac{1}{2} \iint_{\mathcal{O}} |u|^2 dxdt,$$

which is actually a scaled version of the problem (3.1)–(3.2). Further, by introducing an auxiliary variable $z \in L^2(\mathcal{O})$ such that $u = z$, the problem (3.1)–(3.2)

can be written as the following separable convex optimization problem

$$\begin{cases} \min\limits_{(u,z)\in L^2(\mathcal{O})\times L^2(\mathcal{O})} & \tilde{J}(u) + I_{U_{ad}}(z) \\ \text{s.t.} & u = z, \end{cases} \tag{3.4}$$

where $I_{U_{ad}}(\cdot)$ is the indicator function of the admissible set U_{ad} and

$$\tilde{J}(u) := \frac{\gamma}{2}\iint_Q |S(u) - y_d|^2 dxdt + \frac{1}{2}\iint_{\mathcal{O}} |u|^2 dxdt, \text{ with } \gamma = \frac{1}{\alpha}. \tag{3.5}$$

The augmented Lagrangian functional associated with the problem (3.4) can be defined as

$$L_\beta(u, z, \lambda) := \tilde{J}(u) + I_{U_{ad}}(z) - (\lambda, u - z) + \frac{\beta}{2}\|u - z\|^2,$$

in which $(\cdot,\cdot)$ and $\|\cdot\|$ are the canonical inner product and norm in $L^2(\mathcal{O})$, respectively; $\lambda \in L^2(\mathcal{O})$ is the Lagrange multiplier associated the constraint $u = z$, and $\beta > 0$ is a penalty parameter. To simplify the discussion, the penalty parameter is fixed throughout our discussion. Then, implementing the ADMM in [72] to (3.4), we immediately obtain the scheme

$$\begin{cases} u^{k+1} = \arg\min\limits_{u\in L^2(\mathcal{O})} L_\beta(u, z^k, \lambda^k), & \text{(3.6a)} \\ z^{k+1} = \arg\min\limits_{z\in L^2(\mathcal{O})} L_\beta(u^{k+1}, z, \lambda^k), & \text{(3.6b)} \\ \lambda^{k+1} = \lambda^k - \beta(u^{k+1} - z^{k+1}). & \text{(3.6c)} \end{cases}$$

3.2.1 Remarks on the direct application of ADMM

The ADMM can be regarded as a splitting version of the classic augmented Lagrangian method (ALM) proposed in [100, 148]. At each iteration of the ALM, the subproblem is decomposed into two parts and they are solved in the Gauss-Seidel manner. A key feature of the ADMM is that the decomposed subproblems usually are much easier than the ALM subproblems and it becomes more likely to take advantage of the properties and structures of the model under investigation. Also, it generally does not require specific initial iterates to guarantee its satisfactory numerical performance. All these advantages make the ADMM a benchmark algorithm in various areas such as image processing, statistical learning, and so on; we refer to [19, 77] for some review papers on the ADMM. In particular, the ADMM and its variants have been applied to solve

some optimal control problems constrained by time-independent PDEs in, e.g.,[4, 91, 162]. In [84], the ADMM was applied to parabolic optimal control problems with state constraints, and its convergence is proved without any assumption on the existence and regularity of the Lagrange multiplier. In [83], the Peaceman–Rachford splitting method (see [142]) which is closely related to the ADMM was suggested to solve approximate controllability problems of parabolic equations numerically.

On the other hand, the ADMM is a first-order algorithm; hence its convergence is at most linear and it may not be efficient for finding very high-precision solutions. For a numerical scheme solving the problem (3.1)–(3.2), total errors consist of the discretization error resulted by discretizing the model and the iteration error resulted by solving the discretized model numerically. In general, first-order numerical schemes such as the backward Euler finite difference method or piecewise constant finite element method with the step size τ is implemented for the time discretization (see e.g., [83, 133]). As a result, the error order of the time discretization is $O(\tau)$ (see e.g., [133]) and this estimate may dominate the magnitude of the total error. For such cases, pursuing too high-precision solutions of the discretized model does not help reduce the total error and it is more appropriate to just apply a first-order algorithm to find a medium-precision solution of the discretized model. This also motivates us to consider the ADMM (3.6) for the problem (3.1)–(3.2).

3.2.2 Difficulties and goals

It is straightforward to obtain the ADMM (3.6) for the problem (3.1)–(3.2). But the scheme (3.6) is only conceptual, and it cannot be used immediately. As will be shown in Section 3.3, the z-subproblem (3.6b) is easy; its closed-form solution can be computed by the projection onto the admissible set U_{ad}. But the u-subproblem (3.6a) is essentially a standard unconstrained parabolic optimal control problem, and it can only be solved iteratively by certain existing algorithms. For instance, as studied in [81, 83], we can apply the conjugate gradient (CG) method to solve it. Clearly, solving (3.6a) dominates the computation of

each iteration of the ADMM (3.6). Notice that the dimensionality of the time-dependent u-subproblem (3.6a) after space-time discretization is inevitably high. Hence it is impractical to solve these subproblems too accurately. Meanwhile, there is indeed no necessity to pursue too accurate solutions for these subproblems, especially when the iterates are still far away from the solution point. Therefore, the subproblem (3.6a) should be solved iteratively and inexactly, and the implementation of the ADMM (3.6) must be embedded by an internal iterative process for the subproblem (3.6a). Interesting mathematical problems arise soon: How to determine an appropriate inexactness criterion to execute the internal iterations for solving the subproblem (3.6a); and how to rigorously prove the convergence for the ADMM scheme (3.6) with two-layer nested iterations?

Preferably, the inexactness criterion for solving the subproblem (3.6a) should be easy to implement, free of setting empirically perceived constant accuracy a prior, independent of space-time discretization mesh sizes and the regularization parameter α, accurate enough to yield good approximate solutions which are good enough to ensure the overall convergence, yet efficient to avoid unnecessarily too accurate solutions so as to save overall computation. Moreover, though the convergence of the original ADMM has been well studied in both earlier literatures [67, 69, 76, 80] and recent literatures [96, 97], the scheme (3.6) with the nested internal iterations subject to a given inexactness criterion should be analyzed from scratch. In short, our goals are: (I) proposing an easily implementable and appropriately accurate inexactness criterion for solving the subproblem (3.6a) inexactly and hence an inexact version of the ADMM (3.6), (II) establishing the convergence for the resulting inexact ADMM rigorously, (III) specifying the inexact ADMM as concrete algorithms that are applicable to the problem (3.1)–(3.2), and (IV) extending the inexact ADMM to other versions that can be used for a range of other optimal control problems.

3.3 An inexact ADMM

In this section, we first take a closer look at the solutions of the subproblems (3.6a)–(3.6c), and then propose an inexactness criterion for solving the subprob-

lem (3.6a) iteratively. An inexact version of the ADMM (3.6) with two-layer nested iterations is thus proposed. For the simplicity of notations, hereinafter, we denote by U and Y the space $L^2(\mathcal{O})$ and $L^2(Q)$, respectively.

3.3.1 Elaboration of subproblems

Subproblem (3.6a)

For the u-subproblem (3.6a), it follows from

$$L_\beta(u, z^k, \lambda^k) = \tilde{J}(u) - (\lambda^k, u - z^k) + \frac{\beta}{2}\|u - z^k\|^2,$$

that the u-subproblem (3.6a) is equivalent to the following unconstrained parabolic optimal control problem:

$$\min_{u\in U} j_k(u) := \tilde{J}(u) - (\lambda^k, u - z^k) + \frac{\beta}{2}\|u - z^k\|^2.$$

Let $Dj_k(u)$ be the first-order derivative of j_k at u. By perturbation analysis discussed in [81, 83], we have

$$Dj_k(u) = u + p|_{\mathcal{O}} + \beta(u - z^k) - \lambda^k.$$

Hereafter, p is the adjoint variable associated with u and it is obtained from the successive solution of the following two parabolic equations:

$$\frac{\partial y}{\partial t} - \nu\Delta y + a_0 y = u\chi_{\mathcal{O}} \text{ in } \Omega \times (0, T), \quad y = 0 \text{ on } \Gamma \times (0, T), \quad y(0) = \varphi, \tag{3.7}$$

and

$$-\frac{\partial p}{\partial t} - \nu\Delta p + a_0 p = \gamma(y - y_d) \text{ in } \Omega \times (0, T), \quad p = 0 \text{ on } \Gamma \times (0, T), \quad p(T) = 0. \tag{3.8}$$

It is clear that the equation (3.7) is just the state equation (3.2) and it can be characterized by the operator S with $y = S(u)$. Furthermore, we denote by S^* the adjoint operator of S. Then, it is easy to derive that $S^* : L^2(Q) \longrightarrow L^2(\mathcal{O})$ satisfies $p|_{\mathcal{O}} = S^*(\gamma(y - y_d))$, where p is the solution of the adjoint equation (3.8). Then, we obtain the following first-order optimality condition of the u-subproblem (3.6a).

Theorem 3.1. *Let u^{k+1} be the unique solution of the subproblem (3.6a). Then, u^{k+1} satisfies*

$$Dj_k(u^{k+1}) = u^{k+1} + p^{k+1}|_{\mathcal{O}} + \beta(u^{k+1} - z^k) - \lambda^k = 0, \tag{3.9}$$

where p^{k+1} is the adjoint variable associated with u^{k+1}.

Remark on β

According to (3.9), $Dj_k(u^{k+1})$ consists of the minimization of $\tilde{J}(u)$ and the satisfaction of the constraint on the control variable. It is natural to consider choosing some value that is not different from 1 for β so that these two objectives can be well balanced. Our numerical experiments show that, $\beta = 2$ or 3, is usually a good choice to generate robust and fast numerical performance. Also, because of this reason, we reformulate the original problem (3.1)–(3.2) as (3.4) with a scaled objective functional $\tilde{J}(u)$. If no scaling is considered, it is easy to show that the optimality condition of the corresponding u-subproblem reads

$$\alpha(u^{k+1} + p^{k+1}|_{\mathcal{O}}) + \beta(u^{k+1} - z^k) - \lambda^k = 0, \tag{3.10}$$

and it implies that the penalty parameter β should be close to α in order to balance the two objectives in (3.10). Since α is generally very small (e.g., less than 10^{-3}), β is also forced to be small for this case. According to our numerical experiments, too small values of β may easily cause some stability and round-off problems in numerical implementation, and they also easily result in unbalanced magnitudes for the primal variables u and z, and the dual variable λ. All these issues are inclined to deteriorate convergence of the ADMM.

Subproblem (3.6b)

For the z-subproblem (3.6b), notice that

$$L_\beta(u^{k+1}, z, \lambda^k) = \tilde{J}(u^{k+1}) + I_{U_{ad}}(z) - (\lambda^k, u^{k+1} - z) + \frac{\beta}{2}\|u^{k+1} - z\|^2,$$

which implies that

$$z^{k+1} = \arg\min_{z \in U} I_{U_{ad}}(z) - (\lambda^k, u^{k+1} - z) + \frac{\beta}{2}\|u^{k+1} - z\|^2.$$

Hence, z^{k+1} is given by

$$z^{k+1} = P_{U_{ad}}(u^{k+1} - \frac{\lambda^k}{\beta}), \tag{3.11}$$

where $P_{U_{ad}}(\cdot)$ denotes the projection onto the admissible set U_{ad}:

$$P_{U_{ad}}(v) := \max\{a, \min\{v, b\}\}, \forall v \in U.$$

3.3.2 Inexactness criterion

In this subsection, we propose an inexactness criterion that achieves the mentioned goals, and an inexact version of the ADMM (3.6) is obtained for the problem(3.1)–(3.2). Various inexact versions of the ADMM in different settings can be found in the literature. For example, inexact versions of the ADMM for the generic case have been discussed in [52, 54, 137, 189]. These works require summable conditions on the sequence of accuracy (represented in terms of either the absolute or relative errors). Such a condition forces the subproblems to be solved with increasing accuracy and requires specifying the accuracy (indeed an infinite series of constants) a prior; both are difficult to be realized practically. A particular inexact version is the so-called proximal ADMM in, e.g., [22, 92], which adds appropriate quadratic terms to regularize the subproblems and may alleviate these subproblems for some cases by specifying the proximal terms appropriately. Because of the different and much more difficult setting in the problem (3.1)–(3.2), however, a specific criterion tailored for the subproblem (3.6a) should be found in order to solve it more efficiently.

Recall that the optimality condition of the u-subproblem (3.6a) can be characterized by (3.9). Since the u-subproblem (3.6a) is strongly convex, the above necessary condition is also sufficient. Therefore, if $\tilde{u} \in U$ satisfies $Dj_k(\tilde{u}) = 0$, then $\tilde{u}$ is the unique solution of the u-subproblem (3.6a). To propose an inexactness criterion, we define $e_k(u)$ as

$$e_k(u) := (1 + \beta)u + S^*(\gamma(S(u) - y_d)) - \beta z^k - \lambda^k. \tag{3.12}$$

It follows from the definitions of the solution operator S and its adjoint operator S^* that $e_k(u)$ can be written as

$$e_k(u) = (1 + \beta)u + p|_{\mathcal{O}} - \beta z^k - \lambda^k, \tag{3.13}$$

where p is the adjoint variable associated with u.

It is clear that $e_k(u) = Dj_k(u)$ and u^{k+1} is the solution of the u-subproblem (3.6a) at the $(k+1)$-th iteration if and only if $e_k(u^{k+1}) = 0$. Hence, we can use $e_k(u)$ as a residual for the u-subproblem (3.6a). With the help of $e_k(u)$, we propose the following inexactness criterion. For a given constant σ satisfying

$$0 < \sigma < \frac{\sqrt{2}}{\sqrt{2} + \sqrt{\beta}} \in (0, 1), \tag{3.14}$$

we compute u^{k+1} such that

$$\|e_k(u^{k+1})\| \leq \sigma \|e_k(u^k)\|. \tag{3.15}$$

The inexactness criterion (3.15) is mainly inspired by our previous work [190], and it keeps all advantageous features of the criterion in [190]. Meanwhile, the problem (3.1)–(3.2) in an infinite-dimensional Hilbert space is much more complicated than the LASSO model considered in [190], and it is worthy to elaborate on the details of executing the inexactness criterion (3.15). Indeed, the residual $e_k(u)$ in (3.13) is derived from the first-order derivative of $j_k(u)$. Conceptually, the computation of $e_k(u)$ requires the solutions of the state equation (3.2) and the adjoint equation (3.8). Practically, the residual $e_k(u)$ can be calculated easily by certain iterative scheme, see Algorithm 3.2 for the detail of implementing the CG method.

Remark 3.1. *We reiterate that the inexactness criterion (3.15) can be checked by current iterates and it can be executed automatically during iterations. There is no need to set any empirically perceived constant accuracy a prior, and it is independent of the mesh sizes for discretization. Also, the relative error* $\|e_k(u^{k+1})\|/\|e_k(u^k)\|$ *is controlled by the constant* σ *(instead of summable sequences as proposed in many ADMM literatures) and it does not need to tend to zero (hence, increasing accuracy can be avoided in iterations). All these features make the inexactness criterion (3.15) easily implementable and more likely to save computation.*

3.3.3 An inexact version of the ADMM (3.6) for (3.1)–(3.2)

Based on the previous discussion, an inexact version of the ADMM (3.6) with the inexactness criterion (3.15) can be proposed for the problem (3.1)–(3.2).

Algorithm 3.1 An Inexact Version of the ADMM (3.6) for (3.1)–(3.2)

Input: $\{u^0, z^0, \lambda^0\}^\top \in U \times U \times U$, $\beta > 0$ and $0 < \sigma < \frac{\sqrt{2}}{\sqrt{2}+\sqrt{\beta}} \in (0, 1)$.

while not converged **do**

Compute $e_k(u^k) = (1+\beta)u^k + p^k|_{\mathcal{O}} - \beta z^k - \lambda^k$.

Find u^{k+1} such that

$$\|e_k(u^{k+1})\| \leq \sigma \|e_k(u^k)\|, \text{ with } e_k(u^{k+1}) = (1+\beta)u^{k+1} + p^{k+1}|_{\mathcal{O}} - \beta z^k - \lambda^k.$$

Update the variable z^{k+1}: $z^{k+1} = P_{U_{ad}}(u^{k+1} - \frac{\lambda^k}{\beta})$.

Update the Lagrange multiplier λ^{k+1}: $\lambda^{k+1} = \lambda^k - \beta(u^{k+1} - z^{k+1})$.

end while

3.4 Convergence analysis

In this section, we prove the strong global convergence for Algorithm 3.1. Though there are many works in the literature studying the convergence of the ADMM and its variants, the convergence of Algorithm 3.1 should be proved from scratch because of the specific inexactness criterion (3.15) and the setting of the problem (3.1)–(3.2). In particular, the proof is essentially different from that in [190], despite of some common ideas in the respective stopping criteria. Note that the strong global convergence to be obtained is because of the strong convexity of the objective functional $\tilde{J}(u)$ in (3.4), which is usually absent for many other problems such as the LASSO model considered in [190].

3.4.1 Preliminary

To present our analysis in a compact form, we denote $w \in W := U \times U \times U$, $v \in V := U \times U$ and the function $F(w)$ as follows:

$$w = \begin{pmatrix} u \\ z \\ \lambda \end{pmatrix}, v = \begin{pmatrix} z \\ \lambda \end{pmatrix}, \text{ and } F(w) = \begin{pmatrix} D\tilde{J}(u) - \lambda \\ \lambda \\ u - z \end{pmatrix}, \tag{3.16}$$

where $D\tilde{J}(u)$ is the first-order derivative of $\tilde{J}(u)$. We also define the norm

$$\|v\|_H = \sqrt{(v, Hv)} := \sqrt{\beta\|z\|^2 + \frac{1}{\beta}\|\lambda\|^2}, \quad \forall v \in V, \tag{3.17}$$

which is induced by the matrix operator

$$H = \begin{pmatrix} \beta I & 0 \\ 0 & \frac{1}{\beta} I \end{pmatrix}.$$

With these notations, it is easy to see that the problem (3.4) can be characterized as the following variational inequality: find $w^* = (u^*, z^*, \lambda^*)^\top \in W$ such that

$$\text{VI}(W, U_{ad}, F): \quad I_{U_{ad}}(z) - I_{U_{ad}}(z^*) + (w - w^*, F(w^*)) \geq 0, \ \forall w \in W. \tag{3.18}$$

We denote by W^* the solution set of the variational inequality (3.18); and it is easy to show that the solution set W^* is a singleton.

From the definition of $\tilde{J}$ in (3.5), we know that it is strongly convex, i.e.

$$\|u - v\|^2 \leq (u - v, D\tilde{J}(u) - D\tilde{J}(v)), \ \forall u, v \in U. \tag{3.19}$$

In addition, one can show that $D\tilde{J}$ is Lipschitz continuous. Indeed, one has

$$D\tilde{J}(u) = u + p|_{\mathcal{O}},$$

where p is the adjoint variable associated with u. We introduce a linear operator $\bar{S} : U \longrightarrow Y$ such that

$$S(v) = \bar{S}v + S(0), \quad \forall v \in U. \tag{3.20}$$

Then, we can derive that

$$(u - v, D\tilde{J}(u) - D\tilde{J}(v)) \leq \kappa\|u - v\|^2, \ \forall u, v \in U, \tag{3.21}$$

where $\kappa = 1 + \gamma\|\bar{S}^*\bar{S}\|$.

3.4.2 Optimality conditions

Recall that in Algorithm 3.1, the u-subproblem (3.6a) is inexactly solved subject to the inexactness criterion (3.15), and the z-subproblem (3.6b) and λ-subproblem (3.6c) can be solved exactly. Hence, for the sequence $w^{k+1} = (u^{k+1}, z^{k+1}, \lambda^{k+1})^\top$ generated by Algorithm 3.1, the first-order optimality conditions can be expressed as:

$$\begin{cases} D_u L_\beta(u^{k+1}, z^k, \lambda^k) = e_k(u^{k+1}), & \text{(3.22a)} \\ I_{U_{ad}}(z) - I_{U_{ad}}(z^{k+1}) + (z - z^{k+1}, \lambda^k - \beta(u^{k+1} - z^{k+1})) \geq 0, \ \forall z \in U_{ad}, & \text{(3.22b)} \\ \lambda^{k+1} = \lambda^k - \beta(u^{k+1} - z^{k+1}), & \text{(3.22c)} \end{cases}$$

where $D_u L_\beta(u^{k+1}, z^k, \lambda^k)$ is the first-order partial derivative of $L_\beta(u, z, \lambda)$ with respect to u at $\left(u^{k+1}, z^k, \lambda^k\right)^\top$.

To prove the convergence of Algorithm 3.1, it is crucial to analyze the residual $e_k(u^{k+1})$. It follows from (3.13) and (3.15) that

$$\begin{aligned} \|e_k(u^{k+1})\| \leq \sigma \|e_k(u^k)\| &= \sigma \|e_{k-1}(u^k) + \beta z^{k-1} + \lambda^{k-1} - \beta z^k - \lambda^k\| \\ &\leq \sigma \|e_{k-1}(u^k)\| + \sigma \|\beta z^{k-1} + \lambda^{k-1} - \beta z^k - \lambda^k\|. \end{aligned} \tag{3.23}$$

In addition, it follows from (3.22b) that

$$I_{U_{ad}}(z^k) - I_{U_{ad}}(z^{k+1}) + (z^k - z^{k+1}, \lambda^k - \beta(u^{k+1} - z^{k+1})) \geq 0, \tag{3.24}$$

and

$$I_{U_{ad}}(z^{k+1}) - I_{U_{ad}}(z^k) + (z^{k+1} - z^k, \lambda^{k-1} - \beta(u^k - z^k)) \geq 0. \tag{3.25}$$

Adding (3.24) and (3.25) together, we have

$$(z^{k+1} - z^k, \lambda^{k+1} - \lambda^k) \leq 0. \tag{3.26}$$

Then, it follows from (3.23) and (3.26) that

$$\begin{aligned} \|e_k(u^{k+1})\| \leq & \sigma \|e_{k-1}(u^k)\| + \sigma \left(\|\beta z^{k-1} - \beta z^k\|^2 + \|\lambda^{k-1} - \lambda^k\|^2\right)^{\frac{1}{2}} \\ = & \sigma \|e_{k-1}(u^k)\| + \sigma \sqrt{\beta} \|v^k - v^{k-1}\|_H. \end{aligned} \tag{3.27}$$

Moreover, we note that the condition (3.14) implies that

$$0 < \frac{\beta}{2} \frac{\sigma^2}{(1-\sigma)^2} = \left(\frac{\sigma}{2(1-\sigma)}\right) \left(\frac{\beta\sigma}{1-\sigma}\right) < 1,$$

then there exits a constant $\mu > 0$ such that

$$(1 - \frac{\mu}{2} \frac{\sigma}{1-\sigma}) > 0 \quad \text{and} \quad (1 - \frac{1}{\mu} \frac{\sigma}{1-\sigma} \beta) > 0. \tag{3.28}$$

These inequalities will be used later.

3.4.3 Convergence

With above preparations, we are now in a position to prove the convergence for Algorithm 3.1. To simplify the notation, let us introduce an auxiliary variable $\bar{w}^k$ as

$$\bar{w}^k = \begin{pmatrix} \bar{u}^k \\ \bar{z}^k \\ \bar{\lambda}^k \end{pmatrix} = \begin{pmatrix} u^{k+1} \\ z^{k+1} \\ \lambda^k - \beta(u^{k+1} - z^k) \end{pmatrix}. \tag{3.29}$$

The role of $\bar{w}^k$ is just for simplifying the notation in our analysis; it is not required to be computed for implementing Algorithm 3.1. Next, we prove some results which will be useful in the following discussion.

First of all, we analyze how different the point $\bar{w}^k$ defined in (3.29) is from the solution w^* of (3.18) and how to quantify this difference by iterates generated by Algorithm 3.1.

Lemma 3.1. *Let $\{w^k\} = \{(u^k, z^k, \lambda^k)^\top\}$ be the sequence generated by Algorithm 3.1 and $\{\bar{w}^k\} = \{(\bar{u}^k, \bar{z}^k, \bar{\lambda}^k)^\top\}$ be defined as in (3.29). Then, for all $w \in W$, one has*

$$\begin{aligned} &I_{U_{ad}}(\bar{z}^k) - I_{U_{ad}}(z) + (\bar{w}^k - w, F(\bar{w}^k)) \\ &\leq \frac{1}{2}\left(\|v^k - v\|_H^2 - \|v^{k+1} - v\|_H^2 - \|v^k - v^{k+1}\|_H^2\right) + \left(u^{k+1} - u, D_u L_\beta(u^{k+1}, z^k, \lambda^k)\right). \end{aligned} \tag{3.30}$$

Proof. We first rewrite $D_u L_\beta\left(u^{k+1}, u^k, \lambda^k\right)$ as

$$D_u L_\beta(u^{k+1}, z^k, \lambda^k) = D\tilde{J}(u^{k+1}) - (\lambda^k - \beta(u^{k+1} - z^k)) = D\tilde{J}(u^{k+1}) - \bar{\lambda}^k,$$

with which we obtain, for all $w \in W$, that

$$\begin{aligned} &I_{U_{ad}}(z) - I_{U_{ad}}(\bar{z}^k) + (w - \bar{w}^k, F(\bar{w}^k)) \\ &= (u - u^{k+1}, D\tilde{J}(u^{k+1}) - \bar{\lambda}^k) \\ &\quad + I_{U_{ad}}(z) - I_{U_{ad}}(z^{k+1}) + (z - z^{k+1}, \bar{\lambda}^k) + (\lambda - \bar{\lambda}^k, u^{k+1} - z^{k+1}) \\ &= (u - u^{k+1}, D_u L_\beta(u^{k+1}, z^k, \lambda^k)) + (z - z^{k+1}, \lambda^k - \beta(u^{k+1} - z^{k+1})) \\ &\quad + I_{U_{ad}}(z) - I_{U_{ad}}(z^{k+1}) + \beta(z - z^{k+1}, z^k - z^{k+1}) + \frac{1}{\beta}(\lambda - \bar{\lambda}^k, \lambda^k - \lambda^{k+1}) \\ &\overset{(3.22b)}{\geq} (u - u^{k+1}, D_u L_\beta(u^{k+1}, z^k, \lambda^k)) + \beta(z - z^{k+1}, z^k - z^{k+1}) \\ &\quad + \frac{1}{\beta}(\lambda - \lambda^{k+1}, \lambda^k - \lambda^{k+1}) + \frac{1}{\beta}(\lambda^{k+1} - \bar{\lambda}^k, \lambda^k - \lambda^{k+1}). \end{aligned} \tag{3.31}$$

Applying the identity

$$(a-c,b-c)=\frac{1}{2}\left(\|a-c\|^2-\|a-b\|^2+\|b-c\|^2\right) \tag{3.32}$$

to (3.31), we have

$$\begin{aligned}
& I_{U_{ad}}(z)-I_{U_{ad}}(\bar{z}^k)+(w-\bar{w}^k,F(\bar{w}^k)) \\
\overset{(3.32)}{\geq}\ & (u-u^{k+1},D_uL_\beta(u^{k+1},z^k,\lambda^k))+\frac{\beta}{2}\left(\|z-z^{k+1}\|^2-\|z-z^k\|^2+\|z^k-z^{k+1}\|^2\right) \\
& +\frac{1}{2\beta}\left(\|\lambda-\lambda^{k+1}\|^2-\|\lambda-\lambda^k\|^2+\|\lambda^k-\lambda^{k+1}\|^2\right)-(z^k-z^{k+1},\lambda^k-\lambda^{k+1}) \\
\overset{(3.26)}{\geq}\ & (u-u^{k+1},D_uL_\beta(u^{k+1},z^k,\lambda^k))+\frac{\beta}{2}\left(\|z-z^{k+1}\|^2-\|z-z^k\|^2+\|z^k-z^{k+1}\|^2\right) \\
& +\frac{1}{2\beta}\left(\|\lambda-\lambda^{k+1}\|^2-\|\lambda-\lambda^k\|^2+\|\lambda^k-\lambda^{k+1}\|^2\right),\quad \forall w\in W.
\end{aligned} \tag{3.33}$$

Using the definition of H-norm in (3.17), the result (3.33) can be rewritten as (3.30) and the proof is complete. □

The difference between the inequality (3.30) and the variational inequality reformulation (3.18) reflects the difference of the point $\bar{w}^k$ from the solution point w^*. For the right-hand side of (3.30), the first three terms are quadratic and they are easy to manipulate over different indicators by algebraic operations, but it is not that explicit how the last crossing term can be controlled towards the eventual goal of proving the convergence of the sequence $\{w^k\}$. We thus look into this term particularly and show that the sum of these crossing terms over K iterations can be bounded by some quadratic terms as well. This result is summarized in the following lemma.

Lemma 3.2. *Let $\{w^k\}=\{(u^k,z^k,\lambda^k)^\top\}$ be the sequence generated by Algorithm 3.1. For any integer $K>0$ and μ satisfying (3.28), one has*

$$\begin{aligned}
& \sum_{k=1}^{K}(u^{k+1}-u,D_uL_\beta(u^{k+1},z^k,\lambda^k)) \\
\leq & \frac{\mu}{2}\sum_{k=1}^{K}\frac{\sigma}{1-\sigma}\|u^{k+1}-u\|^2+\frac{1}{2\mu}\sum_{i=1}^{K-1}\frac{\sigma}{1-\sigma}\beta\|v^i-v^{i+1}\|_H^2 \\
& +\frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left[\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right]^2,\forall u\in U.
\end{aligned} \tag{3.34}$$

Proof. First, it follows from (3.27) that

$$\|e_k(u^{k+1})\|\leq\sum_{i=0}^{k-1}\sigma^{k-i}\sqrt{\beta}\|v^i-v^{i+1}\|_H+\sigma^k\|e_0(u^1)\|. \tag{3.35}$$

From (3.22a) and (3.35), for any $\mu > 0$ satisfying (3.28) and $u \in U$, we have

$$\begin{aligned}
&\sum_{k=1}^{K}(u^{k+1}-u, D_u L_\beta(u^{k+1}, z^k, \lambda^k)) \le \sum_{k=1}^{K}\|u^{k+1}-u\|\|e_k(u^{k+1})\| \\
&\le \sum_{k=1}^{K}\sum_{i=0}^{k-1}\sigma^{k-i}\sqrt{\beta}\|u^{k+1}-u\|\|v^i-v^{i+1}\|_H + \sum_{k=1}^{K}\sigma^k\|u^{k+1}-u\|\|e_0(u^1)\| \\
&\le \sum_{k=1}^{K}\sum_{i=1}^{k-1}\sigma^{k-i}\sqrt{\beta}\|u^{k+1}-u\|\|v^i-v^{i+1}\|_H \\
&\qquad\qquad + \sum_{k=1}^{K}\sigma^k\|u^{k+1}-u\|\left[\|e_0(u^1)\| + \sqrt{\beta}\|v^0-v^1\|_H\right] \\
&\le \frac{\mu}{2}\sum_{k=1}^{K}\sum_{i=1}^{k-1}\sigma^{k-i}\|u^{k+1}-u\|^2 + \frac{1}{2\mu}\sum_{k=1}^{K}\sum_{i=1}^{k-1}\sigma^{k-i}\beta\|v^i-v^{i+1}\|_H^2 \\
&\quad + \frac{\mu}{2}\sum_{k=1}^{K}\sigma^k\|u^{k+1}-u\|^2 + \frac{1}{2\mu}\sum_{k=1}^{K}\sigma^k\left[\|e_0(u^1)\| + \sqrt{\beta}\|v^0-v^1\|_H\right]^2 \\
&= \frac{\mu}{2}\sum_{k=1}^{K}\sum_{i=0}^{k-1}\sigma^{k-i}\|u^{k+1}-u\|^2 + \frac{1}{2\mu}\sum_{k=1}^{K}\sum_{i=1}^{k-1}\sigma^{k-i}\beta\|v^i-v^{i+1}\|_H^2 \\
&\quad + \frac{1}{2\mu}\sum_{k=1}^{K}\sigma^k\left[\|e_0(u^1)\| + \sqrt{\beta}\|v^0-v^1\|_H\right]^2.
\end{aligned}$$

Then, we have

$$\begin{aligned}
&\sum_{k=1}^{K}\left(u^{k+1}-u, D_u L_\beta(u^{k+1}, z^k, \lambda^k)\right) \\
&\le \frac{\mu}{2}\sum_{k=1}^{K}\frac{\sigma-\sigma^{k+1}}{1-\sigma}\|u^{k+1}-u\|^2 + \frac{1}{2\mu}\sum_{i=1}^{K-1}\frac{\sigma-\sigma^{K-i+1}}{1-\sigma}\beta\|v^i-v^{i+1}\|_H^2 \\
&\quad + \frac{1}{2\mu}\frac{\sigma-\sigma^{K+1}}{1-\sigma}\left[\|e_0(u^1)\| + \sqrt{\beta}\|v^0-v^1\|_H\right]^2 \\
&\le \frac{\mu}{2}\sum_{k=1}^{K}\frac{\sigma}{1-\sigma}\|u^{k+1}-u\|^2 + \frac{1}{2\mu}\sum_{i=1}^{K-1}\frac{\sigma}{1-\sigma}\beta\|v^i-v^{i+1}\|_H^2 \\
&\quad + \frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left[\|e_0(u^1)\| + \sqrt{\beta}\|v^0-v^1\|_H\right]^2, \quad \forall u \in U.
\end{aligned}$$

We thus complete the proof. □

Now we can establish the strong global convergence of Algorithm 3.1.

Theorem 3.2. *Let $w^* = (u^*, z^*, \lambda^*)^\top$ be the solution point of the variational inequality (3.18) and $\{w^k\} = \{(u^k, z^k, \lambda^k)^\top\}$ be the sequence generated by Algorithm 3.1. Then, we have the following assertions:*

(1) $\|e_k(u^{k+1})\| \overset{k\to\infty}{\longrightarrow} 0, \quad \|z^k - z^{k+1}\| \overset{k\to\infty}{\longrightarrow} 0, \quad \|u^{k+1} - z^{k+1}\| \overset{k\to\infty}{\longrightarrow} 0;$

(2) $u^k \overset{k\to\infty}{\longrightarrow} u^*$, $z^k \overset{k\to\infty}{\longrightarrow} z^*$ *and* $\lambda^k \overset{k\to\infty}{\longrightarrow} \lambda^*$ *strongly in* U.

Proof. (1). First, it follows from (3.16), (3.19) and (3.29) that

$$(w - \bar{w}^k, F(w) - F(\bar{w}^k)) = (u - \bar{u}^k, D\tilde{J}(u) - D\tilde{J}(\bar{u}^k)) \geq \|u - u^{k+1}\|^2. \quad (3.36)$$

Then, using the results (3.30) and (3.34) established in Lemma 3.1 and Lemma 3.2, respectively, we obtain

$$\begin{aligned}
& \sum_{k=1}^{K} \left\{ I_{U_{ad}}(\bar{z}^k) - I_{U_{ad}}(z) + (\bar{w}^k - w, F(w)) \right\} \\
= \quad & \sum_{k=1}^{K} \left\{ I_{U_{ad}}(\bar{z}^k) - I_{U_{ad}}(z) + (\bar{w}^k - w, F(\bar{w}^k)) + (\bar{w}^k - w, F(w) - F(\bar{w}^k)) \right\} \\
\overset{(3.30)}{\leq} \quad & \frac{1}{2}\left(\|v^1 - v\|_H^2 - \|v^{K+1} - v\|_H^2\right) + \sum_{k=1}^{K} \left\{ (u^{k+1} - u, D_u L_\beta(u^{k+1}, z^k, \lambda^k)) \right. \\
& \left. -(w - \bar{w}^k, F(w) - F(\bar{w}^k)) \right\} - \sum_{k=1}^{K} \frac{1}{2}\|v^k - v^{k+1}\|_H^2 \\
\overset{(3.34)(3.36)}{\leq} \quad & \frac{1}{2}\left(\|v^1 - v\|_H^2 - \|v^{K+1} - v\|_H^2\right) + \sum_{k=1}^{K} \left(\frac{\mu}{2}\frac{\sigma}{1-\sigma} - 1 \right) \|u^{k+1} - u\|^2 \\
& + \sum_{k=1}^{K-1} \frac{1}{2}\left(\frac{\sigma}{1-\sigma}\frac{\beta}{\mu} - 1 \right) \|v^k - v^{k+1}\|_H^2 - \frac{1}{2}\|v^K - v^{K+1}\|_H^2 \\
& + \frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\| + \sqrt{\beta}\|v^0 - v^1\|_H \right)^2, \ \forall w \in W. \qquad (3.37)
\end{aligned}$$

For the solution point w^*, we have

$$I_{U_{ad}}(\bar{z}^k) - I_{U_{ad}}(z^*) + (\bar{w}^k - w^*, F(w^*)) \geq 0, \ \forall k \geq 1.$$

Setting $w = w^*$ in (3.37), together with the above property, for any integer $K > 1$, we have

$$\begin{aligned}
& \sum_{k=1}^{K} \left(1 - \frac{\mu}{2}\frac{\sigma}{1-\sigma} \right) \|u^{k+1} - u^*\|^2 + \sum_{k=1}^{K-1} \left(\frac{1}{2} - \frac{\beta}{2\mu}\frac{\sigma}{1-\sigma} \right) \|v^k - v^{k+1}\|_H^2 \\
\leq \ & \frac{1}{2}\|v^1 - v^*\|_H^2 + \frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\| + \sqrt{\beta}\|v^0 - v^1\|_H \right)^2 \\
& - \frac{1}{2}\|v^{K+1} - v^*\|_H^2 - \frac{1}{2}\|v^K - v^{K+1}\|_H^2.
\end{aligned} \quad (3.38)$$

It follows from (3.28) that

$$(1-\frac{\mu}{2}\frac{\sigma}{1-\sigma})>0 \quad \text{and} \quad (1-\frac{1}{\mu}\frac{\sigma}{1-\sigma}\beta)>0.$$

Then, the inequality (3.38) implies

$$\|u^{k+1}-u^*\| \overset{k\to\infty}{\longrightarrow} 0 \text{ and } \|v^{k+1}-v^k\|_H \overset{k\to\infty}{\longrightarrow} 0. \tag{3.39}$$

For any $\varepsilon>0$, there exists k_0, such that for all $k\geq k_0$, we have $\|v^{k+1}-v^k\|_H<\varepsilon$ and $\sigma^k<\varepsilon$. Then, for all $k\geq k_0$, it follows from (3.35) that

$$\begin{aligned}
\|e_k(u^{k+1})\| &\leq \sum_{i=0}^{k-1}\sigma^{k-i}\sqrt{\beta}\|v^i-v^{i+1}\|_H+\sigma^k\|e_0(u^1)\| \\
&= \sum_{i=0}^{k_0-1}\sigma^{k-i}\sqrt{\beta}\|v^i-v^{i+1}\|_H+\sum_{i=k_0}^{k-1}\sigma^{k-i}\sqrt{\beta}\|v^i-v^{i+1}\|_H+\sigma^k\|e_0(u^1)\| \\
&\leq \big(\sqrt{\beta}\max_{0\leq i\leq k_0-1}\|v^i-v^{i+1}\|_H\sum_{i=0}^{k_0-1}\sigma^{k-k_0-i}\big)\sigma^{k_0}+\sigma^k\|e_0(u^1)\| \\
&\quad+\big(\sqrt{\beta}\max_{k_0\leq i\leq k-1}\|v^i-v^{i+1}\|_H\sum_{i=k_0}^{k-1}\sigma^{k-i}\big) \\
&\leq \varepsilon\big[\sqrt{\beta}\max_{0\leq i\leq k_0-1}\|v^i-v^{i+1}\|_H\sum_{i=0}^{k_0-1}\sigma^{k-k_0-i}+\sqrt{\beta}\sum_{i=k_0}^{k-1}\sigma^{k-i}+\|e_0(u^1)\|\big],
\end{aligned}$$

which implies that

$$\|e_k(u^{k+1})\| \overset{k\to+\infty}{\longrightarrow} 0.$$

In addition, since $\|v^{k+1}-v^k\|_H \overset{k\to\infty}{\longrightarrow} 0$, we conclude that

$$\|z^{k+1}-z^k\| \overset{k\to\infty}{\longrightarrow} 0 \quad \text{and} \quad \|\lambda^{k+1}-\lambda^k\| \overset{k\to\infty}{\longrightarrow} 0.$$

Then, from $\|u^{k+1}-z^{k+1}\|=\frac{1}{\beta}\|\lambda^{k+1}-\lambda^k\|$, we have $\|u^{k+1}-z^{k+1}\| \overset{k\to\infty}{\longrightarrow} 0$.

(2). From (3.39), we know that $u^k \overset{k\to\infty}{\longrightarrow} u^*$ strongly in U. Combining with $\|u^{k+1}-z^{k+1}\| \overset{k\to\infty}{\longrightarrow} 0$, one has $z^k \overset{k\to\infty}{\longrightarrow} z^*$ strongly in U. From (3.18), it is easy to verify that $\lambda^*=D\tilde{J}(u^*)$. On the other hand, one has

$$\lambda^k=D\tilde{J}(u^{k+1})+\beta(u^{k+1}-z^k)-e_k(u^{k+1}).$$

We thus have

$$\lambda^k-\lambda^*=D\tilde{J}(u^{k+1})-D\tilde{J}(u^*)+\beta(u^{k+1}-u^k)+\beta(u^k-z^k)-e_k(u^{k+1}).$$

Noting that $u^k \overset{k\to\infty}{\longrightarrow} u^*$, $u^k - z^k \overset{k\to\infty}{\longrightarrow} 0$, $e_k(u^{k+1}) \overset{k\to\infty}{\longrightarrow} 0$ and $D\tilde{J}$ is Lipschitz continuous (see (3.21)), we have

$$\lambda^k \overset{k\to\infty}{\longrightarrow} \lambda^* \text{ strongly in } U.$$

We thus complete the proof. □

Remark 3.2. *Clearly, it follows from Theorem 3.2 that the state variable $y^k = S(u^k)$ also converges strongly in Y to $y^* = S(u^*)$ since S is continuous.*

Remark 3.3. *Note that the convergence analysis for Algorithm 3.1 does not depend on how the inexactness criterion (3.15) is satisfied and what the specific form of the solution operator S is.*

3.5 Convergence rate

In [96, 97], the ADMM's $O(1/K)$ worst-case convergence rate in both the ergodic and non-ergodic senses have been initiated in the context of convex optimization with consideration of the Euclidean space, where K denotes the iteration counter. Recall that an $O(1/K)$ worst-case convergence rate means that an iterate, whose accuracy to the solution under certain criterion is of the order $O(1/K)$, can be found after K iterations of an iterative scheme. It can be alternatively explained as that it requires at most $O(1/\varepsilon)$ iterations to find an approximate solution with an accuracy of ε. This type of convergence rate is in the worst-case nature, and it provides a worst-case but universal estimate on the speed of convergence. Hence, it does not contradict with some much faster speeds which might be witnessed empirically for a specific application (as to be shown in Section 3.7). In this section, we extend these results to Algorithm 3.1 in an infinite-dimensional Hilbert space.

3.5.1 Ergodic convergence rate

In this subsection, we follow [96] to establish an $O(1/K)$ worst-case convergence rate in the ergodic sense for Algorithm 3.1. We first introduce a criterion

to measure the accuracy of an approximation of the variational inequality (3.18). As analyzed in [62, 96], the solution set W^* of the variational inequality (3.18) has the following characterization.

Theorem 3.3 (cf. [62])**.** *Let W^* be the solution set of the variational inequality (3.18). Then, we have*

$$W^* = \bigcap_{w \in W} \left\{ \hat{w} \in W : I_{U_{ad}}(z) - I_{U_{ad}}(\hat{z}) + (w - \hat{w}, F(w)) \geq 0 \right\}.$$

The above result indicates that $\hat{w} \in W$ is an approximate solution of the variational inequality (3.18) with an accuracy of $\varepsilon > 0$ if

$$I_{U_{ad}}(\hat{z}) - I_{U_{ad}}(z) + (\hat{w} - w, F(w)) \leq \varepsilon. \tag{3.40}$$

Next, we show an $O(1/K)$ worst-case convergence rate for Algorithm 3.1.

Theorem 3.4. *Let $\{w^k\} = \{(u^k, z^k, \lambda^k)^\top\}$ be the sequence generated by Algorithm 3.1; and $\{\bar{w}^k\} = \{(\bar{u}^k, \bar{z}^k, \bar{\lambda}^k)^\top\}$ be defined as in (3.29). For any integer $K \geq 1$, we further define*

$$\hat{w}_K = \frac{1}{K} \sum_{k=1}^{K} \bar{w}^k. \tag{3.41}$$

Then, for all $w \in W$, one has

$$\begin{aligned} & I_{U_{ad}}(\hat{z}_K) - I_{U_{ad}}(z) + (\hat{w}_K - w, F(w)) \\ & \leq \frac{1}{K} \left[\frac{1}{2\mu} \frac{\sigma}{1-\sigma} \left(\|e_0(u^1)\| + \sqrt{\beta} \|v^0 - v^1\|_H \right)^2 + \frac{1}{2} \|v^0 - v\|_H^2 \right]. \end{aligned}$$

Proof. Recall the inequality (3.37). We then have

$$\begin{aligned} & \sum_{k=1}^{K} \left\{ I_{U_{ad}}(\bar{z}^k) - I_{U_{ad}}(z) + (\bar{w}^k - w, F(w)) \right\} \\ \leq \quad & \frac{1}{2} \left(\|v^1 - v\|_H^2 - \|v^{K+1} - v\|_H^2 \right) + \sum_{k=1}^{K} \left(\frac{\mu}{2} \frac{\sigma}{1-\sigma} - 1 \right) \|u^{k+1} - u\|^2 \\ & + \sum_{k=1}^{K-1} \frac{1}{2} \left(\frac{\sigma}{1-\sigma} \frac{\beta}{\mu} - 1 \right) \|v^k - v^{k+1}\|_H^2 - \frac{1}{2} \|v^K - v^{K+1}\|_H^2 \\ & + \frac{1}{2\mu} \frac{\sigma}{1-\sigma} \left(\|e_0(u^1)\| + \sqrt{\beta} \|v^0 - v^1\|_H \right)^2. \end{aligned}$$

Then, using (3.28), for all $w \in W$, we have

$$I_{U_{ad}}(\hat{z}_K) - I_{U_{ad}}(z) + (\hat{w}_K - w, F(w))$$

$$\begin{aligned} &\overset{\text{Convexity}}{\leq} && \frac{1}{K}\sum_{k=1}^{K}\left\{I_{U_{ad}}(\bar{z}^k) - I_{U_{ad}}(z) + (\bar{w}^k - w, F(w))\right\} \\ &\leq && \frac{1}{K}\left[\frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\| + \sqrt{\beta}\|v^0 - v^1\|_H\right)^2 + \frac{1}{2}\|v^1 - v\|_H^2\right], \end{aligned}$$

which completes the proof. □

The above theorem shows that after K iterations, we can find an approximate solution of the variational inequality (3.18) with an accuracy of $O(1/K)$. This approximate solution is given in (3.41), and it is the average of all the points w^k which can be computed by all the known iterates generated by Algorithm 3.1. Hence, this is an $O(1/K)$ worst-case convergence rate in the ergodic sense for Algorithm 3.1.

3.5.2 Non-ergodic convergence rate

In this subsection, we extend the result in [97] to show an $O(1/K)$ worst-case convergence rate in the non-ergodic sense for Algorithm 3.1.

We first need to clarify a criterion to precisely measure the accuracy of an iterate to a solution point. It follows from (3.16) and (3.22) that for the iterate $(u^{k+1}, z^{k+1}, \lambda^{k+1})^\top$ generated by Algorithm 3.1, for all $w \in W$, one has

$$I_{U_{ad}}(z) - I_{U_{ad}}(z^{k+1}) + \left(w - w^{k+1}, F(w^{k+1})) + (w - w^{k+1}, \begin{pmatrix} -\beta\left(z^k - z^{k+1}\right) - e_k(u^{k+1}) \\ 0 \\ \frac{1}{\beta}\left(\lambda^{k+1} - \lambda^k\right) \end{pmatrix}\right) \geq 0.$$

Taking (3.18) into account, we can show that $(u^{k+1}, z^{k+1}, \lambda^{k+1})^\top$ is a solution point of (3.18) if and only if $\|v^k - v^{k+1}\|_H^2 = 0$ and $\|e_k(u^{k+1})\|^2 = 0$. Hence, it is reasonable to measure the accuracy of the iterate $(u^{k+1}, z^{k+1}, \lambda^{k+1})^\top$ by $\|v^k - v^{k+1}\|_H^2$ and $\|e_k(u^{k+1})\|^2$. Our purpose is thus to show that after K iterations of Algorithm 3.1, both $\|v^k - v^{k+1}\|_H^2$ and $\|e_k(u^{k+1})\|^2$ can be bounded by upper bounds in order of $O(1/K)$.

Theorem 3.5. *Let $\{w^k\} = \{(u^k, z^k, \lambda^k)^\top\}$ be the sequence generated by Algorithm 3.1. Then, for any integer $K \geq 1$, we have*

$$\min_{1\leq k\leq K}\left\{\|v^k - v^{k+1}\|_H^2\right\} \leq \frac{1}{K}\left[\frac{1}{\mu_0}\|v^1 - v^*\|_H^2 + \frac{1}{\mu_0\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\| + \sqrt{\beta}\|v^0 - v^1\|_H\right)^2\right], \tag{3.42}$$

and

$$\begin{aligned}
&\min_{1\le k\le K}\left\{\|e_k(u^{k+1})\|^2\right\}\\
\le\ & \frac{2}{K}\left\{\left(\frac{\sigma}{1-\sigma}\sqrt{\beta}\right)^2\left[\frac{1}{\mu_0}\|v^1-v^*\|_H^2+\frac{1}{\mu_0\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2\right]\right\}\\
&+\frac{2}{K^2}\left[\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2\right], \qquad (3.43)
\end{aligned}$$

where w^* is the solution point, μ satisfies (3.28) and $\mu_0=1-\frac{\beta}{\mu}\frac{\sigma}{1-\sigma}>0$.

Proof. According to the inequality (3.38) we obtain

$$\begin{aligned}
&\sum_{k=1}^{K+1}\left(1-\frac{\mu}{2}\frac{\sigma}{1-\sigma}\right)\|u^{k+1}-u^*\|^2+\sum_{k=1}^{K}\left(\frac{1}{2}-\frac{\beta}{2\mu}\frac{\sigma}{1-\sigma}\right)\|v^k-v^{k+1}\|_H^2\\
\le\ & \frac{1}{2}\|v^1-v^*\|_H^2+\frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2,
\end{aligned}$$

which implies that

$$\sum_{k=1}^{K}\left(\frac{1}{2}-\frac{\beta}{2\mu}\frac{\sigma}{1-\sigma}\right)\|v^k-v^{k+1}\|_H^2\le\frac{1}{2}\|v^1-v^*\|_H^2+\frac{1}{2\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2. \qquad (3.44)$$

Consequently, we have

$$\min_{1\le k\le K}\left\{\|v^k-v^{k+1}\|_H^2\right\}\le\frac{1}{K}\left[\frac{1}{\mu_0}\|v^1-v^*\|_H^2+\frac{1}{\mu_0\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2\right],$$

and the assertion (3.42) is proved. Note that μ_0 is positive following from (3.28).

In addition, it follows from the inequality (3.27) that

$$\|e_k(u^{k+1})\|\le\sum_{i=0}^{k-1}\sigma^{k-i}\sqrt{\beta}\|v^i-v^{i+1}\|_H+\sigma^k\|e_0(u^1)\|.$$

Summarizing the above inequality from $k=1$ to $k=K$, we obtain

$$\sum_{k=1}^{K}\|e_k(u^{k+1})\|\ \le\ \sum_{k=1}^{K}\left\{\sum_{i=0}^{k-1}\sigma^{k-i}\sqrt{\beta}\|v^i-v^{i+1}\|_H+\sigma^k\|e_0(u^1)\|\right\}. \qquad (3.45)$$

For the right-hand side of (3.45), we have the following estimate:

$$\sum_{k=1}^{K}\sum_{i=0}^{k-1}\sigma^{k-i}\sqrt{\beta}\cdot\|v^i-v^{i+1}\|_H$$

$$
\begin{aligned}
&= \sum_{i=0}^{K-1}\sum_{k=i+1}^{K}\sigma^{k-i}\sqrt{\beta}\cdot\|v^i-v^{i+1}\|_H \\
&= \sum_{i=0}^{K-1}\frac{\sigma-\sigma^{K-i+1}}{1-\sigma}\sqrt{\beta}\cdot\|v^i-v^{i+1}\|_H \\
&\le \sum_{i=0}^{K-1}\frac{\sigma}{1-\sigma}\sqrt{\beta}\cdot\|v^i-v^{i+1}\|_H,
\end{aligned}
$$

and

$$
\sum_{k=1}^{K}\sigma^k\|e_0(u^1)\| \le \frac{\sigma-\sigma^{K+1}}{1-\sigma}\|e_0(u^1)\| \le \frac{\sigma}{1-\sigma}\|e_0(u^1)\|.
$$

Then, by simple calculations, we have

$$
\begin{aligned}
&\left(\sum_{k=1}^{K}\|e_k(u^{k+1})\|\right)^2 \\
\overset{(3.45)}{\le}\ &\left(\sum_{i=0}^{K-1}\frac{\sigma}{1-\sigma}\sqrt{\beta}\cdot\|v^i-v^{i+1}\|_H+\frac{\sigma}{1-\sigma}\|e_0(u^1)\|\right)^2 \\
\le\ & 2\left(\sum_{i=1}^{K-1}\frac{\sigma}{1-\sigma}\sqrt{\beta}\cdot\|v^i-v^{i+1}\|_H\right)^2+2\left(\frac{\sigma}{1-\sigma}\cdot\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)\right)^2 \\
\le\ & 2\left(\frac{\sigma}{1-\sigma}\sqrt{\beta}\right)^2\cdot\left(\sum_{i=1}^{K}\|v^i-v^{i+1}\|_H\right)^2+2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2 \\
\le\ & 2\left(\frac{\sigma}{1-\sigma}\sqrt{\beta}\right)^2 K\cdot\left(\sum_{i=1}^{K}\|v^i-v^{i+1}\|_H^2\right)+2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2 \\
\overset{(3.44)}{\le}\ & 2\left(\frac{\sigma}{1-\sigma}\sqrt{\beta}\right)^2 K\cdot\left[\frac{1}{\mu_0}\|v^1-v^*\|_H^2+\frac{1}{\mu_0\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2\right] \\
&+2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2.
\end{aligned}
$$

Further, we can derive that

$$
\begin{aligned}
&\left(K\cdot\min_{1\le k\le K}\|e_k\left(u^{k+1}\right)\|\right)^2 \le \left(\sum_{k=1}^{K}\|e_k(u^{k+1})\|\right)^2 \\
\le\ & 2\left(\frac{\sigma}{1-\sigma}\sqrt{\beta}\right)^2 K\cdot\left[\frac{1}{\mu_0}\|v^1-v^*\|_H^2+\frac{1}{\mu_0\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2\right] \\
&+2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2,
\end{aligned}
$$

which implies

$$
\min_{1\le k\le K}\left\{\|e_k(u^{k+1})\|^2\right\}
$$

$$\leq \frac{1}{K}\left\{2\left(\frac{\sigma}{1-\sigma}\sqrt{\beta}\right)^2\cdot\left[\frac{1}{\mu_0}\|v^1-v^*\|_H^2+\frac{1}{\mu_0\mu}\frac{\sigma}{1-\sigma}\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2\right]\right\}$$
$$+\frac{1}{K^2}\left[2\left(\frac{\sigma}{1-\sigma}\right)^2\cdot\left(\|e_0(u^1)\|+\sqrt{\beta}\|v^0-v^1\|_H\right)^2\right].$$

We thus complete the proof. □

We note that both values in the right-hand sides of (3.42) and (3.43) are order of $O(1/K)$. Therefore, this theorem provides an $O(1/K)$ worst-case convergence rate in the non-ergodic sense for Algorithm 3.1.

3.6 Implementation of Algorithm 3.1

In this section, we discuss how to execute the inexactness criterion (3.15) so as to specify Algorithm 3.1 as a concrete algorithm for the problem (3.1)–(3.2), and delineate the implementation details.

Indeed, the u-subproblem (3.6a) is a typical unconstrained parabolic optimal control problem and various numerical methods in the literature can be applied. Whichever such method is applied, we should and only need to ensure that the inexactness criterion (3.15) is satisfied in order to guarantee the overall convergence of Algorithm 3.1. Below we illustrate by the CG method how to execute the inexactness criterion (3.15) in the inner-layer iterations. Recall that the u-subproblem (3.6a) is

$$u^{k+1}=\arg\min_{u\in U}\ j_k(u)=\frac{\gamma}{2}\|S(u)-y_d\|^2+\frac{1}{2}\|u\|^2-(\lambda^k,u-z^k)+\frac{\beta}{2}\|u-z^k\|^2,\ (3.46)$$

and the associated optimality condition is given in Theorem 3.1. Next, we show that the optimality condition of the problem (3.46) can be characterized by a symmetric and positive definite linear system, hence the CG method can be applied. To this end, we first recall that the linear operator $\bar{S}$ defined in (3.20) satisfies

$$S(v)=\bar{S}v+S(0),\quad \forall v\in U.$$

Then, $y=\bar{S}u$ is equivalent to the following equation:

$$\frac{\partial y}{\partial t}-\nu\Delta y+a_0y=u\chi_{\mathcal{O}}\text{ in }\Omega\times(0,T),\quad y=0\text{ on }\Gamma\times(0,T),\quad y(0)=0.$$

In addition, it is easy to show that the adjoint operator $\bar{S}^* : Y \longrightarrow U$ satisfies $\bar{S}^* y = p|_{\mathcal{O}}$, where p solves

$$-\frac{\partial p}{\partial t} - \nu \Delta p + a_0 p = y \text{ in } \Omega \times (0,T), \quad p = 0 \text{ on } \Gamma \times (0,T), \quad p(T) = 0.$$

Hence, the u-subproblem (3.46) can be reformulated as

$$u^{k+1} = \arg\min_{u \in U} \; j_k(u) = \frac{\gamma}{2}\|\bar{S}u + S(0) - y_d\|^2 + \frac{1}{2}\|u\|^2 - (\lambda^k, u - z^k) + \frac{\beta}{2}\|u - z^k\|^2,$$

and the corresponding optimality condition is

$$(1 + \beta + \gamma \bar{S}^* \bar{S}) u^{k+1} + \gamma \bar{S}^* (S(0) - y_d) - \lambda^k - \beta z^k = 0. \tag{3.47}$$

Note that (3.47) is a symmetric and positive definite linear system of u^{k+1} and the CG method can be applied. Obviously, at each iteration of Algorithm 1, we need to solve a linear system discretized from (3.47), with the same coefficient matrix, but different right-hand sides. Hence, a uniform preconditioner can be applied when certain iterative method (e.g., CG method) is employed to solve these linear systems. Recall that if SSN methods are applied, the coefficient matrices of the resulting Newton systems vary iteratively and preconditioners should also be adjusted iteratively. This is a major difference of the ADMM from SSN methods for the problem (3.1)–(3.2) .

With the inexactness criterion (3.15), the CG method for solving the u-subproblem (3.6a) is presented in Algorithm 3.2. Compared with the classical CG method (see e.g., [75, Chapter 3] and [78, Chapter 2]), Algorithm 3.2 requires updating the adjoint variable p to verify the specific inexactness criterion (3.15). It is clear that the update of p_{m+1}^k can be computed cheaply. Hence, our proposed inexactness criterion (3.15) can be verified by negligible extra computation. More discussions, including the convergence properties of CG type methods applied to the solution of linear systems in Hilbert spaces, can also be found in the mentioned references.

Now, with these discussions, Algorithm 3.1 can be specified as an ADMM–CG two-layer nested iterative scheme for the problem (3.1)–(3.2). We list it as Algorithm 3.3.

Remark 3.4. *As mentioned, to execute the inexactness criterion (3.15), the CG method can be replaced by other numerical schemes such as the preconditioned*

Algorithm 3.2 CG for the u-subproblem (3.6a)

Input $u_0^k = u^k, p_0^k = p^k$. Compute $g_0^k = u_0^k + p_0^k|_{\mathcal{O}} + \beta(u_0^k - z^k) - \lambda^k$, set $w_0^k = g_0^k$ and $e_k(u^k) = g_0^k$.

while $\|e_k(u_m^k)\| > \sigma\|e_k(u^k)\|$ **do**

Solving $\bar{y}_m^k = \bar{S}w_m^k$ and $\bar{p}_m^k|_{\mathcal{O}} = \bar{S}^*(\gamma\bar{y}_m^k)$. Then compute the step size:

$$\rho_m^k = \frac{(g_m^k, w_m^k)}{(\bar{g}_m^k, w_m^k)}, \quad \text{with} \quad \bar{g}_m^k = (1+\beta)w_m^k + \bar{p}_m^k|_{\mathcal{O}}.$$

Update u, p, the gradient g and the residual $e_k(u_{m+1}^k)$ via:

$$u_{m+1}^k = u_m^k - \rho_m^k w_m^k, \quad p_{m+1}^k = p_m^k - \rho_m^k \bar{p}_m^k,$$
$$g_{m+1}^k = g_m^k - \rho_m^k \bar{g}_m^k, \quad e_k(u_{m+1}^k) = g_{m+1}^k.$$

Compute $r_m^k = \|g_{m+1}^k\|^2/\|g_m^k\|^2$, and then update $w_{m+1}^k = g_{m+1}^k + r_m^k w_m^k$.

end while

Output $u^{k+1} = u_{m+1}^k$ and $p^{k+1} = p_{m+1}^k$.

Algorithm 3.3 An ADMM–CG two-layer nested iterative scheme for the problem (3.1)–(3.2).

Output: $\{u^0, z^0, \lambda^0\}^\top$ in $U \times U \times U$, $\beta > 0$ and $0 < \sigma < \frac{\sqrt{2}}{\sqrt{2}+\sqrt{\beta}} \in (0, 1)$.

for $k \geq 0$ **do** $\{u^k, z^k, \lambda^k\} \to u^{k+1} \to z^{k+1} \to \lambda^{k+1}$ via

Compute u^{k+1} by the CG method in Algorithm 3.2;

Compute z^{k+1} by (3.11);

Update the Lagrange multiplier $\lambda^{k+1} = \lambda^k - \beta(u^{k+1} - z^{k+1})$.

end for

MinRes method in [144] which has been verified to be efficient for unconstrained parabolic optimal control problems. Hence, depending on how to satisfy the inexactness criterion (3.15) internally, Algorithm 3.1 can be specified as various algorithms.

3.7 Numerical results of Algorithm 3.3 for (3.1)–(3.2)

In this section, we report some preliminary numerical results to validate the efficiency of Algorithm 3.3 for the parabolic optimal control problem (3.1)–(3.2). All codes were written in MATLAB R2016b and numerical experiments were conducted on a Surface Pro 5 laptop with 64-bit Windows 10.0 operation system, Intel(R) Core(TM) i7-7660U CPU (2.50 GHz), and 16 GB RAM.

First, for numerical discretization, we employ the backward Euler finite difference method (with step size τ) for the time discretization and piecewise linear finite element method (with mesh size h) for the space discretization. In order to implement (3.11), we perform at each time step a *nodal projection* of the continuous piecewise affine function $\left(u_h^{k+1} - \frac{\lambda_h^k}{\beta}\right)(n\tau)(\in V_{0h})$ over the convex set $U_{ad} \cap V_{0h}$, where $U_{ad} = \{\phi | \phi \in L^2(\Omega), a \leq \phi \leq b\}$, and (assuming that Ω is a bounded polygonal domain of $\mathbb{R}^2$)

$$V_{0h} = \{\phi | \phi \in C^0(\bar{\Omega}), \phi|_{\mathbb{T}} \in P_1, \forall \mathbb{T} \in \mathcal{T}_h, \phi|_{\Gamma} = 0\}.$$

Here, $\mathcal{T}_h$ is a triangulation of Ω and P_1 is the space of the polynomial functions of two variables of degree ≤ 1. In addition, we denote by $P^{\text{nodal}}_{U_{ad}\cap V_{0h}}$ the above projection operator, which is defined by

$$\begin{cases} P^{\text{nodal}}_{U_{ad}\cap V_{0h}}(\phi) \in U_{ad} \cap V_{0h}, \forall \phi \in V_{0h}, \\ P^{\text{nodal}}_{U_{ad}\cap V_{0h}}(\phi)(Q_k) = \max\{a, \min\{b, \phi(Q_k)\}\}, \forall k = 1, \cdots, N_{0h}. \end{cases} \tag{3.48}$$

In (3.48), $\{Q_k\}_{k=1}^{N_{0h}}$ is the set of the vertices of triangulation $\mathcal{T}_h$ not located on Γ. This nodal projection can facilitate the implementation of Algorithm 3.3; and we refer to Remark 5 in [84] for more discussions.

For the linear systems arising at each time step of the discretized parabolic equations, they are solved by the permuted LDL factorization in, e.g., [159], because the coefficient matrices are sparse and invariant. Other methodologies such as Krylov subspace methods, domain decomposition methods and multigrid methods can also be applied to further improve the numerical efficiency. In addition, an adjoint approach is employed for the u-subproblem (3.6a), which

requires storing the solution of the state equation (3.2) at each time step. This is a demanding request on memory, and it may not be applicable for, e.g., time-dependent problems in three-dimensional space, due to the huge scale of systems after discretization. To tackle this issue, some memory saving methodologies can be embedded into our algorithmic design. All these numerical techniques are important but beyond the scope of our discussion; we refer to [158] for fast linear algebra solvers and [13] for a memory saving strategy.

To test the efficiency of Algorithm 3, the primal residual π_s and dual residual d_s are respectively defined as

$$\pi_s = \frac{\|z^k - z^{k-1}\|_{L^2(\mathcal{O})}}{\|z^{k-1}\|_{L^2(\mathcal{O})}}, \quad d_s = \frac{\|u^k - z^k\|_{L^2(\mathcal{O})}}{\max\{\|u^{k-1}\|_{L^2(\mathcal{O})}, \|z^{k-1}\|_{L^2(\mathcal{O})}\}}.$$

The stopping criterion for all numerical experiments is

$$\max\{\pi_s, d_s\} \leq tol,$$

where $tol > 0$ is a prescribed tolerance. The initial values are set as $u = 0, z = 0$ and $\lambda = 0$ in the following discussion. For the constant σ in the inexactness criterion (3.15), according to (3.14), we choose $\sigma = 0.99\frac{\sqrt{2}}{\sqrt{2}+\sqrt{\beta}}$ because larger values of σ mean that the criterion is looser and hence less computation is needed for solving the subproblems. In addition, we define the relative distance "RelDis" and the objective functional value "Obj" as

$$\text{RelDis} := \|y - y_d\|^2_{L^2(Q)}/\|y_d\|^2_{L^2(Q)} \quad \text{and} \quad \text{Obj} := \frac{1}{2}\|y - y_d\|^2_{L^2(Q)} + \frac{\alpha}{2}\|u\|^2_{L^2(\mathcal{O})},$$

to verify the accuracy of the numerical solution.

Example 1. We consider an example of the problem (3.1)–(3.2) with a known exact solution; it is a variant of the problem discussed in [1]. The model is

$$\begin{aligned} \min_{u\in U_{ad}, y\in L^2(Q)} \quad & \frac{1}{2}\iint_Q |y - y_d|^2 dxdt + \frac{\alpha}{2}\iint_Q |u|^2 dxdt \\ \text{s.t.} \quad & \begin{cases} \dfrac{\partial y}{\partial t} - \Delta y = f + u, & \text{in} \quad \Omega \times (0,T), \\ y = 0, & \text{on} \quad \Gamma \times (0,T), \\ y(0) = \varphi, \end{cases} \end{aligned} \tag{3.49}$$

with $\Omega = (0,1)^2$, $\omega = \Omega$, $T = 1$. In (3.49), the function $f \in L^2(Q)$ is a source term that helps us construct the exact solution without affection to the numerical

implementation. We further let

$$\begin{cases} y = (1-t)\sin\pi x_1 \sin\pi x_2,\ p = \alpha(1-t)\sin 2\pi x_1 \sin 2\pi x_2,\ \varphi = \sin\pi x_1 \sin\pi x_2, \\ f = -u + \dfrac{\partial y}{\partial t} - \Delta y, \quad y_d = y + \dfrac{\partial p}{\partial t} + \Delta p, \quad u = \min(a, \max(b, -\dfrac{p}{\alpha})). \end{cases}$$

Then, it is easy to verify that $(u^*, y^*) := (u, y)$ is the optimal solution of the problem (3.49). Moreover, the admissible set is

$$U_{ad} = \{v | v \in L^\infty(\mathcal{O}), -0.5 \leq v(x_1, x_2; t) \leq 0.5 \text{ a.e. in } \mathcal{O}\} \subset L^2(\mathcal{O}).$$

We set the regularization parameter $\alpha = 10^{-5}$ throughout.

We first test Algorithm 3.3 with different values of β to show how its performance depends on the choice of β. As discussed in Section 3.3, β should be close to 1 to balance the minimization of $\tilde{J}(u)$ and the satisfaction of the control constraint $u \in U_{ad}$. On the other hand, it is clear that the system (3.47) becomes increasingly ill-conditioned as β decreases; and a smaller β tends to result in slower convergence for the CG method. As a result, the trade-off between the inexactness criterion (3.15) and the conditioning of the u-subproblem (3.6a) should also be considered for choosing β. The results with $\tau = h = 2^{-6}$ and different values of β are reported in Table 3.1, in which the notation "ADMM$_{Iter}$" represents the total out-layer ADMM iteration numbers, "Mean/Max CG" denote the average and maximum steps of the inner CG method, respectively. Results in Table 3.1 empirically show that $\beta = 2$ or $\beta = 3$ is a good choice. In the following, we choose $\beta = 3$.

Table 3.1: Numerical results of Algorithm 3.3 with different β for Example 1.

β	0.1	0.5	1	2	3	4	5
ADMM$_{Iter}$	297	60	29	20	22	25	29
Mean/Max CG	6.01/10	7.80/10	7.48/10	6.75/9	6.00/8	5.36/7	4.97/7

Next, we validate the efficiency of the inexactness criterion (3.15). We compare Algorithm 3.3 with the intuitive implementation of the ADMM (3.6) whose accuracy for solving the u-subproblem (3.6a) by the CG method is empirically set as a constant a prior. For this set of numerical experiments, $tol = 10^{-4}$ and various space mesh sizes h and time step sizes τ as $h = \tau = 2^{-i}$ with $i = 5, 6, 7, 8$, are considered. The accuracy for solving the u-subproblem (3.6a)

is $\|e_k(u^k_{m+1})\| \leq 10^{-j}$ with j an integer. We test various values for the accuracy constant: $j = 2, 4, 6, 8, 10$, which represent from low to very high levels of accuracy. Numerical results are reported in Table 3.2, in which "ADMM_{1e-j}" denotes the accuracy constant for solving the u-subproblem (3.6a) is 10^{-j}. Here and in what follows, the notation "$\sim$" means that the ADMM does not converge within 500 iterations.

Table 3.2: Numerical comparison of Algorithm 3.3 and ADMM_{1e-k} for Example 1.

Mesh	Algorithm	ADMM_{Iter}	Mean/Max CG	Time (s)	RelDis	Obj
2^{-5}	ADMM_{1e-10}	21	61.71/83	17.49	7.5987×10^{-7}	3.6825×10^{-7}
	ADMM_{1e-8}	21	44.81/65	16.94	7.5987×10^{-7}	3.6825×10^{-7}
	ADMM_{1e-6}	21	28.47/49	8.59	7.5986×10^{-7}	3.6825×10^{-7}
	ADMM_{1e-4}	21	13.30/32	4.23	7.5990×10^{-7}	3.6825×10^{-7}
	ADMM_{1e-2}	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
	Algorithm 3	24	5.88/8	1.93	7.5954×10^{-7}	3.6823×10^{-7}
2^{-6}	ADMM_{1e-10}	19	60.20/94	196.68	6.7055×10^{-7}	3.5036×10^{-7}
	ADMM_{1e-8}	19	45.05/71	170.48	6.7055×10^{-7}	3.5036×10^{-7}
	ADMM_{1e-6}	19	27.47/48	93.65	6.7055×10^{-7}	3.5036×10^{-7}
	ADMM_{1e-4}	19	12.84/31	46.79	6.7056×10^{-7}	3.5035×10^{-7}
	ADMM_{1e-2}	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
	Algorithm 3	22	6.00/8	20.86	6.7075×10^{-7}	3.5035×10^{-7}
2^{-7}	ADMM_{1e-10}	19	59.10/93	3372.30	6.5295×10^{-7}	3.4473×10^{-7}
	ADMM_{1e-8}	19	44.25/70	2884.61	6.5295×10^{-7}	3.4473×10^{-7}
	ADMM_{1e-6}	19	27.15/48	1653.10	6.5295×10^{-7}	3.4473×10^{-7}
	ADMM_{1e-4}	19	12.70/30	793.48	6.5299×10^{-7}	3.4473×10^{-7}
	ADMM_{1e-2}	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
	Algorithm 3	20	6.20/8	307.06	6.5294×10^{-7}	3.4473×10^{-7}
2^{-8}	ADMM_{1e-10}	19	58.30/93	37106.76	6.4876×10^{-7}	3.4260×10^{-7}
	ADMM_{1e-8}	19	43.45/70	26570.61	6.4876×10^{-7}	3.4260×10^{-7}
	ADMM_{1e-6}	19	26.95/48	15801.55	6.4876×10^{-7}	3.4260×10^{-7}
	ADMM_{1e-4}	19	12.55/30	7627.94	6.4879×10^{-7}	3.4260×10^{-7}
	ADMM_{1e-2}	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
	Algorithm 3	20	6.05/8	3839.67	6.4877×10^{-7}	3.4260×10^{-7}

According to Table 3.2, the automatically adjustable inexactness criterion (3.15) is favorable for the implementation of ADMM (3.6). If the accuracy is set as a constant a prior, then it is not easy to probe an appropriate value. An either too large or too small value may result in troubles. For a too large value, e.g., 10^{-2}, the accuracy for solving the subproblems may not be sufficient and the convergence may not be guaranteed. For a too small value, e.g., 10^{-8} or 10^{-10}, the accuracy for solving the subproblems may be unnecessarily high and it does not help accelerate the overall convergence. Especially, if the mesh size for discretization is small, then the resulting u-subproblem is high dimensional and it becomes less practical to solve it to a high precision. For the cases tested, retrospectively, the accuracy 10^{-4} is a good choice. But there is neither theory nor hint to fathom this value a prior. Indeed, as to be shown in Example 2, this

value could be heavily dependent on the specific problem under discussion. The inexactness criterion (3.15), however, can find an appropriate accuracy automatically for finding an approximate solution of the u-subproblem (3.6a). Hence, Algorithm 3.3 does not have these mentioned difficulties, and it generally works well for all the tested cases. Table 3.2 also shows that the efficiency of Algorithm 3.3 is independent from the mesh size used for discretization. This is an important feature to guarantee the numerical efficiency when an algorithm is applied to the discretized version of some model with fine mesh for discretization, as mentioned in some well-known works such as [14, 102, 103, 114].

Table 3.3: Numerical errors of Algorithm 3.3 with $\beta = 3$ and $tol = 10^{-4}$ for Example 1.

error	$h = \tau = 2^{-5}$	$h = \tau = 2^{-6}$	$h = \tau = 2^{-7}$	$h = \tau = 2^{-8}$
$\|u - u^*\|_{L^2(\mathcal{O})}$	1.8421×10^{-2}	4.6767×10^{-3}	1.1715×10^{-3}	2.9013×10^{-4}
$\|y - y^*\|_{L^2(Q)}$	3.6426×10^{-5}	8.6088×10^{-6}	2.1106×10^{-6}	4.9269×10^{-7}

Since the ADMM (3.6) is a first-order algorithm and generally it is not favorable to generate iterates in very high precisions, it is necessary to verify if the ADMM (3.6) can be accurate enough to guarantee the iterative accuracy. In other words, whether or not it is still the discretization error that constitutes the main part of the total error when the ADMM (3.6) is applied to the discretized version of the problem (3.49). Recall that the solution of Example 1 is known. In Table 3.3, we report the L^2-error for the iterate (u, y) obtained by Algorithm 3.3 for various values of h and τ. For succinctness, we only give the results for the case where $\beta = 3$ and $tol = 10^{-4}$. It is clear from Table 3.3 that, when the ADMM (3.6) is applied to the problem (3.49), the iterative accuracy is sufficient and the overall error of u and y are both dominated by the discretization error.

Evolutions of the residuals and objective functional values with respect to the outer ADMM iterations are displayed in Figure 3.1. These curves indicate the fast convergence of Algorithm 3.3. In addition, the state variable y and the control variable u, and the errors $y^* - y$ and $u^* - u$ at $t = 0.25$ with $h = \tau = 2^{-6}$ are depicted in Figures 3.2 and 3.3, respectively.

Example 2. We consider another case of the problem (3.1)–(3.2) where the control region ω is a subset of the domain Ω. Let $\Omega = (0, 1)^2$, $\omega = (0, 0.25)^2 \subsetneq \Omega$

Figure 3.1: Residuals (left) and objective functional values (right) with respect to outer ADMM iterations for Example 1.

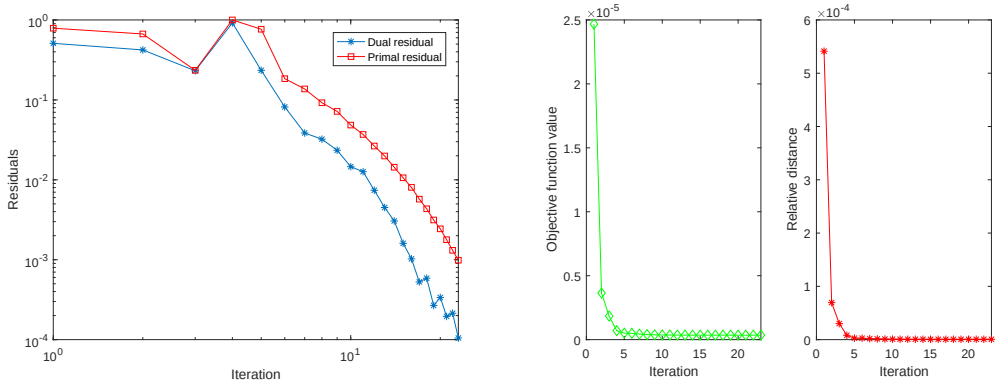


Figure 3.2: Numerical solutions y (left) and u (right) at $t = 0.25$ for Example 1.

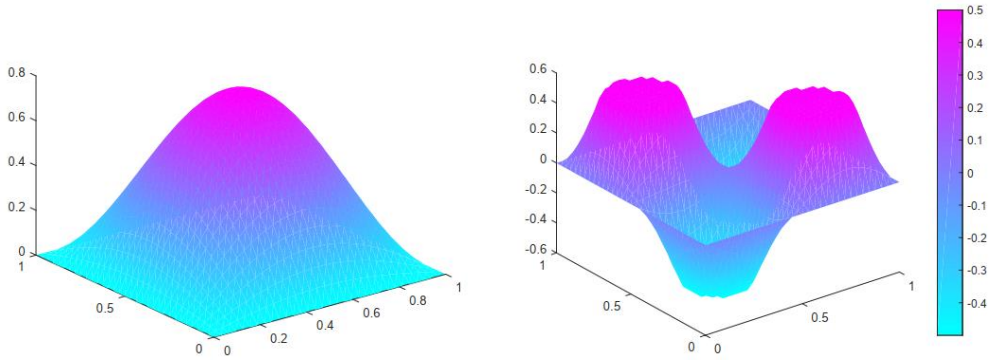
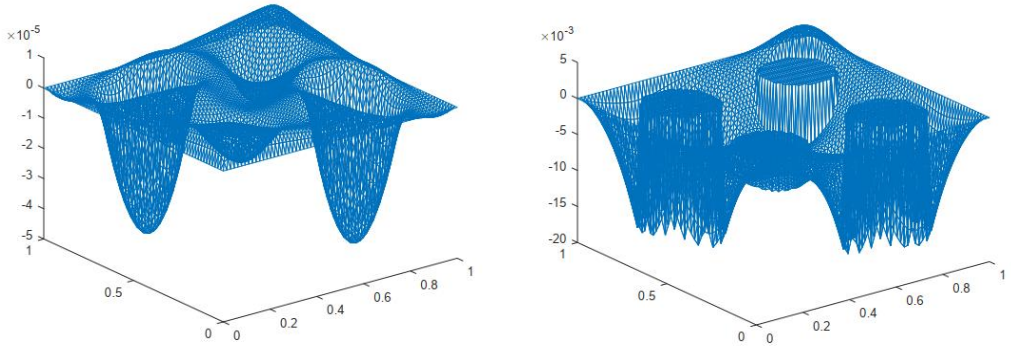


Figure 3.3: Errors $y^* - y$ (left) and $u^* - u$ (right) at $t = 0.25$ for Example 1.



and $Q = \Omega \times (0,T), \mathcal{O} = \omega \times (0,T)$ with $T = 1$. The regularization parameter $\alpha = 10^{-6}$ and the admissible set is defined as

$$U_{ad} = \{v | v \in L^\infty(\mathcal{O}), -300 \leq v(x;t) \leq 300 \text{ a.e. in } \mathcal{O}\} \subset L^2(\mathcal{O}).$$

The target function y_d is given by $y_d = e^t \sin 4x_1 \sin 4x_2$, and the coefficients $\nu = a_0 = 1$.

We set $\beta = 3$ and $tol = 10^{-3}$ throughout, and test various choices of the mesh size. The numerical results are summarized in Table 3.4. Residuals and the objective functional values are plotted in Figure 3.4; numerical results for y and u with $h = \tau = 2^{-6}$ at $t = 0.5$ are presented in Figure 3.5. We observe that Algorithm 3.3 is also very efficient and robust for the small control region case; and solving the u-subproblem (3.6a) subject to the inexactness criterion (3.15) reduces the computational cost significantly. Similar conclusions as those for Example 1 can be drawn for this example.

Figure 3.4: Residuals (left) and objective functional values (right) with respect to outer ADMM iterations for Example 2.

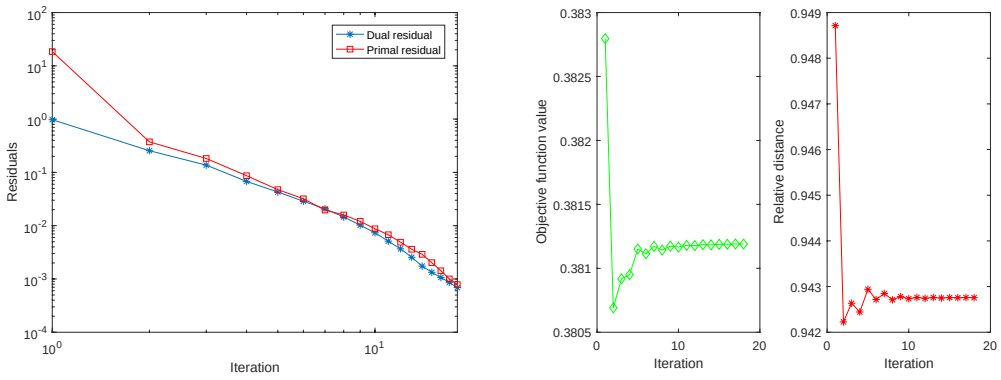


Figure 3.5: Numerical solutions y (left) and u (right) at $t = 0.5$ for Example 2.

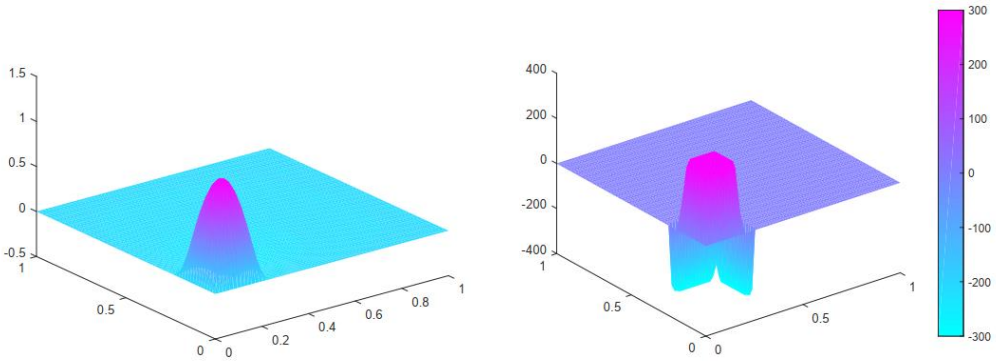


Table 3.4: Numerical comparison of Algorithm 3.3 and ADMM_{1e-k} for Example 2.

Mesh	Algorithm	ADMM_{Iter}	Mean/Max CG	Time (s)	RelDis	Obj
2^{-5}	ADMM_{1e-10}	14	51.50/62	8.56	0.9388	0.3726
	ADMM_{1e-8}	14	41.86/52	6.64	0.9388	0.3726
	ADMM_{1e-6}	14	32.64/43	5.34	0.9388	0.3726
	ADMM_{1e-4}	14	23.00/32	3.70	0.9388	0.3726
	ADMM_{1e-2}	14	13.71/23	2.24	0.9388	0.3726
	Algorithm 3	17	3.35/4	0.83	0.9388	0.3726
2^{-6}	ADMM_{1e-10}	16	51.63/62	110.05	0.9428	0.3812
	ADMM_{1e-8}	16	41.88/52	85.20	0.9428	0.3812
	ADMM_{1e-6}	16	32.31/43	64.43	0.9428	0.3812
	ADMM_{1e-4}	16	22.81/33	45.82	0.9428	0.3812
	ADMM_{1e-2}	16	13.25/23	27.15	0.9428	0.3812
	Algorithm 3	18	3.39/4	9.29	0.9428	0.3812
2^{-7}	ADMM_{1e-10}	16	50.50/61	1834.32	0.9455	0.3821
	ADMM_{1e-8}	16	41.25/52	1550.68	0.9455	0.3821
	ADMM_{1e-6}	16	31.81/42	1291.11	0.9455	0.3821
	ADMM_{1e-4}	16	22.13/32	883.59	0.9455	0.3821
	ADMM_{1e-2}	16	12.81/23	401.55	0.9455	0.3821
	Algorithm 3	18	3.33/4	129.33	0.9455	0.3821
2^{-8}	ADMM_{1e-10}	16	49.69/60	22540.18	0.9470	0.3817
	ADMM_{1e-8}	16	40.44/51	18869.58	0.9470	0.3817
	ADMM_{1e-6}	16	31.25/41	14969.83	0.9470	0.3817
	ADMM_{1e-4}	16	22.06/32	10437.38	0.9470	0.3817
	ADMM_{1e-2}	16	12.63/22	6281.95	0.9470	0.3817
	Algorithm 3	18	3.33/4	1609.73	0.9470	0.3817

3.8 Extensions

In previous sections, our discussion is focused on the parabolic optimal control problem with control constraints (3.1)–(3.2) in order to expose our main ideas clearly. The discussion can be easily extended to various other optimal control problems. For instances, the objective functional in (3.1) can be replaced by the L^1-control cost functional in [161], and the control variable u can be replaced by the Neumann or Dirichlet boundary control variable in [83]. In addition, note that both of the proposed algorithmic design and the theoretical analysis are independent of the specific form of the solution operator S defined in (3.3), and they can be extended to various optimal control problems constrained by other linear PDEs. To be more concrete, it is clear that the definition of $e_k(u)$ in (3.12) is originated from the optimality system of (3.6a), and it only requires that the solution operator S be affine (i.e., the linearity of the state equation (3.2)). Hence, the parabolic state equation in (3.2) can be replaced by, e.g., the elliptic equation [147], the wave equation [82], the convection-diffusion equation [83], or the fractional parabolic equation [27]. In this section, we consider an optimal control problem constrained by the wave equation to delineate the extensions. Some notations and discussions analogous to previous ones are not repeated for succinctness.

3.8.1 Model

We consider the following optimal control problem with control constraints:

$$\min_{u\in U_{ad}, y\in L^2(Q)} \quad \frac{1}{2}\iint_Q |y-y_d|^2 dxdt + \frac{\alpha}{2}\iint_{\mathcal{O}} |u|^2 dxdt, \tag{3.50}$$

and it is subject to the wave equation

$$\frac{\partial^2 y}{\partial t^2} - \Delta y = u\chi_{\mathcal{O}} \text{ in } \Omega\times(0,T), \quad y = 0 \text{ on } \Gamma\times(0,T), \quad y(0) = y_0,\ \frac{\partial y}{\partial t}(0) = y_1. \tag{3.51}$$

Notation in (3.50)–(3.51) is the same as that in (3.1)–(3.2) except that the initial conditions $y_0 \in H_0^1(\Omega)$ and $y_1 \in L^2(\Omega)$. For the existence, uniqueness, and regularity of the solution of (3.50)–(3.51), we refer to, e.g., [125].

For the special case of (3.50)–(3.51) where $d = 1$ or $\omega = \Omega$, SSN type methods have been studied in the literature, see, e.g., [114, 122, 129]. For the general case of (3.50)–(3.51) where $d \geq 2$ and $\omega \subsetneq \Omega$, similar difficulties as those mentioned in the introduction for the problem (3.1)–(3.2) arise if SSN type methods are applied. Below, we briefly show the details of extending Algorithm 3.1 to the general case of (3.50)–(3.51).

3.8.2 Algorithm

Similarly, the direct implementation of ADMM to the problem (3.50)–(3.51) reads as

$$\begin{cases} u^{k+1} = \arg\min\limits_{u \in L^2(\mathcal{O})} \bar{L}_\beta(u, z^k, \lambda^k), & \text{(3.52a)} \\ z^{k+1} = \arg\min\limits_{z \in L^2(\mathcal{O})} \bar{L}_\beta(u^{k+1}, z, \lambda^k), & \text{(3.52b)} \\ \lambda^{k+1} = \lambda^k - \beta(u^{k+1} - z^{k+1}), & \text{(3.52c)} \end{cases}$$

where the augmented Lagrangian functional $\bar{L}_\beta(u, z, \lambda)$ has the same form as the $L_\beta(u, z, \lambda)$ in (3.6) except that the solution operator S is associated with the wave equation (3.51) instead of the parabolic equation (3.2).

For the z-subproblem (3.52b), it amounts to computing the projection onto the admissible set U_{ad}; and the u-subproblem (3.52a) is an unconstrained optimal control problem subject to the wave equation (3.51). Note that the u-subproblem (3.52a) shares the same numerical challenges as the subproblem (3.6a); we may apply the CG method such as [83] to solve it iteratively at each iteration. To propose the inexactness criterion, we first need to introduce a residual $e_k(u)$ for the u-subproblem (3.52a) as we have done in Section 3.3. For this purpose, inspired by (3.12), we define $e_k(u)$ as

$$e_k(u) := (1 + \beta)u + S^*(\frac{1}{\alpha}(S(u) - y_d)) - \beta z^k - \lambda^k,$$

where $S : L^2(\mathcal{O}) \longrightarrow L^2(Q)$ is the solution operator associated with the wave equation (3.51) and $S^* : L^2(Q) \longrightarrow L^2(\mathcal{O})$ is the adjoint operator of S. It is easy to show that

$$e_k(u) = (1 + \beta)u + p|_{\mathcal{O}} - \beta z^k - \lambda^k, \tag{3.53}$$

where p is the successive solution of the wave equation (3.51) and the following adjoint equation:

$$\frac{\partial^2 p}{\partial t^2} - \Delta p = \frac{1}{\alpha}(y - y_d) \text{ in } \Omega \times (0,T), \quad p = 0 \text{ on } \Gamma \times (0,T), \quad p(T) = 0, \ \frac{\partial p}{\partial t}(T) = 0. \tag{3.54}$$

Then, the inexactness criterion for computing u^{k+1} in (3.52a) is

$$\|e_k(u^{k+1})\| \leq \sigma \|e_k(u^k)\|, \tag{3.55}$$

with the constant σ given in (3.14).

Although the same letter in (3.13) is used, the definition of $e_k(u)$ in (3.53) is determined by the wave equation (3.51) and the adjoint equation (3.54). It is thus different from (3.13) for the parabolic equation (3.2) and its adjoint equation (3.8). Embedding the inexactness criterion (3.55) into the ADMM scheme (3.52), an inexact version of the ADMM (3.52) similar as Algorithm 1 is readily available for the problem (3.50)–(3.51), and its convergence can be proved similarly. We omit the details.

3.8.3 Numerical results

We test the ADMM scheme (3.52) with the inexactness criterion (3.55), and report some preliminary numerical results for the problem (3.50)–(3.51) where $\omega \subsetneq \Omega$ and $d = 2$.

Example 3. Let us consider the following optimal control problem constrained by the wave equation with a known exact solution:

$$\begin{aligned} \min_{u \in U_{ad}, y \in L^2(Q)} \quad & \frac{1}{2} \iint_Q |y - y_d|^2 dxdt + \frac{\alpha}{2} \iint_{\mathcal{O}} |u|^2 dxdt \\ \text{s.t.} \quad & \begin{cases} \dfrac{\partial^2 y}{\partial t^2} - \Delta y = f + u\chi_{\mathcal{O}}, & \text{in} \quad \Omega \times (0,T), \\ y = 0, & \text{on} \quad \Gamma \times (0,T), \\ y(0) = y_0, \ \dfrac{\partial y}{\partial t}(0) = y_1, \end{cases} \end{aligned} \tag{3.56}$$

where $\Omega = (0,1)^2$, $T = 1$ and the control region $\omega = (0, 0.5)^2 \subsetneq \Omega$. In addition,

we set

$$\begin{cases} y = e^t \sin \pi x_1 \sin \pi x_2,\ p = \sqrt{\alpha}(t-T)^2 \sin \pi x_1 \sin \pi x_2, \\ u = \min(a, \max(b, -\dfrac{1}{\alpha} p|_{\mathcal{O}})),\ f = -u\chi_{\mathcal{O}} + \dfrac{\partial^2 y}{\partial t^2} - \Delta y, \\ y_d = y - \dfrac{\partial^2 p}{\partial t^2} + \Delta p,\ y_0 = \sin \pi x_1 \sin \pi x_2,\ y_1 = \sin \pi x_1 \sin \pi x_2. \end{cases}$$

It is easy to verify that $(u^*, y^*) := (u, y)$ is the solution point of the problem (3.56). Moreover, we set the regularization parameter $\alpha = 10^{-4}$ and

$$U_{ad} = \{v | v \in L^\infty(\mathcal{O}), -5 \leq v(x;t) \leq 0 \text{ a.e. in } \mathcal{O}\} \subset L^2(\mathcal{O}).$$

By implementing the CG method to solve the u-subproblem (3.52a) subject to the inexactness criterion (3.53), an ADMM–CG iterative scheme similar as Algorithm 3.3 can be obtained for the problem (3.50)–(3.51). For numerical discretization, we employ the central difference method (with step size τ) for the time discretization and piecewise linear finite element method (with mesh size h) for the space discretization. All notations and remarks in Section 3.7 are used here again. Let $\beta = 5$ and $tol = 10^{-5}$. We test the cases where the space mesh size h and the time step size τ are $h = \tau = 2^{-i}$ with $i = 5, 6, 7, 8$. Numerical results are presented in Table 3.5.

Table 3.5: Numerical comparison of ADMM–CG and ADMM_{1e-k} for Example 3.

Mesh	Algorithm	ADMM_{Iter}	Mean/Max CG	Time (s)	RelDis	Obj
2^{-5}	ADMM_{1e-10}	46	17.69/72	16.26	3.8248×10^{-3}	1.6716×10^{-3}
	ADMM_{1e-8}	46	12.75/31	11.89	3.8248×10^{-3}	1.6716×10^{-3}
	ADMM_{1e-6}	46	8.54/18	8.05	3.8248×10^{-3}	1.6716×10^{-3}
	ADMM_{1e-4}	46	4.89/11	4.86	3.8248×10^{-3}	1.6716×10^{-3}
	ADMM_{1e-2}	~	~	~	~	~
	ADMM–CG	46	1.96/2	2.23	3.8248×10^{-3}	1.6716×10^{-3}
2^{-6}	ADMM_{1e-10}	48	16.75/23	168.04	3.7670×10^{-3}	1.6197×10^{-3}
	ADMM_{1e-8}	48	12.85/20	109.32	3.7670×10^{-3}	1.6197×10^{-3}
	ADMM_{1e-6}	48	8.77/15	89.06	3.7670×10^{-3}	1.6197×10^{-3}
	ADMM_{1e-4}	48	5.00/11	54.21	3.7670×10^{-3}	1.6197×10^{-3}
	ADMM_{1e-2}	~	~	~	~	~
	ADMM–CG	49	1.96/2	24.29	3.7670×10^{-3}	1.6197×10^{-3}
2^{-7}	ADMM_{1e-10}	49	16.73/23	3511.81	3.7169×10^{-3}	1.5845×10^{-3}
	ADMM_{1e-8}	49	12.78/19	2198.52	3.7169×10^{-3}	1.5845×10^{-3}
	ADMM_{1e-6}	49	8.76/15	1814.87	3.7169×10^{-3}	1.5845×10^{-3}
	ADMM_{1e-4}	50	4.90/11	1131.26	3.7169×10^{-3}	1.5845×10^{-3}
	ADMM_{1e-2}	~	~	~	~	~
	ADMM–CG	50	1.96/2	415.58	3.7169×10^{-3}	1.5845×10^{-3}
2^{-8}	ADMM_{1e-10}	50	16.42/22	49802.84	3.6863×10^{-3}	1.5643×10^{-3}
	ADMM_{1e-8}	50	12.46/19	31824.46	3.6863×10^{-3}	1.5643×10^{-3}
	ADMM_{1e-6}	50	8.54/15	24823.09	3.6863×10^{-3}	1.5643×10^{-3}
	ADMM_{1e-4}	50	4.94/11	10533.96	3.6863×10^{-3}	1.5643×10^{-3}
	ADMM_{1e-2}	~	~	~	~	~
	ADMM–CG	51	1.96/2	4561.64	3.6863×10^{-3}	1.5643×10^{-3}

According to Table 3.5, the ADMM–CG iterative scheme is also very efficient for the general case of the problem (3.50)–(3.51) where $\omega \subsetneq \Omega$ and $d = 2$. Similar as the parabolic case, it suffices to solve the u-subproblem (3.52a) inexactly subject to the criterion (3.55). The independence of the convergence to the mesh size of discretization is also observed.

Evolutions of the residuals and objective functional values with respect to the outer ADMM iterations are plotted in Figure 3.6. These curves indicate the fast convergence of the ADMM–CG, despite the fact that the theoretical worst-case convergence rate is only $O(1/K)$. In addition, the iterative errors $\|y^k - y^*\|$ and $\|u^k - u^*\|$ in Figure 3.6 (right) show that the discretization errors dominate the total errors of the numerical solution. This means the ADMM–CG finds a rather precise iterative solution very fast. The control variable u, state variable y, and the errors $u^* - u$ and $y^* - y$ at $t = 0.75$ with $h = \tau = 2^{-6}$ are depicted in Figures 3.7 and 3.8, respectively.

Figure 3.6: Residuals (left), objective functional value (middle), and errors of u and y (right) with respect to the outer ADMM iterations for Example 3.

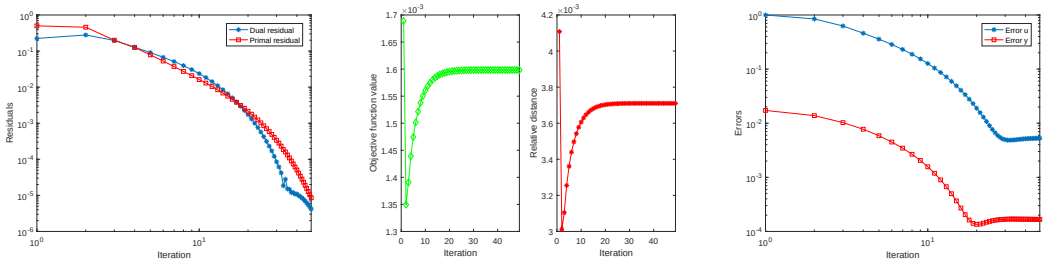


Figure 3.7: Numerical solutions u (left) and y (right) at $t = 0.75$ for Example 3.

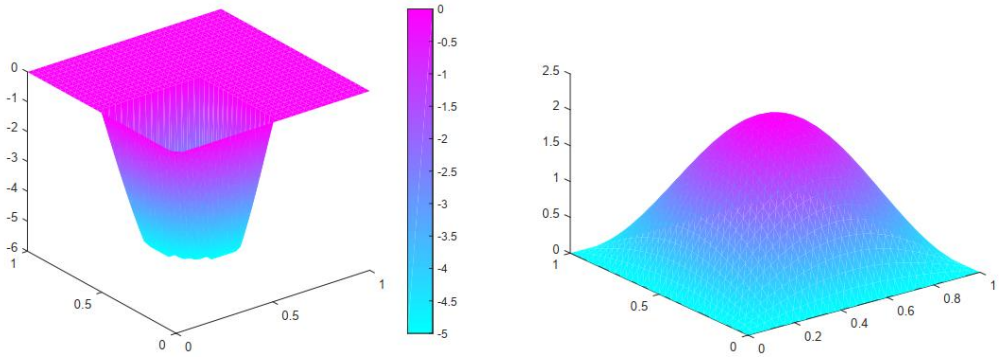
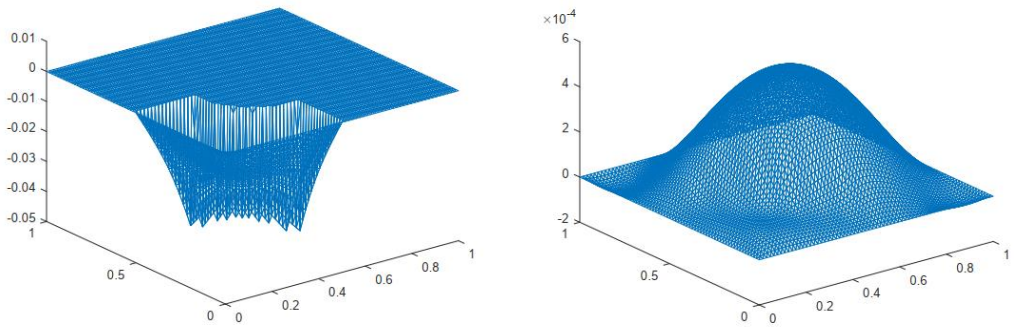


Figure 3.8: Errors $u^* - u$ (left) and $y^* - y$ (right) at $t = 0.75$ for Example 3.



Chapter 4

An Optimal Control based Two-Stage Numerical Approach for the Sparse Initial Source Identification of Diffusion-Advection Equations

4.1 Motivations

Inverse problems are ubiquitous in many contexts of science and engineering: among other fields, we may mention applications in medical imaging [59], acoustics [71], machine learning [149], and oceanography [167]. A prototypical and relevant example is the identification of moving pollution sources in either compressible or incompressible fluids that can be described by diffusion-advection systems [56, 87, 124, 123]. Such kind of inverse problem has important practical applications in various areas of science and engineering. For instance, an accurate estimation of pollution sources plays a crucial role in the environmental safeguard of densely populated cities, see e.g., [56, 124]. Following [34, 35, 123, 134], such a pollution sources identification problem can be mathematically modeled by

an initial source identification problem of diffusion-advection systems. Mainly inspired by applications that aim at identifying point-wise pollution sources [56, 124, 123], a situation of particular concern is that the initial source is known to be sparse, i.e., to have a support of Lebesgue measure zero. This motivates us to consider the following sparse initial source identification problem of diffusion-advection systems.

4.1.1 Problem statement

Let $\Omega \subset \mathcal{R}^N$ with $N \geq 1$ be a bounded domain and let $\partial\Omega$ be its boundary. We consider the following linear diffusion-advection equation

$$\begin{cases} \partial_t u - d\Delta u + v \cdot \nabla u = 0, & (x,t) \in \Omega \times (0,T) \\ u = 0, & (x,t) \in \partial\Omega \times (0,T) \\ u(x,0) = u_0(x), & x \in \Omega, \end{cases} \tag{4.1}$$

where $0 < T < +\infty$ is a given final time, $d > 0$ is the diffusivity coefficient and the vector v is the velocity field of the advection. In what follows, we assume d and v to be constant for simplicity, although the techniques we are going to present can be adapted also to variable diffusivity and velocity fields.Then, the corresponding sparse initial source identification problem reads as:

Problem 4.1. *Given a target or observed function u_d, we aim at identifying a sparse initial condition $\widehat{u}_0^*$ such that the corresponding final state $\widehat{u}^*(\cdot,T)$ of (4.1) is as close as possible to u_d in the sense that*

$$\|\widehat{u}^*(\cdot,T) - u_d\|_{L^2(\Omega)} \leq \varepsilon, \quad a.e \text{ in } \Omega, \ \varepsilon \geq 0. \tag{4.2}$$

Moreover, since we are interested in the situation in which the initial source is known to be sparse, it is sufficient to assume a priori that the initial condition $\widehat{u}_0^*$ to be determined is a linear combination of Dirac measures:

$$\widehat{u}_0^* = \sum_{i=1}^{\ell} \widehat{\alpha}_i^* \delta(\widehat{x}_i^*), \quad \widehat{x}_i^* \in \Omega, \tag{4.3}$$

where $\{\widehat{\alpha}_i^*\}_{i=1}^{\ell} \in \mathbb{R}^\ell$ and $\widehat{x}_i^* \in \Omega, 1 \leq i \leq \ell$, are the intensities and locations to be identified, respectively. The number ℓ of locations is fixed and the Dirac measure δ is defined by $\delta(x) = 1$ if $x = \widehat{x}_i^*$, and $\delta(x) = 0$ otherwise.

Above, the a priori assumption (4.3) implies that the support of $\widehat{u}_0^*$ is $\{\widehat{x}_i^*\}_{i=1}^{\ell} \subset \Omega$, whose Lebesgue measure is zero, and thus guarantees the initial source to be identified is sparse. In general, Problem 4.1 is known to be exponentially ill-posed, see, e.g., [107]. In particular, the strong smoothing property of the equation (4.1) makes it difficult to design some numerical algorithms to identify such a sparse initial source in the form of (4.3).

4.1.2 State-of-the-art

In the literature, some work has already been done for sparse initial source identification problems, based on the natural idea of taking advantage of the sparse nature of the initial condition. For instance, in [123], the sparse initial source identification of the heat equation is solved by minimizing the ℓ^1-norm of the initial condition, under the constraint that the corresponding final state of the heat equation and the observations at the final time are close, through the classical Bregman iteration method [24].

A widely used strategy to address sparse initial source identification problems is to formulate them as optimal control problems with partial differential equations (PDEs) constraints, in which the initial condition is assumed to play the role of a control term. This is the seminal idea at the basis of some research articles, see e.g., [34, 35, 120, 134].

In [34], sparse optimal control techniques are used to identify sparse initial sources for diffusion-convection equations. Existence and uniqueness of optimal controls are proved, and necessary and sufficient optimality conditions are obtained. Based on these conditions, the sparsity structure of the solutions is derived, which relates to the initial sources to be identified. In [35], the adjoint methodology for sparse initial source identification problems governed by parabolic equations is introduced. It is proved that the sparse initial condition can be recovered by minimizing its measure-norm under the constraint that the corresponding solution and the given target are close at the final time. In [120], the identification of an unknown sparse initial source for a homogeneous parabolic equation is addressed by considering an optimal control problem, where

the control variable is considered in the space of regular Borel measures and the corresponding norm is used as a regularization term in the objective functional. Under a certain structural assumption, the authors show that the initial source is a finite linear combination of Dirac measures as that in (4.3).

It is remarkable that, in the above references, the sparse initial source identification problems are formulated as optimal control problems in measure spaces. The presence of measures promotes the sparsity of the initial source but entails appropriate discretization for measure-valued quantities and may invalidate the application of some well-known numerical methods. For instance, the first-order optimality condition cannot be reformulated in a non-smooth point-wise form and thus the well-known semi-smooth Newton type methods cannot be applied directly. Hence, some new numerical algorithms should be deliberately designed from scratch. In this regard, we note that a Primal-Dual Active Point (PDAP) method is proposed in [120] to solve the optimal control problem with measures resulted from the sparse initial source identification of homogeneous parabolic equations. At each iteration of the PDAP, one entails the solutions of two parabolic equations to update the adjoint variable, an optimization subproblem to find a new support point, and a non-smooth optimization subproblem to compute the new iterate. This non-smooth optimization problem has no closed-form solution and can only be solved iteratively by some optimization algorithm, such as the Semi-Smooth Newton (SSN) method suggested therein. Hence, nested iterations are resulted, which may cause some new challenges in the overall rigorous convergence and significant computational loads in the implementation.

In [134], the sparse initial source identification for diffusion-advection equations is considered. As that in (4.3), the initial source is assumed to be a linear combination of Dirac deltas and the optimal locations and intensities need to be identified. To solve this problem, an optimal control based two-stage numerical approach is proposed. First, the sparse initial source identification problem is formulated as an optimal control problem with L^1-regularized functional, where the initial condition is treated as the control variable and is assumed to be in $L^1(\Omega)$ to promote the sparsity. The employment of measures is thus avoided,

and various well-developed optimization algorithms can be applied directly to solve the resulting optimal control problem. In particular, a Gradient Descent (GD) method is suggested therein with the gradient computed by the adjoint methodology. However, due to the smoothing property of diffusion systems, the resulting optimal control (i.e., the identified initial source from the optimal control problem) is not sparse as desired. Nevertheless, it has been empirically observed that the local maxima/minima of the optimal control fall into the exact locations where the actual initial sources are placed. Consequently, the optimal locations are identified by determining where the maxima/minima of the optimal control are. Then, a post-processing is proposed, where a least squares problem is solved to identify the corresponding optimal intensities of the initial source. Several test cases both in one and two-dimensional spaces validate that this two-stage approach identifies the sparse initial sources very successfully even in heterogeneous media. Despite this fact, we shall remark that the focus in [134] is on the development and discussion of the numerical algorithm, but from a mathematical viewpoint, the optimal control model considered in [134] is not well-posed. In particular, since the control variable is considered in the non-reflexive space $L^1(\Omega)$, the existence of a solution to the optimal control problem cannot be guaranteed.

For completeness, we mention that other types of optimal control problems with sparsity properties have also been widely discussed in the existing literature. In [32, 36] for elliptic problems and in [33, 116] for parabolic problems, distributed controls with strong sparsity properties are obtained by considering optimal control problems in the space of measures. Distributed elliptic and parabolic optimal control problems with L^1-regularized functional are discussed in [161] and [164], respectively. The use of L^1-regularization has been shown to be very efficient to obtain optimal controls with support in small regions of the domain, the domain being adjustable in terms of the tuning of suitable parameters entering in the cost functional.

4.1.3 Our numerical approach

To address Problem 4.1, we propose a new optimal control based two-stage numerical approach, which consists of a sparsity detection stage and a structure enhancement stage. Our approach is mainly inspired by [134], and it keeps all advantageous features of the framework in [134] while avoids the aforementioned issues encountered therein. First, in the sparsity detection stage, we treat the initial condition u_0 as a control variable and formulate Problem 4.1 as an optimal control problem with $L^2 + L^1$-regularization. As to be shown in Section 4.2, the presence of the L^1-term can detect the sparsity of the initial source. However, as observed in [134], the identified initial source from the optimal control problem is not sparse due to the absence of the assumption (4.3) in the formulation of the optimal control problem and the smoothing property of diffusion systems. Hence, a structure enhancement stage should be complemented to ensure that (4.3) holds while identify the locations $\{\widehat{x}_i^*\}_{i=1}^{\ell}$ and the intensities $\{\widehat{\alpha}_i^*\}_{i=1}^{\ell}$.

To be concrete, we formulate Problem 4.1 in terms of the following optimal control model:

$$\min_{u_0 \in L^2(\Omega)} J(u_0) := \frac{1}{2}\int_\Omega |u(\cdot, T) - u_d|^2 \, dx + \frac{\alpha}{2}\int_\Omega |u_0|^2 \, dx + \beta \int_\Omega |u_0| \, dx, \qquad (4.4)$$

subject to the diffusion-advection equation (4.1). In (4.4), the constants $\alpha > 0$ and $\beta > 0$ are regularization parameters. Similar as the model in [134], the first term of $J(u_0)$ seeks for an initial condition u_0 such that the corresponding final state of equation (4.1) is as close as possible to u_d; and the last term detects the sparsity of the initial source. Meanwhile, inspired by [164], we introduce the L^2-regularization $\frac{\alpha}{2}\int_\Omega |u_0|^2 \, dx$ to guarantee the well-posedness of (4.4) while preventing possible ill-conditioning to allow for a more efficient numerical resolution.

Clearly, our proposed model (4.4) operates in function spaces and avoids the employment of measures. As a consequence, it can be easily addressed numerically and various well-developed optimization algorithms can be applied directly. On the other hand, due to the introduction of the L^2-regularization term, our proposed model (4.4) is well-posed and allows identifying the sparse initial sources much more efficiently than the one in [134] as to be validated by some numerical

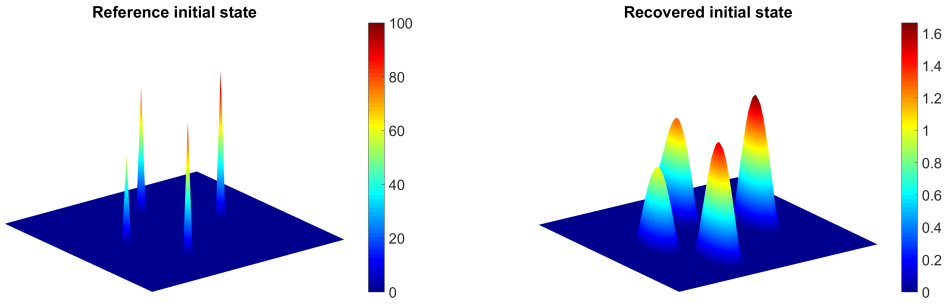
tests.

It is worth noticing that the control variable u_0 in (4.4) is considered as a general function in $L^2(\Omega)$ and it is not assumed to be a linear combination of Dirac measures as in (4.3). Therefore, one may further assume that $u_0 = \sum_{i=1}^{\ell} \alpha_i \delta(x_i)$ with $\alpha_i \in \mathbb{R}$ and $x_i \in \Omega$ in the formulation of (4.4). As a result, the intensities $\{\alpha_i\}_{i=1}^{\ell}$ and the locations $\{x_i\}_{i=1}^{\ell}$ become the control variables. However, this leads to a non-convex optimization problem which is challenging to be addressed both in terms of theory and algorithms. Meanwhile, it causes practical difficulties related to the computation of the derivatives with respect to $\{x_i\}_{i=1}^{\ell}$. By contrast, our formulation (4.4) is convex and the computation of the derivatives to u_0 is relatively easier as to be shown in Section 4.2.2.

Moreover, when a certain gradient-based algorithm is employed to solve the optimal control problem (4.4), the solutions of the state equation (4.1) and the adjoint equation (4.8) (or its analogue) are always required to compute the gradient. These are both diffusive processes that smooth out the corresponding states. Because of that, as observed in [134], the recovered initial condition u_0 is outside the sparse ansatz (4.3). To validate this fact, we set $T = 0.1$, $d = 0.05$ and $v = (1, 2)^\top$ in the equation (4.1) and solve the optimal control problem (4.4) by the Primal-Dual Hybrid Gradient (PDHG) method described in Section 4.3. Additional details are presented in Section 5.5. The numerical results are visualized in Figure 4.1, where the left plot corresponds to the reference initial datum $\widehat{u}_0$ assigned a prior in the form of (4.3), while the right plot shows the recovered initial datum u_0^* by solving the optimal control problem (4.4). We can clearly see that $\widehat{u}_0$ and u_0^* do not coincide, the latter one being of a non-sparse nature and with intensities way below the ones of $\widehat{u}_0$.

For the above reasons, once a numerical solution of the optimal control problem (4.4) is computed, a structure enhancement stage exploiting the assumption (4.3) is necessary to identify the optimal locations $\{\widehat{x}_i^*\}_{i=1}^{\ell}$ and the intensities $\{\widehat{\alpha}_i^*\}_{i=1}^{\ell}$. To this end, inspired by [134], we propose to solve two simple and low-dimensional optimization problems. More precisely, to identify the optimal locations $\{\widehat{x}_i^*\}_{i=1}^{\ell}$, we consider an optimization problem in terms of the spatial variable $x \in \Omega$, which is derived from the structural property of the solution

Figure 4.1: Reference initial datum $\widehat{u}_0$ (left) and recovered initial datum u_0^* (right) by solving (4.4).



of the optimal control problem (4.4) stated in Theorem 4.3. Then, motivated by the facts that the initial source $\widehat{u}_0^*$ to be recovered is a linear combination of Dirac measures and the associated final state $\widehat{u}^*(\cdot, T)$ should be as close as possible to u_d, we solve a least squares problem to identify the optimal intensities $\{\widehat{\alpha}_i^*\}_{i=1}^{\ell}$. An optimal control based two-stage numerical approach is thus proposed for Problem 4.1. The validity of our proposed approach can be guaranteed mathematically. Meanwhile, it allows identifying the sparse initial sources very successfully, even for some heterogeneous materials or coupled models as validated by some numerical experiments in Section 5.5.

4.1.4 PDHG methods for the solution of (4.4)

We note that the identification of the optimal locations and intensities is based on the solution of the optimal control problem (4.4) and thus it is of significance to employ an efficient numerical algorithm for solving (4.4). To this end, recall that the optimal control problem (4.4) is defined in function spaces and various well-developed optimization algorithms can be applied directly. For instance, SSN-type methods [175] or the Alternating Direction Method of Multipliers (ADMM) [72] can be conceptually applied and they indeed have been successful in solving some other types of optimal control problems in the literature (we refer for instance to [84, 99, 103, 114, 163] for a few contributions in this direction). Nevertheless, we note that at each iteration of SSN and ADMM, a complicated large-scale and ill-conditioned saddle point system and an optimal control subproblem should be iteratively solved, respectively, which are numeri-

cally challenging and expensive for such a time-dependent model.

To avoid the above issues, we advocate the PDHG method [37], which has been widely used in various areas, such as image processing [37, 193], inverse problems [169], and statistical learning [85]. But its application for solving optimal control problems has not been sufficiently explored. As to be shown in Section 4.3, the PDHG method decouples the original complicated optimal control problem (4.4) into two much simpler subproblems. At each iteration, the main computational load consists of solving only two PDEs which can be efficiently addressed by various well-developed PDE solvers. Hence, the PDHG method is very easy and cheap to be implemented. To further speed up the convergence of PDHG method, we introduce a PDHG-based prediction-correction algorithmic framework following [93, 95]. Moreover, we compare the numerical efficiency of the PDHG with the GD described in [134], and we show that the PDHG method yields significant improvements in the performance of the source identification procedure.

4.2 Analysis of the optimal control problem (4.4)

In this section, we analyze the properties of the optimal control problem (4.4). First, the existence and uniqueness of an optimal control u_0^* are proved. Then, we derive the optimality conditions and deduce some structural properties of u_0^*.

4.2.1 Existence and uniqueness of an optimal control

Let us start by discussing the existence and uniqueness of an optimal control u_0^* to the problem (4.4). This fact comes from a very standard argument.

As a matter of fact, the existence of u_0^* minimizing the functional $J(u_0)$ can be proved by taking a minimizing sequence and using the compactness of the map $u_0 \in L^2(\Omega) \mapsto u(\cdot, T) \in L^2(\Omega)$, which can be easily obtained by adapting the proof of [35, Lemma 2.3] and is a consequence of the smoothing properties of the heat semi-group. We leave the details to the reader.

Concerning the uniqueness, instead, this is a direct consequence of the strict convexity of the functional. Indeed, suppose that the optimal control problem (4.4) has two solutions $u^*_{0,1}$ and $u^*_{0,2}$. Then, if we denote

$$\overline{u}_0 := \frac{1}{2}\left(u^*_{0,1} + u^*_{0,2}\right)$$

we have

$$J(\overline{u}_0) = J\left(\frac{u^*_{0,1}}{2} + \frac{u^*_{0,2}}{2}\right) < \frac{1}{2}J\left(u^*_{0,1}\right) + \frac{1}{2}J\left(u^*_{0,2}\right) = \inf(J),$$

which contradicts the fact that $u^*_{0,1}$ and $u^*_{0,2}$ are minimizers. We have then proved the following theorem.

Theorem 4.1. *There exists a unique solution $u^*_0 \in L^2(\Omega)$ of the optimal control problem* (4.4)*.*

4.2.2 First-order optimality condition

To derive the first-order optimality condition, we introduce the Lagrangian formulation

$$L(u,\psi) = \frac{1}{2}\int_\Omega |u(\cdot,T) - u_d|^2\,dx + \frac{\alpha}{2}\int_\Omega |u_0|^2\,dx + \beta\int_\Omega |u_0|\,dx + \int_0^T\int_\Omega \psi(-\partial_t u + d\Delta u - v\cdot\nabla u)\,dxdt.$$

Let

$$\varphi(u_0) = \beta\int_\Omega |u_0| dx \tag{4.5}$$

and denote by

$$\partial\varphi(u_0) := \left\{\lambda_{u_0} \in L^2(\Omega) : \varphi(v) - \varphi(u_0) - (\lambda_{u_0}, v - u_0) \geq 0, \forall v \in L^2(\Omega)\right\}$$

the subdifferential of φ at $u_0 \in L^2(\Omega)$. Here and in what follows, we denote by $(\cdot,\cdot)$ the canonical inner product in $L^2(\Omega)$. Then, we compute the directional derivative $\delta L(u,\psi)$ as

$$\delta L(u,\psi) = \int_\Omega (u(\cdot,T) - u_d)\delta u(\cdot,T)\,dx + \alpha\int_\Omega u_0\delta u_0\,dx + \int_\Omega \lambda_{u_0}\delta u_0\,dx$$

$$+\int_0^T\int_\Omega \psi(-\partial_t\delta u + d\Delta\delta u - v\cdot\nabla\delta u)\,dxdt, \tag{4.6}$$

with

$$\delta u = 0 \quad \text{on } \partial\Omega\times(0,T). \tag{4.7}$$

We now integrate by parts the last term in the above expression, obtaining

$$\begin{aligned}
&\int_0^T\int_\Omega \psi(-\partial_t\delta u + d\Delta\delta u - v\nabla\cdot\delta u)\,dxdt\\
&= -\int_\Omega \psi(\cdot,T)\delta u(\cdot,T)\,dx + \int_\Omega \psi(\cdot,0)\delta u(\cdot,0)\,dx\\
&\quad + \int_0^T\int_\Omega (\partial_t\psi + d\Delta\psi + v\cdot\nabla\psi)\delta u\,dxdt,
\end{aligned}$$

where we took into account (4.7) and the fact that we are assuming v to be constant. Hence, we obtain from (4.6) that

$$\begin{aligned}
\delta L(u,\psi) = &\int_\Omega (u(\cdot,T) - u_d - \psi(\cdot,T))\delta u(\cdot,T)\,dx + \int_\Omega (\alpha u_0 + \lambda_{u_0} + \psi(\cdot,0))\delta u_0\,dx\\
&+\int_0^T\int_\Omega (\partial_t\psi + d\Delta\psi + v\cdot\nabla\psi)\delta u\,dxdt
\end{aligned}$$

or, equivalently,

$$\delta L(u,\psi) = \int_\Omega \Big(\alpha u_0 + \lambda_{u_0} + \psi(\cdot,0)\Big)\delta u_0\,dx$$

with the constraint that ψ is a solution of the adjoint (backward) equation

$$\begin{cases}
\partial_t\psi + d\Delta\psi + v\cdot\nabla\psi = 0, & (x,t)\in\Omega\times(0,T)\\
\psi = 0, & (x,t)\in\partial\Omega\times(0,T)\\
\psi(\cdot,T) = u(\cdot,T) - u_d := \psi_T, & x\in\Omega.
\end{cases} \tag{4.8}$$

This, in particular, implies that the gradient $\nabla J(u_0)$ is given by the expression

$$\nabla J(u_0) = \psi(\cdot,0) + \alpha u_0 + \lambda_{u_0} \tag{4.9}$$

The above discussions lead to the following result.

Theorem 4.2. *Suppose that $u_0^*\in L^2(\Omega)$ is the unique solution of the optimal control problem (4.4). Then, the following first-order optimality condition holds:*

$$\psi^*(\cdot,0) + \alpha u_0^* + \lambda_{u_0}^* = 0,$$

where $\lambda_{u_0}^*\in\partial\varphi(u_0^*)$, and ψ^* is the successive solution of the state equation (4.1) and the adjoint equation (4.8) provided the initial datum u_0^*.

4.2.3 Structural properties of u_0^*

Recall that $\lambda_{u_0}^* \in \partial\varphi(u_0^*) = \beta\partial\int_\Omega |u_0^*|dx$, and it follows from the results of [110] that

$$\lambda_{u_0}^* \in \beta \text{sign}(u_0^*),$$

where the set-valued function $\text{sign}(\cdot)$ is given by

$$\text{sign}(v) = \begin{cases} \dfrac{v}{|v|}, & \text{if } v \neq 0 \\ \{\eta : |\eta| \leq 1\}, & \text{otherwise.} \end{cases}$$

We thus obtain that

$$\begin{aligned} &\lambda_{u_0}^* = \beta, \text{ a.e. on } \{x \in \Omega : u_0^* > 0\} \\ &\lambda_{u_0}^* = -\beta, \text{ a.e. on } \{x \in \Omega : u_0^* < 0\} \\ &|\lambda_{u_0}^*| \leq \beta, \text{ a.e. on } \{x \in \Omega : u_0^* = 0\}, \end{aligned} \tag{4.10}$$

which implies the following structural information for the solution u_0^*.

Theorem 4.3. *Let $u_0^* \in L^2(\Omega)$ be the unique solution of the problem (4.4) and $\lambda_{u_0}^* \in \beta\partial\int_\Omega |u_0^*|dx$, then the support of u_0^* verifies*

$$supp(u_0^*) \subset \{x \in \Omega : \lambda_{u_0}^*(x) = \pm\beta\}.$$

The above result implies the following structural property: for any $x_0 \in \Omega$, the solution $u_0^*(x_0)$ will be nonzero only if $\lambda_{u_0}^*$ reach its maximum β or minimum $-\beta$ at x_0. Moreover, we can also show that for a fixed α, the larger the parameter β, the smaller the support of the solution u_0^*. Indeed, when β is sufficient large, using some similar arguments as those in [164], we can prove that $u_0^* = 0$ on the whole domain Ω.

Theorem 4.4. *Let $\mathcal{L} : L^2(\Omega) \to L^2(\Omega)$ be the solution operator associated with the linear diffusion-advection equation (4.1), i.e. $\mathcal{L}u_0 = u(\cdot, T)$, and let $\mathcal{L}^*$ denote its adjoint. Then, if $\beta \geq \beta_0 := \|\mathcal{L}^* u_d\|_{L^\infty(\Omega)}$, the unique solution of the problem (4.4) is $u_0^* = 0$.*

Proof. We first note that, with $\mathcal{L}u_0 = u(\cdot, T)$, the objective functional $J(u_0)$ in (4.4) can be rewritten as

$$J(u_0) = \frac{1}{2}\int_\Omega |\mathcal{L}u_0 - u_d|^2\,dx + \frac{\alpha}{2}\int_\Omega |u_0|^2\,dx + \beta\int_\Omega |u_0|\,dx.$$

Then, it is easy to obtain that

$$\begin{aligned} J(u_0) - J(0) &= \frac{1}{2}\int_\Omega |\mathcal{L}u_0|^2\,dx - \int_\Omega \mathcal{L}u_0 u_d\,dx + \frac{\alpha}{2}\int_\Omega |u_0|^2\,dx + \beta\int_\Omega |u_0|\,dx \\ &= \frac{1}{2}\|\mathcal{L}u_0\|^2_{L^2(\Omega)} - (u_0, \mathcal{L}^* u_d) + \frac{\alpha}{2}\|u_0\|^2_{L^2(\Omega)} + \beta\|u_0\|_{L^1(\Omega)} \\ &\geq \frac{1}{2}\|\mathcal{L}u_0\|^2_{L^2(\Omega)} - \|u_0\|_{L^1(\Omega)}\|\mathcal{L}^* u_d\|_{L^\infty(\Omega)} + \frac{\alpha}{2}\|u_0\|^2_{L^2(\Omega)} + \beta\|u_0\|_{L^1(\Omega)} \\ &= \frac{1}{2}\|\mathcal{L}u_0\|^2_{L^2(\Omega)} + (\beta - \|\mathcal{L}^* u_d\|_{L^\infty(\Omega)})\|u_0\|_{L^1(\Omega)} + \frac{\alpha}{2}\|u_0\|^2_{L^2(\Omega)}. \end{aligned}$$

If $\beta \geq \beta_0$, we have that $J(0) \leq J(u_0)$ for any $u_0 \in L^2(\Omega)$, which implies that the unique solution of the problem (4.4) is $u_0^* = 0$. □

4.3 PDHG algorithms for the optimal control problem (4.4)

In this section, we first elaborate the application of the PDHG method [37] to the optimal control problem (4.4) and delineate the implementation details. Then, following the ideas in [93, 95], we introduce in Subsection 4.3.3 a generalized PDHG algorithmic framework which allows the output of the PDHG subroutine to be further updated by relaxation steps with constant step sizes. With different choices of parameters, a class of generalized PDHG schemes can be obtained for solving the problem (4.4). These generalized algorithms are usually more efficient than the original PDHG algorithm in practice, as we will show in Section 4.7.

4.3.1 Iterative scheme of the PDHG method

Let us now briefly describe the PDHG methodology. To this end, let us define

$$f(\mathcal{L}u_0) = \int_\Omega |\mathcal{L}u_0 - u_d|^2\,dx \quad \text{and} \quad g(u_0) = \frac{\alpha}{2}\int_\Omega |u_0|^2\,dx + \beta\int_\Omega |u_0|\,dx.$$

Then, the optimal control problem (4.4) can be reformulated as

$$\min_{u_0\in L^2(\Omega)} f(\mathcal{L}u_0) + g(u_0). \tag{4.11}$$

Introducing now an auxiliary variable $p \in L^2(\Omega)$, by applying the standard Fenchel-Rockafellar duality (see e.g., [55, Chapter VII]) we can show that (4.11) is equivalent to the following saddle point problem:

$$\min_{u_0\in L^2(\Omega)} \max_{p\in L^2(\Omega)} g(u_0) + (p, \mathcal{L}u_0) - f^*(p) \tag{4.12}$$

where $f^*(p) := \sup_{q\in L^2(\Omega)} \Big((p,q) - f(q)\Big)$ is the convex conjugate of $f(q)$ and can be explicitly computed as

$$f^*(p) = \frac{1}{2}\int_\Omega |p|^2\, dx + (p, u_d).$$

Then, applying the PDHG method proposed in [37] to the problem (4.12), we immediately obtain the following iterative scheme:

$$\begin{cases} u_0^{k+1} = \arg\min\limits_{u_0\in L^2(\Omega)} \Big(g(u_0) + (p^k, \mathcal{L}u_0) + \dfrac{1}{2r}\|u_0 - u_0^k\|^2_{L^2(\Omega)}\Big), & (4.13)\\ \bar{u}_0^k = 2u_0^{k+1} - u_0^k, & (4.14)\\ p^{k+1} = \arg\max\limits_{p\in L^2(\Omega)} \Big((p, \mathcal{L}\bar{u}_0^k) - f^*(p) - \dfrac{1}{2s}\|p - p^k\|^2_{L^2(\Omega)}\Big). & (4.15) \end{cases}$$

4.3.2 Implementation of the PDHG method (4.13)-(4.15)

In this subsection, we discuss the implementation details of the PDHG algorithm. First of all, we observe that the u-subproblem (4.13) can be equivalently reformulated as

$$u_0^{k+1} = \arg\min_{u_0\in L^2(\Omega)} \left(\frac{\alpha}{2}\int_\Omega |u_0|^2\, dx + \beta\int_\Omega |u_0|\, dx + \frac{1}{2r}\|u_0 - u_0^k + r\mathcal{L}^* p^k\|^2_{L^2(\Omega)}\right), \tag{4.16}$$

where $\mathcal{L}^*\bar{p}^k := \zeta^k(\cdot, 0)$ is the solution at time $t = 0$ of the following backward equation:

$$\begin{cases} \partial_t \zeta^k + d\Delta\zeta^k + v\cdot\nabla\zeta^k = 0, & (x,t)\in\Omega\times(0,T)\\ \zeta^k = 0, & (x,t)\in\partial\Omega\times(0,T)\\ \zeta^k(\cdot, T) = p^k, & x\in\Omega. \end{cases} \tag{4.17}$$

In addition to that, it can be readily checked (see e.g. [110]) that problem (4.16) has the following closed-form solution

$$u_0^{k+1} = \mathcal{S}_{\frac{\beta r}{\alpha r+1}}\left(\frac{u_0^k - r\zeta^k(\cdot,0)}{\alpha r+1}\right)$$

where, for any constant $\gamma > 0$, we denoted by $\mathcal{S}_\gamma$ the Shrinkage operator defined as

$$\mathcal{S}_\gamma(a) = \begin{cases} a-\gamma, & a > \gamma \\ 0, & |a| \le \gamma \\ a+\gamma, & a < -\gamma. \end{cases}$$

Concerning instead the solution of the p-subproblem (4.15), the latter can be computed explicitly taking into account that p^{k+1} has satisfy

$$\nabla_p\Big((p, \mathcal{L}\bar{u}_0^k) - f^*(p) - \frac{1}{2s}\|p-p^k\|^2_{L^2(\Omega)}\Big)\Big|_{p=p^{k+1}} = 0.$$

In particular, we have

$$p^{k+1} = \frac{1}{s+1}p^k + \frac{s}{s+1}\Big(\mathcal{L}\bar{u}_0^k - u_d\Big).$$

Clearly, at each iteration, the main computation of the PDHG method only requires the solutions of one forward equation (4.1) and one backward equation (4.17), and both of them can be efficiently solved by various well-developed PDE solvers. Hence, the PDHG method is very cheap and easy to be implemented. On the other hand, as discussed in the introduction, the numerical solution obtained from (4.4) by the PDHG algorithm will not be as sparse as desired in (4.3) due to the smoothing property of the equations (4.1) and (4.17). Hence, a structure enhancement stage to be introduced in Section 4.6 should be considered in order to identify a sparse initial condition in the form of (4.3) such that (4.2) holds.

4.3.3 A generalized PDHG-based prediction-correction algorithmic framework

Inspired by [93, 95], we consider the following generalized PDHG-based prediction-correction algorithmic framework presented in Algorithm 4.1, which can further

improve the numerical efficiency of the PDHG method (4.13)-(4.15) in practice, as to be shown in Section 4.7.

Algorithm 4.1 A generalized PDHG-based prediction-correction algorithmic framework for (4.4)

input: initial values $u_0^0 \in L^2(\Omega)$ and $p^0 \in L^2(\Omega)$. Choose constants $\rho > 0, \sigma > 0$ and $\theta \in (0, 1]$, and step sizes $r > 0$, $s > 0$ such that the following condition holds:

$$rs < \frac{1}{\|\mathcal{L}\mathcal{L}^*\|}. \tag{4.18}$$

while not converged **do**

PREDICTION STEP: compute $\widetilde{w}^k := (\widetilde{u}_0^k, \widetilde{p}^k)^\top$ by

$$\widetilde{u}_0^k = \arg\min_{u \in L^2(\Omega)} \Big(g(u_0) + (p^k, \mathcal{L}u_0) + \frac{1}{2r}\|u_0 - u^k\|^2_{L^2(\Omega)} \Big) \tag{4.19a}$$

$$\bar{u}_0^k = \widetilde{u}_0^k + \theta(\widetilde{u}_0^k - u_0^k) \tag{4.19b}$$

$$\widetilde{p}^k = \arg\max_{p \in L^2(\Omega)} \Big((p, \mathcal{L}\bar{u}_0^k) - f^*(p) - \frac{1}{2s}\|p - p^k\|^2_{L^2(\Omega)} \Big) \tag{4.19c}$$

CORRECTION STEP: update the new iterate $w^{k+1} := (u_0^{k+1}, p^{k+1})^\top$ via

$$u_0^{k+1} = u_0^k - \rho(u_0^k - \widetilde{u}_0^k) \tag{4.20a}$$

$$p^{k+1} = p^k - \sigma(p^k - \widetilde{p}^k) \tag{4.20b}$$

end while

With different choices of parameters θ and ρ, σ, a class of new generalized PDHG schemes can be obtained. We refer to [93] for the details. Here, in particular, we take $\theta = 1$ in (4.19b) and thus obtain a PDHG-based prediction-correction (PDHG-PC) method. We note that the correction steps (4.20a) and (4.20b) are easy to compute and the resulting subproblems (4.19a)-(4.19b) are similar to those in (4.13)-(4.15). Hence, the implementation of the PDHG-PC method for solving the optimal control problem (4.4) shares the similar routine as the PDHG method (4.13)-(4.15). Clearly, if we further let $\sigma = \rho = 1$, the PDHG-PC method reduces to the classical PDHG method (4.13)-(4.15).

4.4 Convergence analysis of Algorithm 4.1

In this section, we prove the strong global convergence and derive the worst-case $O(1/K)$ convergence rate measured by the iteration complexity in both the ergodic and non-ergodic senses for Algorithm 4.1 in the context of optimal control problems. All the results can be extended to the classical PDHG method (4.13)-(4.15) directly since it can be covered by Algorithm 4.1 with $\theta = 1$ and $\rho = \sigma = 1$.

4.4.1 Preliminaries

Denote $(u_0^*, p^*)^\top \in L^2(\Omega) \times L^2(\Omega)$ the saddle point of (4.12), which in particular means that u_0^* is the unique solution of (4.4). Then, the following variational inequalities (VIs) hold (see (4.5) for the definition of φ):

$$\varphi(u_0) - \varphi(u_0^*) + \Big(u_0 - u_0^*, \alpha u_0^* + \mathcal{L}^* p^*\Big) \geq 0, \qquad \forall u_0 \in L^2(\Omega), \tag{4.21a}$$

$$\Big(p - p^*, p^* + u_d - \mathcal{L} u_0^*\Big) \geq 0, \qquad \forall p \in L^2(\Omega). \tag{4.21b}$$

We observe that the VIs (4.21a) and (4.21b) can be written in a compact form:

$$\varphi(u_0) - \varphi(u_0^*) + \Big(w - w^*, F(w^*)\Big) \geq 0, \quad \forall w \in W, \tag{4.22}$$

where

$$W = L^2(\Omega) \times L^2(\Omega), \quad w = \begin{pmatrix} u_0 \\ p \end{pmatrix}, \quad F(w) = \begin{pmatrix} \alpha u_0 + \mathcal{L}^* p \\ p - \mathcal{L} u_0 + u_d \end{pmatrix}. \tag{4.23}$$

Moreover, a direct calculation shows that, for all $w_1, w_2 \in W$,

$$\begin{aligned} &\Big(w_1 - w_2, F(w_1) - F(w_2)\Big) && (4.24)\\ &= \Big(u_{0,1} - u_{0,2}, \alpha(u_{0,1} - u_{0,2})\Big) + \Big(u_{0,1} - u_{0,2}, \mathcal{L}(p_1 - p_2)\Big) \\ &\quad + \Big(p_1 - p_2, p_1 - u_2\Big) - \Big(\mathcal{L}(u_{0,1} - u_{0,2}), p_1 - p_2\Big) \\ &= \|p_1 - p_2\|^2_{L^2(\Omega)} + \alpha \|u_{0,1} - u_{0,2}\|^2_{L^2(\Omega)}, && (4.25) \end{aligned}$$

which implies that F is strongly monotone.

Then, we rewrite also the iterative scheme (4.13)-(4.15) in a VI form. For this purpose, we first note that the optimality conditions of (4.19a) and (4.19c) are

$$\begin{aligned}
&\varphi(u_0) - \varphi(\widetilde{u}_0^k) + \Big(u_0 - \widetilde{u}_0^k, \alpha\widetilde{u}_0^k + \mathcal{L}^* p^k + \frac{1}{r}(\widetilde{u}_0^k - u_0^k)\Big) \geq 0, \quad \forall u_0 \in L^2(\Omega),\\
&\Big(p - \widetilde{p}^k, \widetilde{p}^k + u_d - \mathcal{L}\widetilde{u}_0^k + \frac{1}{s}(\widetilde{p}^k - p^k)\Big) \geq 0, \qquad\qquad \forall p \in L^2(\Omega),
\end{aligned}$$

respectively. Taking (4.14) into account, we obtain the following VIs:

$$\varphi(u_0) - \varphi(\widetilde{u}_0^k) + \Big(u_0 - \widetilde{u}_0^k, \alpha\widetilde{u}_0^k + \mathcal{L}^*\widetilde{p}^k - \mathcal{L}^*(\widetilde{p}^k - p^k) + \frac{1}{r}(\widetilde{u}_0^k - u_0^k)\Big) \geq 0, \ \forall u_0 \in L^2(\Omega), \tag{4.27a}$$

$$\Big(p - \widetilde{p}^k, \widetilde{p}^k + u_d - \mathcal{L}\widetilde{u}_0^k - \theta\mathcal{L}(\widetilde{u}_0^k - u_0^k) + \frac{1}{s}(\widetilde{p}^k - p^k)\Big) \geq 0, \ \forall p \in L^2(\Omega). \tag{4.27b}$$

To simplify the notation, we define the following matrix-form operators

$$\mathcal{D} := \begin{pmatrix} \rho I & 0 \\ 0 & \sigma I \end{pmatrix}, \quad G := \begin{pmatrix} \frac{1}{r} I & -\mathcal{L}^* \\ -\theta\mathcal{L} & \frac{1}{s} I \end{pmatrix}, \quad \mathcal{K} := G\mathcal{D}^{-1}, \quad \mathcal{M} := G + G^* - \mathcal{D}^*\mathcal{K}\mathcal{D}. \tag{4.28}$$

With the notations in (4.23) and (4.28), the VIs (4.27a) and (4.27b), as well as the correction steps (4.20a) and (4.20b), can be respectively written in the following compact forms

$$\varphi(u_0) - \varphi(\widetilde{u}_0^k) + \Big(w - \widetilde{w}^k, F(\widetilde{w}^k) + G(\widetilde{w}^k - w^k)\Big) \geq 0, \quad \forall w \in W, \tag{4.29}$$

and

$$w^{k+1} = w^k - \mathcal{D}(w^k - \widetilde{w}^k). \tag{4.30}$$

We recall that in Algorithm 4.1, the combination parameter $\theta > 0$, and the step-sizes $r, s > 0$ are chosen such that

$$\theta \in (0, 1], \quad rs < \frac{1}{\|\mathcal{L}^*\mathcal{L}\|}. \tag{4.31}$$

To prove the convergence of Algorithm 4.1, we further impose the following condition: both $\mathcal{K}$ and $\mathcal{M}$ are self-adjoint and positive definite, namely,

$$\begin{aligned}
&\mathcal{K} = \mathcal{K}^* \ \text{ and } \ (\mathcal{K}w, w) \geq c_1\|w\|^2_{L^2(\Omega)},\\
&\mathcal{M} = \mathcal{M}^* \ \text{ and } \ (\mathcal{M}w, w) \geq c_2\|w\|^2_{L^2(\Omega)}, \quad \forall w \in W, w \neq 0,
\end{aligned} \tag{4.32}$$

where c_1 and c_2 are two positive constants.

This condition (4.32) yields further restrictions, in addition to (4.31), on the involved parameters in Algorithm 4.1. Using some similar arguments as those in [93], we have the following result.

Lemma 4.1. *Suppose that r and s satisfy* (4.31). *If we choose θ, ρ and σ such that either*

$$\theta = 1 \quad \textit{and} \quad \rho = \sigma \in (0, 2),$$

or

$$\theta \in (0, 1), \quad 0 < \rho \leq 1 + \theta - \sqrt{1-\theta} \quad \textit{and} \quad \rho = \sigma\theta,$$

then the matrix-form operators $\mathcal{K}$ and $\mathcal{M}$ defined in (4.28) *satisfy the convergence condition* (4.32).

4.4.2 Global convergence of Algorithm 4.1

In this subsection, we prove the convergence of Algorithm 4.1 under the conditions (4.31) and (4.32). Here and in what follows, we denote by $\|w\|_{\mathcal{A}} := (\mathcal{A}w, w), \forall w \in W$ the norm induced by a self-adjoint and positive definite matrix-operator $\mathcal{A}$. First, we show the strict contraction property of the sequence w^k generated by Algorithm 4.1.

Theorem 4.5. *Let $w^k = (u_0^k, p^k)^\top$ be the sequence generated by Algorithm 4.1 and $w^* = (u_0^*, p^*)^\top$ be the solution of problem* (4.12). *Suppose that the condition* (4.31) *holds and the matrix-form operators $\mathcal{K}$ and $\mathcal{M}$ satisfy the convergence condition* (4.32). *Then we have*

$$\|w^{k+1} - w^*\|_{\mathcal{K}}^2 \leq \|w^k - w^*\|_{\mathcal{K}}^2 - \|w^k - \widetilde{w}^k\|_{\mathcal{M}}^2 - 2\|\widetilde{p}^k - p^*\|_{L^2(\Omega)}^2 - 2\alpha\|\widetilde{u}_0^k - u_0^*\|_{L^2(\Omega)}^2. \tag{4.33}$$

Proof. First of all, it follows from (4.28) and (4.30) that the VI (4.29) can be written as

$$\varphi(u_0) - \varphi(\widetilde{u}_0^k) + \Big(w - \widetilde{w}^k, F(\widetilde{w}^k)\Big) \geq \Big(w - \widetilde{w}^k, \mathcal{K}(w^k - w^{k+1})\Big), \quad \forall w \in W. \tag{4.34}$$

Under the condition (4.32), we apply the identity

$$\big(a-b,\mathcal{K}(c-d)\big)=\frac{1}{2}\left(\|a-d\|_{\mathcal{K}}^2-\|a-c\|_{\mathcal{K}}^2\right)+\frac{1}{2}\left(\|c-b\|_{\mathcal{K}}^2-\|d-b\|_{\mathcal{K}}^2\right)$$

to the right-hand side of (4.34) with

$$a=w,\quad b=\widetilde{w}^k,\quad c=w^k,\quad \text{and}\quad d=w^{k+1}.$$

We then obtain

$$\Big(w-\widetilde{w}^k,\mathcal{K}(w^k-w^{k+1})\Big) \tag{4.35}$$

$$=\frac{1}{2}\left(\|w-w^{k+1}\|_{\mathcal{K}}^2-\|w-w^k\|_{\mathcal{K}}^2\right)+\frac{1}{2}\left(\|w^k-\widetilde{w}^k\|_{\mathcal{K}}^2-\|w^{k+1}-\widetilde{w}^k\|_{\mathcal{K}}^2\right). \tag{4.36}$$

Considering the last two terms in (4.35) and using (4.28) and (4.30), we have

$$\begin{aligned}
&\|w^k-\widetilde{w}^k\|_{\mathcal{K}}^2-\|w^{k+1}-\widetilde{w}^k\|_{\mathcal{K}}^2\\
=&\|w^k-\widetilde{w}^k\|_{\mathcal{K}}^2-\|(w^k-\widetilde{w}^k)-(w^k-w^{k+1})\|_{\mathcal{K}}^2\\
=&\|w^k-\widetilde{w}^k\|_{\mathcal{K}}^2-\|(w^k-\widetilde{w}^k)-\mathcal{D}(w^k-\widetilde{w}^k)\|_{\mathcal{K}}^2\\
=&2\Big(w^k-\widetilde{w}^k,\mathcal{K}\mathcal{D}(w^k-\widetilde{w}^k)\Big)-\Big(\mathcal{D}(w^k-\widetilde{w}^k),\mathcal{K}\mathcal{D}(w^k-\widetilde{w}^k)\Big)\\
=&2\Big(w^k-\widetilde{w}^k,G(w^k-\widetilde{w}^k)\Big)-\Big(w^k-\widetilde{w}^k,\mathcal{D}^*\mathcal{K}\mathcal{D}(w^k-\widetilde{w}^k)\Big)\\
=&\Big(w^k-\widetilde{w}^k,(G+G^*-\mathcal{D}^*\mathcal{K}\mathcal{D})(w^k-\widetilde{w}^k)\Big)\\
=&\|w^k-\widetilde{w}^k\|_{\mathcal{M}}^2.
\end{aligned} \tag{4.37}$$

Combining (4.34), (4.35) and (4.37), we obtain that

$$\begin{aligned}
&\varphi(u_0)-\varphi(\widetilde{u}_0^k)+\Big(w-\widetilde{w}^k,F(\widetilde{w}^k)\Big)\\
\geq&\frac{1}{2}\Big(\|w-w^{k+1}\|_{\mathcal{K}}^2-\|w-w^k\|_{\mathcal{K}}^2\Big)+\frac{1}{2}\|w^k-\widetilde{w}^k\|_{\mathcal{M}}^2,\quad \forall w\in W.
\end{aligned} \tag{4.38}$$

It follows from (4.38) that, for all $w\in W$,

$$\begin{aligned}
\varphi(\widetilde{u}_0^k)-\varphi(u_0)+\Big(\widetilde{w}^k-w,F(w)\Big)+\Big(\widetilde{w}^k-w,F(\widetilde{w}^k)-F(w)\Big)&\\
\leq\frac{1}{2}\Big(\|w^k-w\|_{\mathcal{K}}^2-\|w^{k+1}-w\|_{\mathcal{K}}^2\Big)-\frac{1}{2}\|w^k-\widetilde{w}^k\|_{\mathcal{M}}^2.&
\end{aligned} \tag{4.39}$$

Moreover, we recall that (see (4.24))

$$\Big(\widetilde{w}^k-w,F(\widetilde{w}^k)-F(w)\Big)=\|\widetilde{p}^k-p\|_{L^2(\Omega)}^2+\alpha\|\widetilde{u}_0^k-u_0\|_{L^2(\Omega)}^2.$$

Hence, setting $w=w^*$ in (4.39), and using (4.22), we finally obtain

$$\|w^{k+1}-w^*\|_{\mathcal{K}}^2\leq\|w^k-w^*\|_{\mathcal{K}}^2-\|w^k-\widetilde{w}^k\|_{\mathcal{M}}^2-2\|\widetilde{p}^k-p^*\|_{L^2(\Omega)}^2-2\alpha\|\widetilde{u}_0^k-u_0^*\|_{L^2(\Omega)}^2.$$

□

Theorem 4.5 shows that the sequence $w^k = (u_0^k, p^k)^\top$ generated by Algorithm 4.1 is strictly contractive. This, in turns, implies the convergence of w^k to the solution point w^* of the problem (4.12), as we shall see in the following theorem.

Theorem 4.6. *Let $w^k = (u_0^k, p^k)^\top$ be the sequence generated by Algorithm 4.1 and $w^* = (u_0^*, p^*)^\top$ be the solution of problem (4.12). Suppose that the condition (4.31) holds and the matrix-form operators $\mathcal{K}$ and $\mathcal{M}$ satisfy the convergence condition (4.32). Then u_0^k converges to u_0^* strongly in $L^2(\Omega)$ and p^k converges to p^* strongly in $L^2(\Omega)$.*

Proof. First of all, it follows from (4.33) that

$$\sum_{k=0}^{\infty} \Big(\|\widetilde{w}^k - w^k\|_{\mathcal{M}}^2 + 2\|\widetilde{p}^k - p^*\|_{L^2(\Omega)}^2 + 2\alpha \|\widetilde{u}_0^k - u_0^*\|_{L^2(\Omega)}^2 \Big) \leq \|w^0 - w^*\|_{\mathcal{K}}^2.$$

This means that the series

$$\sum_{k=0}^{\infty} \Big(\|\widetilde{w}^k - w^k\|_{\mathcal{M}}^2 + 2\|\widetilde{p}^k - p^*\|_{L^2(\Omega)}^2 + 2\alpha \|\widetilde{u}_0^k - u_0^*\|_{L^2(\Omega)}^2 \Big)$$

is convergent which, in particular, implies

$$\|\widetilde{w}^k - w^k\|_{\mathcal{M}}^2 \to 0,\ \|\widetilde{u}_0^k - u_0^*\|_{L^2(\Omega)}^2 \to 0,\ \text{and}\ \|\widetilde{p}^k - p^*\|_{L^2(\Omega)}^2 \to 0, \quad \text{as } k \to \infty. \tag{4.40}$$

Thus

$$\widetilde{p}^k \to p^*,\ \widetilde{u}_0^k \to u_0^*, \quad \text{strongly in } L^2(\Omega). \tag{4.41}$$

It follows from (4.32) and (4.40) that

$$\|\widetilde{w}^k - w^k\|_{L^2(\Omega)}^2 = \|\widetilde{u}_0^k - u_0^k\|_{L^2(\Omega)}^2 + \|\widetilde{p}^k - p^k\|_{L^2(\Omega)}^2 \to 0,$$

which, in particular, yields

$$\|\widetilde{p}^k - p^k\|_{L^2(\Omega)}^2 \to 0, \quad \text{and} \quad \|\widetilde{u}_0^k - u_0^k\|_{L^2(\Omega)}^2 \to 0, \quad \text{as} \quad k \to \infty.$$

This, together with (4.41), implies that

$$p^k \to p^*,\ u_0^k \to u_0^* \quad \text{strongly in } L^2(\Omega).$$

Our proof is then concluded. □

4.4.3 Convergence rate of Algorithm 4.1

In this subsection, we analyze the convergence rate of Algorithm 4.1. In particular, we establish an $O(1/K)$ worst-case convergence rate in both ergodic and non-ergodic senses.

An $O(1/K)$ worst-case convergence rate means that an iterate whose accuracy to the solution under certain criterion is of the order $O(1/K)$ can be found after K iterations of an iterative scheme. This can also be understood as the need of at most $O(1/\varepsilon)$ iterations to find an approximate solution with an accuracy of ε. Besides, we emphasize that such a convergence rate is in the worst-case nature, meaning that it provides a worst-case but universal estimate on the speed of convergence. Hence, it does not contradict with some much faster speeds which might be observed empirically for a specific application (as to be shown in Section 4.7).

Convergence rate in the ergodic sense

We first establish the $O(1/K)$ worst-case convergence rate in the ergodic sense for Algorithm 4.1 in the following theorem.

Theorem 4.7. *Let $w^k = (u_0^k, p^k)^\top$ be the sequence generated by Algorithm 4.1 and $w^* = (u_0^*, p^*)^\top$ be the solution of the problem* (4.12)*. For any $K \in \mathbb{N}$, define*

$$w_K = \frac{1}{K+1}\sum_{k=0}^{K} \widetilde{w}^k, \ \text{and } u_{0,K} = \frac{1}{K+1}\sum_{k=0}^{K} \widetilde{u}_0^k. \tag{4.42}$$

Then we have

$$\varphi(u_{0,K}) - \varphi(u_0^*) + \Big(w_K - w^*, F(w^*)\Big) \leq \frac{1}{2(K+1)}\|w^0 - w^*\|_{\mathcal{K}}^2.$$

Proof. Setting $w = w^*$ in (4.39), it follows from the monotonicity of F that

$$\varphi(\widetilde{u}_0^k) - \varphi(u_0^*) + \Big(\widetilde{w}^k - w^*, F(w^*)\Big) \leq \frac{1}{2}\Big(\|w^k - w^*\|_{\mathcal{K}}^2 - \|w^{k+1} - w^*\|_{\mathcal{K}}^2\Big). \tag{4.43}$$

Summing the inequality (4.43) over $k = 0, \dots K$, we then have

$$\frac{1}{K+1}\sum_{k=0}^{K}\Big(\varphi(\widetilde{u}_0^k)-\varphi(u_0^*)\Big)+\left(\frac{1}{K+1}\sum_{k=0}^{K}\widetilde{w}^k - w^*, F(w^*)\right) \leq \frac{1}{2(K+1)}\|w^0-w^*\|_{\mathcal{K}}^2.$$

Then from the convexity of g and (4.42), we immediately obtain

$$\varphi(u_{0,K}) - \varphi(u_0^*) + \Big(w_K - w^*, F(w^*)\Big) \leq \frac{1}{2(K+1)}\|w^0 - w^*\|_{\mathcal{K}}^2.$$

□

The above theorem shows that, after K iterations of Algorithm 4.1, we can find an approximate solution with an $O(1/K)$ accuracy. This approximate solution is given by w_K, and it is the average of all the points w^k which can be computed by all the known iterates generated Algorithm 4.1. Hence, this is an $O(1/K)$ worst-case convergence rate in the ergodic sense for Algorithm 4.1.

Convergence rate in the non-ergodic sense

Next, we establish the $O(1/K)$ worst-case convergence rate in a non-ergodic sense for Algorithm 4.1. For this purpose, we first need to define a criterion to precisely measure the accuracy of an iterate.

It follows from (4.29) and $G = \mathcal{K}\mathcal{D}$ that the specific sequence $\{w^k\}_{k\geq 0}$ with w^k given by Algorithm 4.1 is a solution point of (4.22) if $\|\mathcal{D}(w^k - \widetilde{w}^k)\|_{\mathcal{K}} = 0$. Hence, it is reasonable to use $\|\mathcal{D}(w^k - \widetilde{w}^k)\|_{\mathcal{K}}$ or $\|\mathcal{D}(w^k - \widetilde{w}^k)\|_{\mathcal{K}}^2$ to measure the accuracy of an iterate w^k to a solution point. We have the following result.

Theorem 4.8. *Let $w^k = (u_0^k, p^k)^\top$ be the sequence generated by Algorithm 4.1 and $w^* = (u_0^*, p^*)^\top$ be the solution of the problem* (4.12)*. Then for any $K \in \mathbb{N}$, we have*

$$\|\mathcal{D}(w^K - \widetilde{w}^K)\|_{\mathcal{K}}^2 \leq \frac{1}{c_0(K+1)}\|w^0 - w^*\|_{\mathcal{K}}^2. \tag{4.44}$$

Proof. We set $w = \widetilde{w}^{k+1}$ in (4.29) and obtain

$$\varphi(\widetilde{u}_0^{k+1}) - \varphi(\widetilde{u}_0^k) + \Big(\widetilde{w}^{k+1} - \widetilde{w}^k, F(\widetilde{w}^k) + G(\widetilde{w}^k - w^k)\Big) \geq 0. \tag{4.45}$$

Moreover, we notice that (4.29) also holds for $k := k+1$, which yields

$$\varphi(u_0) - \varphi(\widetilde{u}_0^{k+1}) + \Big(w - \widetilde{w}^{k+1}, F(\widetilde{w}^{k+1}) + G(\widetilde{w}^{k+1} - w^{k+1})\Big) \geq 0, \quad \forall w \in W.$$

Let $w = \widetilde{w}^k$ in the above inequality. Hence, we have that

$$\varphi(\widetilde{u}_0^k) - \varphi(\widetilde{u}_0^{k+1}) + \Big(\widetilde{w}^k - \widetilde{w}^{k+1}, F(\widetilde{w}^{k+1}) + G(\widetilde{w}^{k+1} - w^{k+1})\Big) \geq 0. \tag{4.46}$$

Adding up (4.45) and (4.46), and taking into account (4.24), we obtain that

$$\Big(\widetilde{w}^k - \widetilde{w}^{k+1}, G(\widetilde{w}^{k+1} - w^{k+1}) - G(\widetilde{w}^k - w^k)\Big) \geq 0.$$

Furthermore, observing that $\widetilde{w}^k - \widetilde{w}^{k+1} = \widetilde{w}^k - \widetilde{w}^{k+1} + w^k - w^k + w^{k+1} - w^{k+1}$, the above inequality yields

$$\Big(w^k - w^{k+1}, G(\widetilde{w}^{k+1} - w^{k+1}) - G(\widetilde{w}^k - w^k)\Big) \geq \frac{1}{2}\|(\widetilde{w}^k - w^k) - (\widetilde{w}^{k+1} - w^{k+1})\|^2_{G^*+G}, \tag{4.47}$$

where we used the fact that

$$\Big(w, Gw\Big) = \frac{1}{2}\Big(w, (G^* + G)w\Big), \quad \forall w \in W.$$

It follows from (4.28) and (4.30) that (4.47) is equivalent to

$$\Big(w^k - \widetilde{w}^k, \mathcal{D}^*\mathcal{K}\mathcal{D}\big((\widetilde{w}^{k+1} - w^{k+1}) - (\widetilde{w}^k - w^k)\big)\Big) \geq \frac{1}{2}\|(\widetilde{w}^k - w^k) - (\widetilde{w}^{k+1} - w^{k+1})\|^2_{G^*+G}. \tag{4.48}$$

Applying the identity

$$\big(a, \mathcal{K}(a-b)\big) = \frac{1}{2}\left(\|a\|^2_{\mathcal{K}} - \|b\|^2_{\mathcal{K}} + \|a-b\|^2_{\mathcal{K}}\right)$$

to the left-hand side of (4.48) with $a = \mathcal{D}(w^k - \widetilde{w}^k)$ and $b = \mathcal{D}(w^{k+1} - \widetilde{w}^{k+1})$, we obtain

$$\begin{aligned} &\Big(w^k - \widetilde{w}^k, \mathcal{D}^*\mathcal{K}\mathcal{D}\big((\widetilde{w}^{k+1} - w^{k+1}) - (\widetilde{w}^k - w^k)\big)\Big) \qquad (4.49)\\ &= \frac{1}{2}\|\mathcal{D}(w^k - \widetilde{w}^k)\|^2_{\mathcal{K}} - \frac{1}{2}\|\mathcal{D}(w^{k+1} - \widetilde{w}^{k+1})\|^2_{\mathcal{K}} + \frac{1}{2}\|\mathcal{D}(w^k - \widetilde{w}^k) - \mathcal{D}(w^{k+1} - \widetilde{w}^{k+1})\|^2_{\mathcal{K}}. \end{aligned}$$

Combining (4.48) and (4.49), we thus obtain

$$\begin{aligned} &\|\mathcal{D}(w^k - \widetilde{w}^k)\|^2_{\mathcal{K}} - \|\mathcal{D}(w^{k+1} - \widetilde{w}^{k+1})\|^2_{\mathcal{K}}\\ \geq &\|(\widetilde{w}^k - w^k) - (\widetilde{w}^{k+1} - w^{k+1})\|^2_{G^*+G} - \|\mathcal{D}(w^k - \widetilde{w}^k) - \mathcal{D}(w^{k+1} - \widetilde{w}^{k+1})\|^2_{\mathcal{K}} \end{aligned}$$

$$=\|(\widetilde{w}^k - w^k) - (\widetilde{w}^{k+1} - w^{k+1})\|^2_{G^*+G-\mathcal{D}^*\mathcal{K}\mathcal{D}} \geq 0.$$

This implies that the sequence $\|\mathcal{D}(w^k - \widetilde{w}^k)\|^2_{\mathcal{K}}$ is non-increasing, i.e.

$$\|\mathcal{D}(w^{k+1} - \widetilde{w}^{k+1})\|^2_{\mathcal{K}} \leq \|\mathcal{D}(w^k - \widetilde{w}^k)\|^2_{\mathcal{K}}, \quad \forall k \geq 0. \tag{4.50}$$

Furthermore, it follows from (4.33) and (4.32) that there exists a positive constant $c_0 > 0$ such that

$$\|w^{k+1} - w^*\|^2_{\mathcal{K}} \leq \|w^k - w^*\|^2_{\mathcal{K}} - c_0\|\mathcal{D}(w^k - \widetilde{w}^k)\|^2_{\mathcal{K}},$$

which implies that

$$c_0 \sum_{k=0}^{\infty} \|\mathcal{D}(w^k - \widetilde{w}^k)\|^2_{\mathcal{K}} \leq \|w^0 - w^*\|^2_{\mathcal{K}}. \tag{4.51}$$

Therefore, it follows from (4.50) and (4.51) that for any integer $K > 0$, we have

$$(K+1)\|\mathcal{D}(w^K - \widetilde{w}^K)\|^2_{\mathcal{K}} \leq \sum_{k=0}^{K} \|\mathcal{D}(w^k - \widetilde{w}^k)\|^2_{\mathcal{K}} \leq \frac{1}{c_0}\|w^0 - w^*\|^2_{\mathcal{K}}.$$

Our proof is then complete. □

We note that the number in the right-hand side of (4.44) is of order $O(1/K)$. Therefore, Theorem 4.8 provides an $O(1/K)$ worst-case convergence rate in a non-ergodic sense for Algorithm 4.1.

4.5 Space and time discretization

In this section, we describe the space-time discretization scheme employed in our numerical simulations. Letting $\mathbf{u} : [0, T] \to \mathcal{R}^{N_x}$, where N_x is the number of grid points on Ω, a general discretization of the diffusion-advection equation in (4.1) can be written in a compact form as:

$$\mathbf{M}\dot{\mathbf{u}}(t) + d\mathbf{A}\mathbf{u}(t) + v\mathbf{V}\mathbf{u}(\mathbf{t}) = \mathbf{0},$$

where the matrices $\mathbf{A}$ and $\mathbf{V}$ are associated with the Laplacian operator and the advection field, respectively, while $\mathbf{M}$ is the mass matrix.

In order to get a time discretized version of the above expression, we first introduce a uniform partition of the time interval $[0, T]$:

$$0 = t_0 < t_1 < \ldots < t_n < t_{n+1} < \ldots < t_{N_t} = T,$$

with $t_n = n\Delta t$ for $n = 0, 1, \ldots, N_t$ and $\Delta t := T/N_t$. We then denote $\mathbf{u}^n = \mathbf{u}(t_n) \in \mathcal{R}^{N_x}$ and apply an implicit Euler method on the mesh $\{t_n\}_{n=0}^{N_t}$. The fully discrete version of (4.1) thus reads as follows

$$(\mathbf{M} + d\Delta t\mathbf{A} + v\Delta t\mathbf{V})\mathbf{u}^n = \mathbf{M}\mathbf{u}^{n-1}.$$

Summarizing, the fully discrete scheme for the numerical resolution of the forward dynamics (4.1) reads as: given $\mathbf{u}^0 = u_0$, then for $n = 1, 2, \ldots, N_t$, solve

$$\mathbf{u}^n = (\mathbf{M} + d\Delta t\mathbf{A} + v\Delta t\mathbf{V})\backslash\mathbf{M}\mathbf{u}^{n-1} \tag{4.52}$$

In the same spirit, the fully discrete scheme for the numerical resolution of the backward dynamics (4.8) and (4.17) reads as: given $\boldsymbol{\psi}^{N_t} = \psi_T$, for $n = N_t, N_t - 1, \ldots, 1$, solve

$$\boldsymbol{\psi}^{n-1} = (\mathbf{M} + d\Delta t\mathbf{A} - v\Delta t\mathbf{V})\backslash\mathbf{M}\boldsymbol{\psi}^n$$

Note that the above fully discrete schemes are presented in a general form which is preserved for any choice of the spatial discretization method, e.g., finite element (FE) methods, finite difference (FD) methods or finite volume (FV) methods. The discretization scheme chosen will only change the specific entries of the matrices $\mathbf{M}$, $\mathbf{A}$ and $\mathbf{V}$.

In this paper, we will always use FE on equidistant structured meshes. For this FE discretization, we employ triangular elements depicted in Figure 4.2 and

the following pyramidal test functions

$$\phi_k(x,y) = \begin{cases} \dfrac{x+y}{\Delta x} - 1, & \text{if } (x,y) \in \text{Region 1} \\ \dfrac{y}{\Delta x}, & \text{if } (x,y) \in \text{Region 2} \\ \dfrac{\Delta x - x}{\Delta x}, & \text{if } (x,y) \in \text{Region 3} \\ 1 - \dfrac{x+y}{\Delta x}, & \text{if } (x,y) \in \text{Region 4} \\ \dfrac{\Delta x - y}{\Delta x}, & \text{if } (x,y) \in \text{Region 5} \\ \dfrac{x}{\Delta x}, & \text{if } (x,y) \in \text{Region 6} \\ 0, & \text{otherwise} \end{cases} \tag{4.53}$$

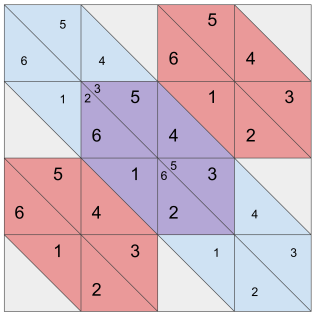


Figure 4.2: Sketch of the regions for the pyramidal test functions defined in (4.53).

4.6 A structure enhancement stage for identifying the optimal locations and intensities

As discussed in the introduction, due to the strong smoothing property of the forward equation (4.1) and the backward equation (4.17), the numerical solution of the optimal control problem (4.4) is not sparse as desired. This suggests the need of introducing a second procedure to project the obtained non-sparse initial source into the set of admissible sparse solutions in the form of (4.3) and identify the locations $\widehat{x}^* := \{\widehat{x}_i^*\}_{i=1}^{\ell}$ and the intensities $\widehat{\boldsymbol{\alpha}}^* := \{\widehat{\alpha}_i^*\}_{i=1}^{\ell}$. We thus obtain an optimal control based two-stage numerical approach for solving Problem 4.1.

4.6.1 Optimal locations identification

To identify the optimal locations, we recall (see (4.3)) that the initial condition to be recovered is assumed to be a linear combination of Dirac measures. Then, it follows from Theorem 4.3 that the optimal locations $\widehat{x}^*$ can be determined by solving

$$\widehat{x}^* = \arg\max_{x\in\Omega} |\lambda_{u_0^*}(x)|.$$

Notice that $\lambda_{u_0^*} = -\alpha u_0^* - \psi^*(\cdot, 0)$, which implies that

$$\widehat{x}^* = \arg\max_{x\in\Omega} |\alpha u_0^*(x) + \psi^*(x, 0)|. \tag{4.54}$$

Hence, with the solution u_0^* obtained by the PDHG algorithms described in Section 4.3, we can compute the associated adjoint variable $\psi^*(\cdot, 0)$ by solving the state equation (4.1) and the adjoint equation (4.8). Then, the optimal locations can be determined via the solution of (4.54).

Additionally, we shall mention that, in [134], the optimal locations $\widehat{x}^*$ is identified by solving

$$\widehat{x}^* = \arg\max_{x\in\Omega} |u_0^*(x)|, \tag{4.55}$$

which is empirically derived from an observation that the local maxima of $|u_0^*(x)|$ fall into the optimal locations. Despite the fact that the strategy (4.55) works well in practice (at least for the numerical examples presented in [134]), in contrary to (4.54), its validity seems to lack rigorous theoretical support.

4.6.2 Optimal intensities identification

In this subsection, we explain how to find the intensities $\{\widehat{\alpha}_i^*\}_{i=1}^{\ell}$ of the initial sources once we have identified their locations $\{\widehat{x}_i^*\}_{i=1}^{\ell}$ by solving (4.54). To this end, we first note that the state equation (4.1) is linear. As a consequence, for any $u_0 = \sum_{i=1}^{\ell} \alpha_i \delta(x_i)$ with $\alpha_i \in \mathbb{R}$ and $x_i \in \Omega$, the solution operator $\mathcal{L}$ verifies

$$\mathcal{L}u_0 = \sum_{i=1}^{\ell} \alpha_i \mathcal{L}\delta(x_i), \quad x_i \in \Omega.$$

Recall that we aim at identifying a sparse initial condition u_0 such that $\mathcal{L}u_0$ is as close as possible to the given target u_d. Hence, to find the optimal intensities of the initial source, we follow [134] and consider the following least square problem:

$$\{\widehat{\alpha}_i^*\}_{i=1}^{\ell} = \arg\min_{\{\alpha_i\}_{i=1}^{\ell}\in\mathbb{R}^{\ell}} \frac{1}{2}\left\|\sum_{i=1}^{\ell}\alpha_i\mathcal{L}\delta(\widehat{x}_i^*) - u_d\right\|_{L^2(\Omega)}^2. \tag{4.56}$$

By the numerical scheme described in Section 4.5, the discretized formulation of (4.56) reads

$$\widehat{\boldsymbol{\alpha}}^* = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^{\ell}} \frac{1}{2}\|\mathbf{L}\boldsymbol{\alpha} - \boldsymbol{u_d}\|^2, \tag{4.57}$$

where $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^{\ell}$, the vector $\boldsymbol{u_d} \in \mathbb{R}^{N_x}$ is a discretized version of u_d, and each column of the matrix $\mathbf{L} \in \mathcal{R}^{N_x\times\ell}$ contains the solution of (4.1) with $u(x,0) = \delta(\widehat{x}_i^*), 1 \le i \le \ell$. Note that the support of the desired sparse initial source usually consists of a few points, i.e. ℓ is generally small. Hence, the dimension of the problem (4.57) is low and it can be solved efficiently through various existing techniques. Here, we suggest to solve the corresponding normal equation

$$\mathbf{L}^\top\mathbf{L}\widehat{\boldsymbol{\alpha}}^* = \mathbf{L}^\top\boldsymbol{u_d}, \tag{4.58}$$

to find the vector of intensities $\widehat{\boldsymbol{\alpha}}^*$. Clearly, the problem (4.58) is a $\ell\times\ell$ symmetric positive definite linear system and can be easily solved.

With the computed locations $\{\widehat{x}_i^*\}_{i=1}^{\ell}$ and intensities $\{\widehat{\alpha}_i^*\}_{i=1}^{\ell}$, the recovered initial source is thus given by

$$\widehat{u}_0^* = \sum_{i=1}^{\ell}\widehat{\alpha}_i^*\delta(\widehat{x}_i^*).$$

4.6.3 An optimal control based two-stage numerical approach for Problem 4.1

In view of the above considerations, the procedure for our source identification Problem 4.1 needs to be complemented with the structure enhancement stage we just described. The complete methodology is given by Algorithm 4.2.

Algorithm 4.2 An optimal control based two-stage numerical approach for Problem 4.1.

procedure SparseIdentification(u_d)

compute u_0^* from the optimal control model (4.4) by Algorithm 4.1;

compute $\psi^*(\cdot, 0)$ by solving the state equation (4.1) and the adjoint equation (4.8);

find the locations $\{\widehat{x}_i^*\}_{i=1}^{\ell}$ by solving $\widehat{x}^* = \arg\max_{x\in\Omega} |\alpha u_0^*(x) + \psi^*(x, 0)|$.

for $i = 1, 2, \ldots, \ell$ **do**

compute $\mathbf{L}(:, i)$ by solving (4.52) with $\boldsymbol{u}^0 = \delta(\widehat{x}_i^*)$

end for

$\widehat{\boldsymbol{\alpha}}^* = (\mathbf{L}^\top \mathbf{L}) \backslash \mathbf{L}^\top \boldsymbol{u_d}$

compute $\widehat{u}_0^* = \sum_{i=1}^{\ell} \widehat{\alpha}_i^* \delta(\widehat{x}_i^*)$

4.7 Numerical experiments

In this section, we show several test cases where Algorithm 4.2 allows identifying the sparse initial sources very successfully from reachable targets or noisy observations, even for some heterogeneous materials or coupled models. All our numerical results have been produced by implementing the aforementioned procedures in MATLAB R2016b on a Surface Pro 5 laptop with 64-bit Windows 10.0 operation system, Intel(R) Core(TM) i7-7660U CPU (2.50 GHz), and 16 GB RAM.

4.7.1 Generalities

We consider Problem 4.1 on the domain $\Omega \times (0, T)$ with $\Omega = (0, 2) \times (0, 1)$ and $T = 0.1$, and we test Algorithm 4.2 for two scenarios:

Scenario 1: the given function u_d is reachable.

Scenario 2: the given function u_d is observed with noise.

For each scenario, we further consider the following three cases:

Case I: diffusivity coefficient $d = 0.05$; advection vector $v = (2, -2)^\top$ on Ω.

Case II: diffusivity coefficient $d = 0.08$ on $\Omega_1 = (0, 1) \times (0, 1)$ and $d = 0.05$ on $\Omega_2 = (1, 2) \times (0, 1)$; advection vector $v = (1, 2)^\top$ on Ω.

Case III: diffusivity coefficient $d = 0.05$ on Ω; advection vector $v = (0, 0)^\top$ on $\Omega_1 = (0, 1) \times (0, 1)$ and $v = (0, -3)^\top$ on $\Omega_2 = (1, 2) \times (0, 1)$.

In Case I, several sources are to be identified in a homogeneous medium, namely, the domain Ω is constituted by materials with same diffusivity constants.

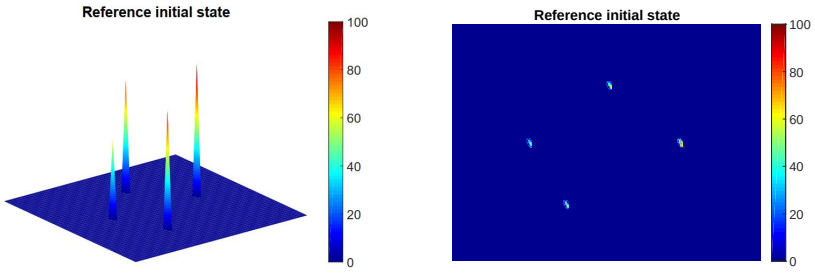
In Case II, we consider the advection-diffusion equation modeled in a heterogeneous medium. To be concrete, the left half subdomain $\Omega_1 = (0, 1) \times (0, 1)$

and the right half one $\Omega_2 = (1,2) \times (0,1)$ are constituted by materials with different diffusivity constants. Consequently, the dynamics of the problem behaves differently in each of them.

Finally, in Case III, we identify several initial sources for coupled-models. This means that different equations are modeled on the left half ($\Omega_1 = (0,1) \times (0,1)$) and the right half ($\Omega_2 = (1,2) \times (0,1)$) of the domain Ω. More precisely, the heat equation is used on Ω_1 and the diffusion-advection equation is used on Ω_2.

The reference initial datum $\widehat{u}_0$ to be recovered for all cases is displayed in Figure 4.3 from front and above views.

Figure 4.3: Reference initial datum $\widehat{u}_0$ for Cases I-III, front view (left) and above view (right)



We implement the PDHG method (4.13)-(4.15) and the PDHG-PC method derived from Algorithm 4.1 to solve the optimal control problem (4.4). Both of them are repeated until the following stopping criterion is fulfilled:

$$e_k := \max\left\{\frac{\|u^{k+1}-u^k\|_{L^2(\Omega)}}{\|u^{k+1}\|_{L^2(\Omega)}}, \frac{\|p^{k+1}-p^k\|_{L^2(\Omega)}}{\|p^{k+1}\|_{L^2(\Omega)}}\right\} \leq tol_{PDHG}$$

with $tol_{PDHG} = 10^{-5}$ or until we reach a maximum number of iterations $k_{max} = 1000$. Moreover, if there are no other specifications, we always use the following parameters:

- Mesh sizes: $\Delta x = 0.02$ and $\Delta t = 0.05$.
- Regularization parameters: $\beta = (\Delta x)^4, \alpha = 10^{-2}$.
- PDHG algorithm: $\theta = 1, r = 6, s = 0.578 (\approx \frac{0.999}{r\|\mathcal{L}^*\mathcal{L}\|})$.

- PDHG-PC algorithm: $\theta = 1, r = 6, s = 0.578, \rho = \sigma = 1.9$.
- Initial values: $u_0^0 = 0, p^0 = 0$.

Moreover, we compare the numerical efficiency of our approach with the one described in [134], and show that our methodology yields significant improvements in the performance of the initial source identification procedure. For completeness, we review the approach in [134] briefly.

In [134], the initial source identification problem 4.1 was formulated as an optimal control model but in the absence of a L^2-regularization in the cost functional (that is, when taking $\alpha = 0$ in (4.4)). To address the resulting optimal control problem numerically, a GD approach was employed, which consists of looking for the minimizer u_0^* as the limit $k \to +\infty$ of the following iterative process:

$$u_0^{k+1} = u_0^k - \eta_k \nabla J(u_0^k). \tag{4.59}$$

In (4.59), the parameter $\eta_k > 0$ is called the step-size and plays a fundamental role in the convergence of the scheme. It is by now well-known that, if one takes η_k constant small enough and the objective functional is regular enough (namely, convex, differentiable, and with Lipschitz gradient), then (4.59) will eventually converge to the minimum (see, e.g., [136, Section 2.1.5]).

Nevertheless, the choice of a constant step-size is most often not optimal: if η_k is too small, the convergence velocity of GD may drastically decrease while, if η_k is too large, one can generate overshooting phenomena and not be able to reach the minimum of J. Hence, in numerical implementations, an adaptive choice of the step-size is usually introduced (e.g., Armijo line search). In this regard, it is worth recalling that these adaptive strategies require the evaluation of the objective function value repeatedly, which in our case is numerically expensive because each one of these evaluations requires solving a forward equation. For the above reasons, in our implementation of GD we always considered a constant step-size although, as we shall see, this choice contributes to making the GD methodology less efficient when solving the source identification problem for (4.1).

Finally, recall that the gradient $\nabla J(u_0)$ has already been computed in (4.9) and is given by the expression

$$\nabla J(u_0) = \psi(\cdot, 0) + \alpha u_0 + \lambda_{u_0}.$$

Consequently, the iterative scheme (4.59) becomes

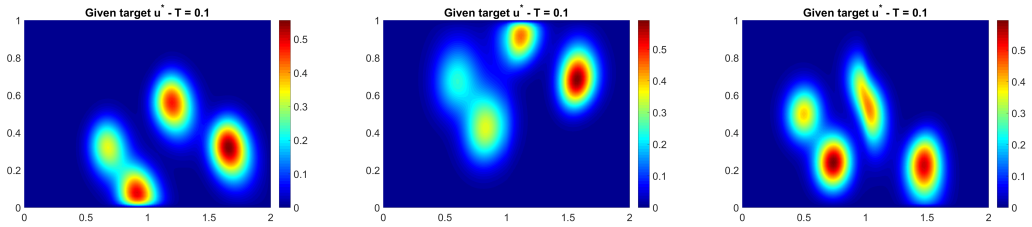
$$u_0^{k+1} = u_0^k - \eta_k(\psi_0^k + \alpha u_0^k + \lambda_{u_0^k})$$

with $\psi_0^k = \psi^k(\cdot, 0)$. It is clear that the computational load of each GD iteration is the same as that of the PDHG. However, we shall stress that the convergence of the GD algorithm can be proved rigorously only in the case of regular enough functionals. In other words, if the gradient presents discontinuities, convergence is not guaranteed. This is exactly the case of the optimal control problem (4.4), as we already discussed in Section 4.2. Notwithstanding that, as we will see in our numerical simulations, for the test cases that we will consider the GD methodology is still capable to compute the correct solution u_0^*. This is due to the fact that the discontinuous part of the gradient λ_{u_0} is always bounded by the parameter β (see (4.10)).

4.7.2 Reachable target u_d case

We first test Algorithm 4.2 for Problem 4.1 where the target function u_d is reachable. In particular, we set the target function u_d as the solution of (4.1) corresponding to the initial condition $u(x, 0) = \widehat{u}_0$. The reference u_d for Case I–III is presented in Figure 4.4.

Figure 4.4: The reference target u_d for Case I-III (from left to right).



We apply the PDHG, PDHG-PC and the GD method in [134] to the model (4.4). The efficiency (in terms of the number of iterations to converge) is collected

in Table 4.1. First, from Table 4.1, we observe that the iteration numbers of the PDHG and PDHG-PC algorithms are almost the same for all cases. We thus conclude that the convergence of the PDHG and PDHG-PC algorithms are robust with respect to the diffusion coefficient d and the convection coefficient v, at least for the cases we considered. We also observe from Table 4.1 that the PDHG-PC algorithm improves the numerical efficiency of PDHG by a factor about 40%, and both of them are more efficient than the GD method.

Table 4.1: Numerical comparisons (in terms of the number of iterations to converge) of different algorithms for Cases I-III.

	Model (4.4)			Model in [134]		
	PDHG	PDHG-PC	GD	PDHG	PDHG-PC	GD
Case I	53	32	86	629	589	673
Case II	54	32	87	632	612	650
Case III	52	32	87	648	601	667

Furthermore, we recall that the PDHG and PDHG-PC algorithms are described on the continuous level and their convergence properties are analyzed in function spaces. Hence, mesh independent property of these algorithms can be expected in practice, which means that the convergence behavior is independent of the fineness of the discretization. This is confirmed by our numerical results presented in Table 4.2.

Table 4.2: Iteration numbers with respect to different mesh sizes for Case I

Mesh size	$\Delta t = 0.1, \Delta x = 0.05$	$\Delta t = 0.05, \Delta x = 0.02$	$\Delta t = 0.025, \Delta x = 0.0125$
PDHG	61	53	49
PDHG-PC	37	32	29

For comparison purposes, we also implement the PDHG, PDHG-PC, and GD methods for the model introduced in [134]. The efficiency of each methodology is once again collected in Table 4.1. It is not surprising that a significantly higher number of iterations is required because the model considered in [134] excludes the term $\frac{\alpha}{2}\int_\Omega |u_0|^2\,dx$ and is much more ill-conditioned than the one we considered.

For Case I, the recovered initial datum $\widehat{u}_0^*$ by Algorithm 4.2 and the corresponding final state u_T are displayed in Figure 4.5. By comparing the plots in

Figures 4.3 and 4.5, one can observe that both the locations and the intensities of the initial condition are recovered very accurately. Furthermore, the recovered initial and final states by the approach described in [134] are presented in Figure 4.6, which is almost the same as the one obtained by Algorithm 4.2c. All these facts validate that the effectiveness and efficiency of our proposed approach.

Figure 4.5: The recovered initial datum $\widehat{u}_0^*$ from front view (left) and above view (middle), and the recovered target u_T (right) by Algorithm 4.2 for Case I with a reachable u_d.

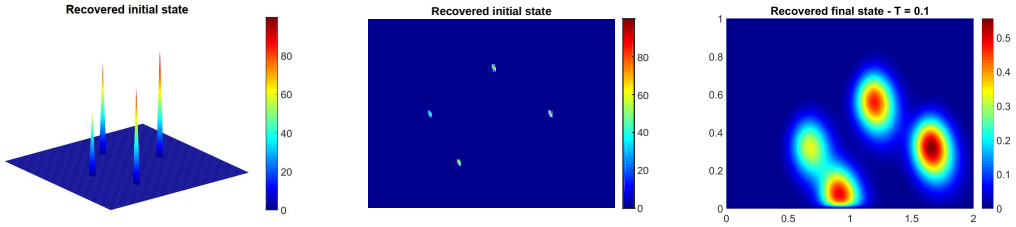
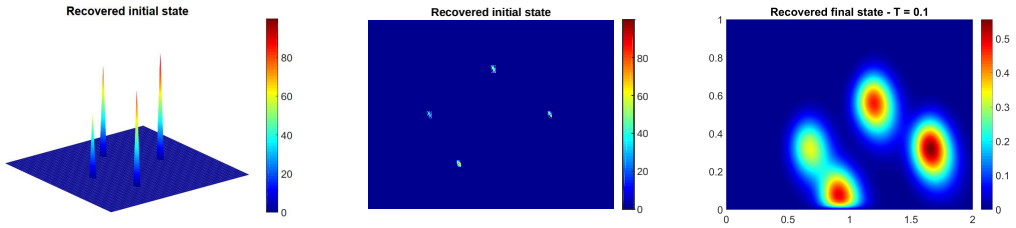


Figure 4.6: The recovered initial datum $\widehat{u}_0^*$ from front view (left) and above view (middle), and the recovered target u_T (right) by the approach in [134] for Case I with a reachable u_d.



Similar to Case I, the results in Table 4.1 show that also in Case II and Case III the PDHG-PC algorithm is the most efficient one and, compared with [134], our proposed model (4.4) allows for a more efficient numerical resolution. Furthermore, compared with Figures 4.3 and 4.4, the recovered initial datum $\widehat{u}_0^*$ by Algorithm 4.2 and the corresponding final state u_T displayed in Figure 4.7 (Case II) and Figure 4.9 (Case III) show that the locations and the intensities of the sparse initial sources are recovered very accurately for heterogeneous materials. Additionally, by comparing the plots in Figures 4.7 and 4.8, we conclude that our results are as accurate as the ones obtained in [134] for Case II, but our approach is numerically more efficient as shown in Table 4.1. The same conclusion can also be drawn for Case III by comparing the plots in Figures 4.9 and 4.10.

Figure 4.7: The recovered initial datum $\widehat{u}_0^*$ from front view (left) and above view (middle), and the recovered target u_T (right) by Algorithm 4.2 for Case II with a reachable u_d.

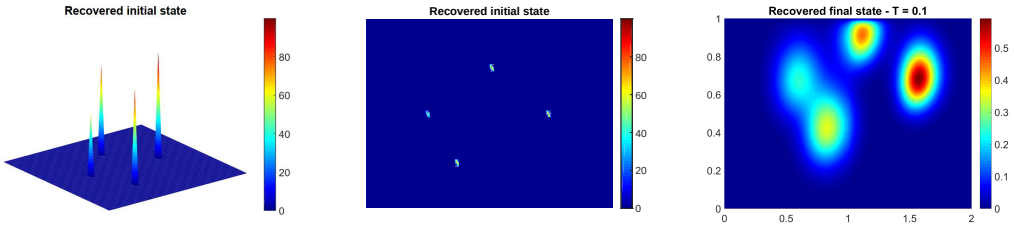


Figure 4.8: The recovered initial datum $\widehat{u}_0^*$ from front view (left) and above view (middle), and the recovered target u_T (right) by the approach in [134] for Case II with a reachable u_d.

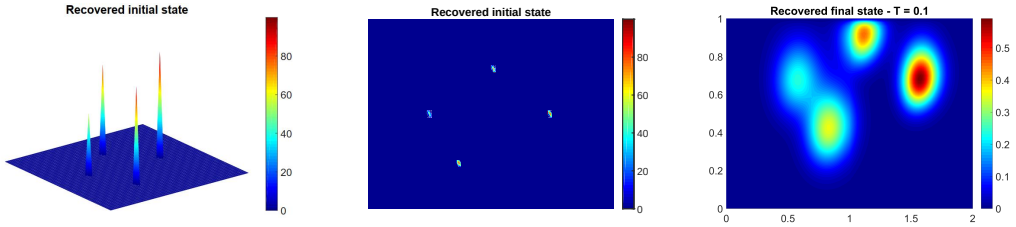


Figure 4.9: The recovered initial datum $\widehat{u}_0^*$ from front view (left) and above view (middle), and the recovered target u_T (right) by Algorithm 4.2 for Case III with a reachable u_d.

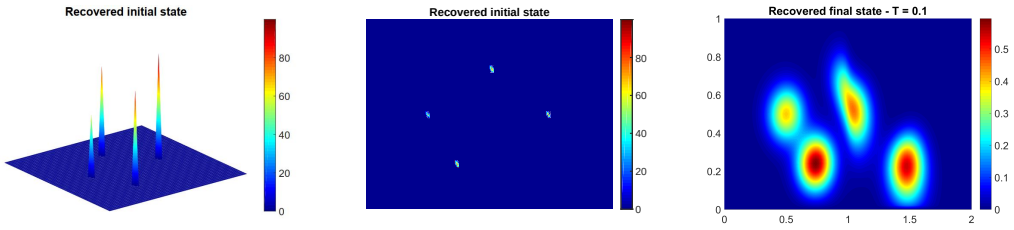
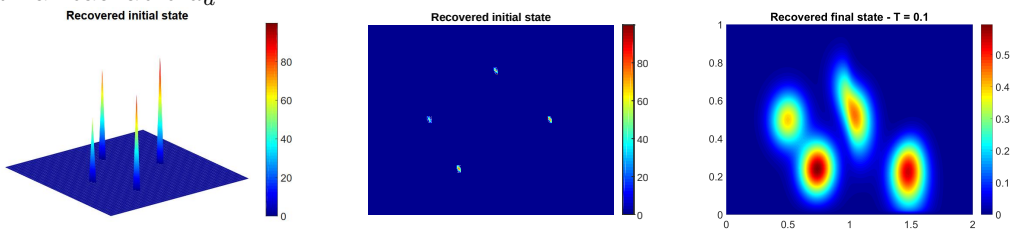


Figure 4.10: The recovered initial datum $\widehat{u}_0^*$ from front view (left) and above view (middle), and the recovered target u_T (right) by the approach in [134] for Case III with a reachable u_d.



4.7.3 Noisy observation u_d case

In this subsection, we aim to identify a sparse initial source term u_0 from noisy observations $u_d = \mathcal{L}u_0 + \delta$ by Algorithm 4.2, where $\delta \in L^2(\Omega)$ is a given noise term satisfying

$$\|\mathcal{L}u_0 - u_d\|/\|\mathcal{L}u_0\| \approx 20\%.$$

For convenience, we still consider a reference initial datum $\widehat{u}_0$ as given in Figure 4.3, and the corresponding noisy observations u_d for Case I-III are displayed in Figure 4.11. As in the previous subsections, we employ the PDHG-PC method to solve the optimal control problem (4.4). We observe that the iteration numbers of the PDHG-PC for all test cases are almost the same as the reachable target case and mesh-independent property can be observed. Hence, we can conclude that the numerical efficiency of the PDHG-PC method is very robust to the noisy observations.

The initial datum $\widehat{u}_0^*$ recovered from the noisy observations u_d by Algorithm 4.2 and the associated final state u_T for Case I-III are presented in Figures 4.12, 4.13 and 4.14. Compared with the reference initial datum in Figure 4.3, we observe that both the locations and the intensities of the sparse initial source are recovered accurately from the noisy observations.

Figure 4.11: The noisy observation u_d for Case I-III (from left to right).

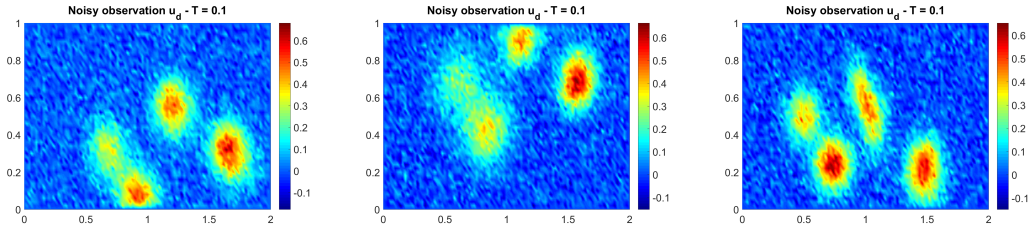


Figure 4.12: The recovered initial datum $\widehat{u}_0^*$ from front view (left) and above view (middle), and the recovered target u_T (right) by Algorithm 4.2 for Case I with a noisy observation u_d.

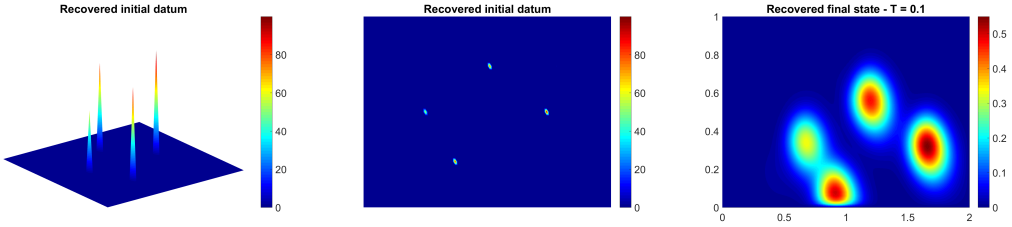


Figure 4.13: The recovered initial datum $\widehat{u}_0^*$ from front view (left) and above view (middle), and the recovered target u_T (right) by Algorithm 4.2 for Case II with a noisy observation u_d.

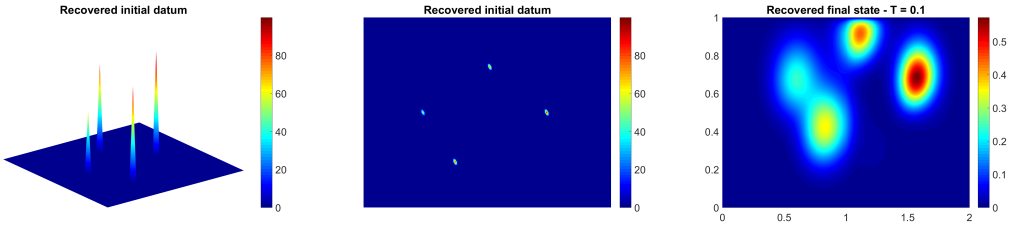
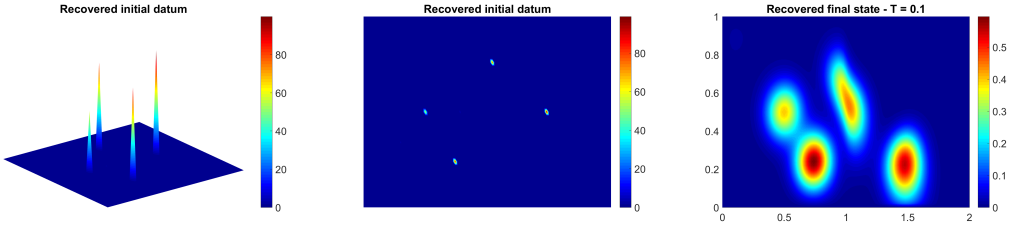


Figure 4.14: The recovered initial datum $\widehat{u}_0^*$ from front view (left) and above view (middle), and the recovered target u_T (right) by Algorithm 4.2 for Case III with a noisy observation u_d.



Chapter 5

Bilinear Optimal Control of an Advection-Reaction-Diffusion System

In this chapter, we intend to study the bilinear optimal control problem (BCP) introduced in Section 1.1 both mathematically and computationally. For convenience, we recall that the bilinear optimal control problem (BCP) reads as

$$\begin{cases} \boldsymbol{u} \in \mathcal{U}, \\ J(\boldsymbol{u}) \leq J(\boldsymbol{v}), \forall \boldsymbol{v} \in \mathcal{U}, \end{cases} \tag{BCP}$$

with the objective functional J defined by

$$J(\boldsymbol{v}) = \frac{1}{2}\iint_Q |\boldsymbol{v}|^2 dxdt + \frac{\alpha_1}{2}\iint_Q |y - y_d|^2 dxdt + \frac{\alpha_2}{2}\int_\Omega |y(T) - y_T|^2 dx, \quad (5.1)$$

and $y = y(t; \boldsymbol{v})$ the solution of the following advection-reaction-diffusion equation

$$\begin{cases} \dfrac{\partial y}{\partial t} - \nu\nabla^2 y + \boldsymbol{v} \cdot \nabla y + a_0 y = f \quad \text{in} \quad Q, \\ \qquad\qquad\qquad\qquad\qquad y = g \quad \text{on} \quad \Sigma, \\ \qquad\qquad\qquad\qquad\qquad y(0) = \phi. \end{cases} \tag{5.2}$$

Above and below, Ω is a bounded domain of $\mathbb{R}^d$ with $d \geq 1$ and Γ is its boundary, $Q = \Omega \times (0, T)$ and $\Sigma = \Gamma \times (0, T)$ with $0 < T < +\infty$; $\alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_1 + \alpha_2 > 0$; the target functions y_d and y_T are given in $L^2(Q)$ and $L^2(\Omega)$, respectively; the

diffusion coefficient $\nu > 0$ and the reaction coefficient a_0 are assumed to be constants; the functions $f \in L^2(Q)$, $g \in L^2(0,T;H^{1/2}(\Gamma))$ and $\phi \in L^2(\Omega)$. The set $\mathcal{U}$ of the admissible controls is defined by

$$\mathcal{U} := \{\boldsymbol{v} | \boldsymbol{v} \in [L^2(Q)]^d, \nabla \cdot \boldsymbol{v} = 0\}.$$

We first study the well-posedness of (5.2), the existence of an optimal control $\boldsymbol{u}$, and its first-order optimality condition. Then, computationally, we propose an efficient and relatively easy to implement numerical method to solve (BCP). For this purpose, we advocate combining a conjugate gradient (CG) method with a finite difference method (for the time discretization) and a finite element method (for the space discretization) for the numerical solution of (BCP). Although these numerical approaches have been well developed in the literature, it is nontrivial to implement them to solve (BCP) as discussed below, due to the complicated problem settings.

5.1 Difficulties and goals

5.1.1 Difficulties in algorithmic design

Conceptually, a CG method for solving (BCP) can be easily derived following [83]. However, CG algorithms are challenging to implement numerically for the following reasons: 1). The state y depends non-linearly on the control $\boldsymbol{v}$ despite the fact that the state equation (5.2) is linear. 2). The additional divergence-free constraint on the control $\boldsymbol{v}$, i.e., $\nabla \cdot \boldsymbol{v} = 0$, is coupled together with the state equation (5.2).

To be more precise, the fact that the state y is a nonlinear function of the control $\boldsymbol{v}$ makes the optimality system a nonlinear problem. Hence, seeking a suitable stepsize in each CG iteration requires solving an optimization problem and it can not be as easily computed as in the linear case [83]. Note that commonly used line search strategies are too expensive to employ in our settings because they require evaluating the objective functional value $J(\boldsymbol{v})$ repeatedly

and every evaluation of $J(\boldsymbol{v})$ entails solving the state equation (5.2). The same concern on the computational cost also applies when the Newton method is employed to solve the corresponding optimization problem for finding a stepsize. To tackle this issue, we propose an efficient inexact stepsize strategy which requires solving only one additional linear parabolic problem and is cheap to implement as shown in Section 5.3.

Furthermore, due to the divergence-free constraint $\nabla \cdot \boldsymbol{v} = 0$, an extra projection onto the admissible set $\mathcal{U}$ is required to compute the first-order differential of J at each CG iteration in order that all iterates of the CG method are feasible. Generally, this projection subproblem has no closed-form solution and has to be solved iteratively. Here, we introduce a Lagrange multiplier associated with the constraint $\nabla \cdot \boldsymbol{v} = 0$, then the computation of the first-order differential $DJ(\boldsymbol{v})$ of J at $\boldsymbol{v}$ is equivalent to solving a Stokes type problem. Inspired by [75], we advocate employing a preconditioned CG method, which operates on the space of the Lagrange multiplier, to solve the resulting Stokes type problem. With an appropriately chosen preconditioner, a fast convergence of the resulting preconditioned CG method can be expected in practice (and indeed, has been observed).

5.1.2 Difficulties in numerical discretization

For the numerical discretization of (BCP), we note that if an implicit finite difference scheme is used for the time discretization of the state equation (5.2), a stationary advection-reaction-diffusion equation should be solved at each time step. To solve this stationary advection-reaction-diffusion equation, it is well known that standard finite element techniques may lead to strongly oscillatory solutions unless the mesh-size is sufficiently small with respect to the ratio between ν and $\|\boldsymbol{v}\|$. In the context of optimal control problems, to overcome such difficulties, different stabilized finite element methods have been proposed and analyzed, see e.g., [9, 48]. Different from the above references, we implement the time discretization by a semi-implicit finite difference method for simplicity, namely, we use explicit advection and reaction terms and treat the diffusion term

implicitly. Consequently, only a simple linear elliptic equation is required to be solved at each time step. We then implement the space discretization of the resulting elliptic equation at each time step by a standard piecewise linear finite element method and the resulting linear system is very easy to solve.

Moreover, we recall that the divergence-free constraint $\nabla \cdot \boldsymbol{v} = 0$ leads to a projection subproblem, which is equivalent to a Stokes type problem, at each iteration of the CG algorithm. As discussed in [74], to discretize a Stokes type problem, direct applications of standard finite element methods always lead to an ill-posed discrete problem. To overcome this difficulty, one can use different types of element approximations for pressure and velocity. Inspired by [74, 75], we employ the Bercovier-Pironneau finite element pair [11] (also known as P_1-P_1 iso P_2 finite element) to approximate the control $\boldsymbol{v}$ and the Lagrange multiplier associated with the divergence-free constraint. More concretely, we approximate the Lagrange multiplier by a piecewise linear finite element space which is twice coarser than the one for the control $\boldsymbol{v}$. In this way, the discrete problem is well-posed and can be solved by a preconditioned CG method. As a byproduct of the above discretization, the total number of degrees of freedom of the discrete Lagrange multiplier is only $\frac{1}{d2^d}$ of the number of the discrete control. Hence, the inner preconditioned CG method is implemented in a lower-dimensional space than that of the state equation (5.2), implying a computational cost reduction. With the above mentioned discretization schemes, we can relatively easily obtain the fully discrete version of (BCP) and derive the discrete analogue of our proposed nested CG method.

5.2 Existence of optimal controls and first-order optimality conditions

In this section, first we present some notation and known results from the literature that will be used in later analysis. Then, we prove the existence of optimal controls for (BCP) and derive the associated first-order optimality conditions. Without loss of generality, we assume that $f = 0$ and $g = 0$ in (1.15) for

convenience.

5.2.1 Preliminaries

Throughout, we denote by $L^s(\Omega)$ and $H^s(\Omega)$ the usual Sobolev spaces for any $s > 0$. The space $H_0^s(\Omega)$ denotes the completion of $C_0^\infty(\Omega)$ in $H^s(\Omega)$, where $C_0^\infty(\Omega)$ denotes the space of all infinitely differentiable functions over Ω with a compact support in Ω. In addition, we shall also use the following vector-valued function spaces:

$$\begin{aligned}
&\boldsymbol{L}^2(\Omega) := [L^2(\Omega)]^d,\\
&\boldsymbol{L}^2_{div}(\Omega) := \{\boldsymbol{v} \in \boldsymbol{L}^2(\Omega), \nabla \cdot \boldsymbol{v} = 0 \text{ in } \Omega\}.
\end{aligned}$$

Let X be a Banach space with a norm $\|\cdot\|_X$. Then, the space $L^2(0,T;X)$ consists of all measurable functions $z : (0,T) \to X$ satisfying

$$\|z\|_{L^2(0,T;X)} := \left(\int_0^T \|z(t)\|_X^2 dt\right)^{\frac{1}{2}} < +\infty.$$

With the above notation, it is clear that the admissible set $\mathcal{U}$ can be denoted as $\mathcal{U} = L^2(0,T;\boldsymbol{L}^2_{div}(\Omega))$. Moreover, the space $W(0,T)$ consists of all functions $z \in L^2(0,T;H_0^1(\Omega))$ such that $\frac{\partial z}{\partial t} \in L^2(0,T;H^{-1}(\Omega))$ exists in a weak sense, i.e.

$$W(0,T) := \{z | z \in L^2(0,T;H_0^1(\Omega)), \frac{\partial z}{\partial t} \in L^2(0,T;H^{-1}(\Omega))\},$$

where $H^{-1}(\Omega)(= H_0^1(\Omega)')$ is the dual space of $H_0^1(\Omega)$.

Next, we summarize some known results for the advection-reaction-diffusion equation (1.15) in the literature for the convenience of further analysis.

The variational formulation of the state equation (1.15) reads: find $y \in W(0,T)$ such that $y(0) = \phi$ and $\forall z \in L^2(0,T;H_0^1(\Omega))$,

$$\begin{aligned}
\int_0^T \left\langle \frac{\partial y}{\partial t}, z \right\rangle_{H^{-1}(\Omega),H_0^1(\Omega)} dt + \nu \iint_Q \nabla y \cdot \nabla z dx dt\\
+ \iint_Q \boldsymbol{v} \cdot \nabla y z dx dt + a_0 \iint_Q y z dx dt = 0,
\end{aligned} \tag{5.3}$$

where $\langle \cdot, \cdot \rangle_{H^{-1}(\Omega),H_0^1(\Omega)}$ denotes the duality pairing between $H^{-1}(\Omega)$ and $H_0^1(\Omega)$. The existence and uniqueness of the solution $y \in W(0,T)$ to problem (5.3) can

be proved by standard arguments relying on the Lax-Milgram theorem; we refer to [125] for the details. For discussions on the space discretization in Section 5.4, we also need the variational formulation of the divergence-free constraint $\nabla \cdot \boldsymbol{v} = 0$:

$$\iint_Q \nabla \cdot \boldsymbol{v} q dx dt \left(= - \iint_Q \boldsymbol{v} \cdot \nabla q dx dt \right) = 0, \forall q \in L^2(0,T; H_0^1(\Omega)).$$

Moreover, we can define the control-to-state operator $S : \mathcal{U} \to W(0,T)$, which maps $\boldsymbol{v}$ to $y = S(\boldsymbol{v})$. Then, the objective functional J in (BCP) can be reformulated as

$$J(\boldsymbol{v}) = \frac{1}{2} \iint_Q |\boldsymbol{v}|^2 dx dt + \frac{\alpha_1}{2} \iint_Q |S(\boldsymbol{v}) - y_d|^2 dx dt + \frac{\alpha_2}{2} \int_\Omega |S(\boldsymbol{v})(T) - y_T|^2 dx,$$

and the nonlinearity of the solution operator S implies that (BCP) is nonconvex.

For the solution $y \in W(0,T)$, we have the following estimates.

Lemma 5.1. *Let $\boldsymbol{v} \in \mathcal{U}$ and then the solution $y \in W(0,T)$ of the state equation (1.15) satisfies the following estimate:*

$$\|y(t)\|^2_{L^2(\Omega)} + 2\nu \int_0^t \|\nabla y(s)\|^2_{L^2(\Omega)} ds + 2a_0 \int_0^t \|y(s)\|^2_{L^2(\Omega)} ds = \|\phi\|^2_{L^2(\Omega)}. \quad (5.4)$$

Proof. We first multiply the state equation (1.15) by $y(t)$ and then applying the Green's formula in space yields

$$\frac{1}{2} \frac{d}{dt} \|y(t)\|^2_{L^2(\Omega)} = -\nu \|\nabla y(t)\|^2_{L^2(\Omega)} - a_0 \|y(t)\|^2_{L^2(\Omega)}. \quad (5.5)$$

The desired result (5.4) can be directly obtained by integrating (5.5) over $[0,t]$. □

Above estimate implies that

$$y \text{ is bounded in } L^2(0,T; H_0^1(\Omega)). \quad (5.6)$$

On the other hand,

$$\frac{\partial y}{\partial t} = \nu \nabla^2 y - \boldsymbol{v} \cdot \nabla y - a_0 y,$$

and the right hand side is bounded in $L^2(0,T; H^{-1}(\Omega))$. Hence,

$$\frac{\partial y}{\partial t} \text{ is bounded in } L^2(0,T; H^{-1}(\Omega)). \quad (5.7)$$

Furthermore, since $\nabla \cdot \boldsymbol{v} = 0$, it is clear that

$$\iint_Q \boldsymbol{v} \cdot \nabla y z dx dt = \iint_Q \nabla y \cdot (\boldsymbol{v} z) dx dt$$
$$= -\iint_Q y \nabla \cdot (\boldsymbol{v} z) dx dt = -\iint_Q y(\boldsymbol{v} \cdot \nabla z) dx dt, \forall z \in L^2(0,T;H_0^1(\Omega)).$$

Hence, the variational formulation (5.3) can be equivalently written as: find $y \in W(0,T)$ such that $y(0) = \phi$ and $\forall z \in L^2(0,T;H_0^1(\Omega))$,

$$\int_0^T \left\langle \frac{\partial y}{\partial t}, z \right\rangle_{H^{-1}(\Omega),H_0^1(\Omega)} dt + \nu \iint_Q \nabla y \cdot \nabla z dx dt - \iint_Q (\boldsymbol{v} \cdot \nabla z) y dx dt + a_0 \iint_Q yz dx dt = 0.$$

5.2.2 Existence of Optimal Controls

With above preparations, we prove in this subsection the existence of optimal controls for (BCP). For this purpose, we first show that the objective functional J is weakly lower semi-continuous.

Lemma 5.2. *The objective functional J given by (1.14) is weakly lower semi-continuous. That is, if a sequence $\{\boldsymbol{v}_n\}$ converges weakly to $\bar{\boldsymbol{v}}$ in $\mathcal{U}$, we have*

$$J(\bar{\boldsymbol{v}}) \leq \liminf_{n\to\infty} J(\boldsymbol{v}_n).$$

Proof. Let $\{\boldsymbol{v}_n\}$ be a sequence that converges weakly to $\bar{\boldsymbol{v}}$ in $\mathcal{U}$, and $y_n := y(x,t;\boldsymbol{v}_n)$ the solution of the following variational problem: find $y_n \in W(0,T)$ such that $y_n(0) = \phi$ and $\forall z \in L^2(0,T;H_0^1(\Omega))$,

$$\begin{aligned} &\int_0^T \left\langle \frac{\partial y_n}{\partial t}, z \right\rangle_{H^{-1}(\Omega),H_0^1(\Omega)} dt + \nu \iint_Q \nabla y_n \cdot \nabla z dx dt \\ &\quad - \iint_Q (\boldsymbol{v}_n \cdot \nabla z) y_n dx dt + a_0 \iint_Q y_n z dx dt = 0. \end{aligned} \tag{5.8}$$

It follows from (5.6) and (5.7) that there exists a subsequence of $\{y_n\}$, still denoted by $\{y_n\}$ for convenience, such that

$$y_n \to \bar{y} \text{ weakly in } L^2(0,T;H_0^1(\Omega)),$$

and

$$\frac{\partial y_n}{\partial t} \to \frac{\partial \bar{y}}{\partial t} \text{ weakly in } L^2(0,T;H^{-1}(\Omega)).$$

Since Ω is bounded, it follows directly from the compactness property (also known as Rellich's Theorem) that

$$y_n \to \bar{y} \text{ strongly in } L^2(0,T;L^2(\Omega)).$$

Taking $\boldsymbol{v}_n \to \bar{\boldsymbol{v}}$ weakly in $\mathcal{U}$ into account, we can pass the limit in (5.8) and derive that $\bar{y}(0) = \phi$ and $\forall z \in L^2(0,T;H_0^1(\Omega))$,

$$\int_0^T \left\langle \frac{\partial \bar{y}}{\partial t}, z \right\rangle_{H^{-1}(\Omega),H_0^1(\Omega)} dt + \nu \iint_Q \nabla \bar{y} \cdot \nabla z dx dt - \iint_Q (\bar{\boldsymbol{v}} \cdot \nabla z) \bar{y} dx dt + a_0 \iint_Q \bar{y} z dx dt = 0,$$

which implies that $\bar{y}$ is the solution of the state equation (1.15) associated with $\bar{\boldsymbol{v}}$.

Since any norm of a Banach space is weakly lower semi-continuous, we have that

$$\begin{aligned}
&\liminf_{n\to\infty} J(\boldsymbol{v}_n) \\
=&\liminf_{n\to\infty} \left(\frac{1}{2} \iint_Q |\boldsymbol{v}_n|^2 dx dt + \frac{\alpha_1}{2} \iint_Q |y_n - y_d|^2 dx dt + \frac{\alpha_2}{2} \int_\Omega |y_n(T) - y_T|^2 dx \right) \\
\geq& \frac{1}{2} \iint_Q |\bar{\boldsymbol{v}}|^2 dx dt + \frac{\alpha_1}{2} \iint_Q |\bar{y} - y_d|^2 dx dt + \frac{\alpha_2}{2} \int_\Omega |\bar{y}(T) - y_T|^2 dx \\
=& J(\bar{\boldsymbol{v}}).
\end{aligned}$$

We thus obtain that the objective functional J is weakly lower semi-continuous and complete the proof. □

Now, we are in a position to prove the existence of an optimal control $\boldsymbol{u}$ to (BCP). Recall the nonconvexity of (BCP). The uniqueness of optimal control $\boldsymbol{u}$ cannot be guaranteed and only a local one can be pursued.

Theorem 5.1. *There exists at least one optimal control $\boldsymbol{u} \in \mathcal{U}$ such that $J(\boldsymbol{u}) \leq J(\boldsymbol{v}), \forall \boldsymbol{v} \in \mathcal{U}$.*

Proof. We first observe that $J(\boldsymbol{v}) \geq 0, \forall \boldsymbol{v} \in \mathcal{U}$. Then, the infimum of $J(\boldsymbol{v})$ exists and we denote it as

$$j = \inf_{\boldsymbol{v} \in \mathcal{U}} J(\boldsymbol{v}),$$

which implies that there is a minimizing sequence $\{\boldsymbol{v}_n\} \subset \mathcal{U}$ such that

$$\lim_{n\to\infty} J(\boldsymbol{v}_n) = j.$$

This fact, together with $\frac{1}{2}\iint_Q |\boldsymbol{v}_n|^2 dxdt \leq J(\boldsymbol{v}_n)$, implies that $\{\boldsymbol{v}_n\}$ is bounded in $\mathcal{U}$. Hence, there exists a subsequence, still denoted by $\{\boldsymbol{v}_n\}$, that converges weakly to $\boldsymbol{u}$ in $\mathcal{U}$. It follows from Lemma 5.2 that J is weakly lower semi-continuous and we thus have

$$J(\boldsymbol{u}) \leq \liminf_{n\to\infty} J(\boldsymbol{v}_n) = j.$$

Since $\boldsymbol{u} \in \mathcal{U}$, we must have $J(\boldsymbol{u}) = j$, and $\boldsymbol{u}$ is therefore an optimal control. □

5.2.3 First-order Optimality Conditions

According to some literatures, e.g. [128], it is easy to know that J is *Gâteaux* differentiable. Let $DJ(\boldsymbol{v})$ be the first-order differential of J at $\boldsymbol{v}$ and $\boldsymbol{u}$ an optimal control of (BCP). It is clear that the first-order optimality condition of (BCP) reads

$$DJ(\boldsymbol{u}) = 0.$$

In the sequel of this subsection, we discuss the computation of $DJ(\boldsymbol{v})$, which will play an important role in subsequent sections.

To compute $DJ(\boldsymbol{v})$, we employ a formal perturbation analysis as in [83]. First, let $\delta\boldsymbol{v} \in \mathcal{U}$ be a perturbation of $\boldsymbol{v} \in \mathcal{U}$, we clearly have

$$\delta J(\boldsymbol{v}) = \iint_Q DJ(\boldsymbol{v}) \cdot \delta\boldsymbol{v} dxdt, \tag{5.9}$$

and also

$$\delta J(\boldsymbol{v}) = \iint_Q \boldsymbol{v} \cdot \delta\boldsymbol{v} dxdt + \alpha_1 \iint_Q (y - y_d)\delta y dxdt + \alpha_2 \int_\Omega (y(T) - y_T)\delta y(T) dx, \tag{5.10}$$

where δy is the solution of the sensitivity equation of (1.15)

$$\begin{cases} \dfrac{\partial \delta y}{\partial t} - \nu\nabla^2\delta y + \delta\boldsymbol{v} \cdot \nabla y + \boldsymbol{v} \cdot \nabla\delta y + a_0\delta y = 0 \quad \text{in} \quad Q, \\ \delta y = 0 \quad \text{on} \quad \Sigma, \\ \delta y(0) = 0. \end{cases} \tag{5.11}$$

Consider now a function p defined over $\overline{Q}$ (the closure of Q); and assume that p is a differentiable function of x and t. Multiplying both sides of the first

equation in (5.11) by p and integrating over Q, we obtain

$$\begin{aligned}&\iint_Q p\frac{\partial}{\partial t}\delta y dxdt - \nu\iint_Q p\nabla^2\delta y dxdt\\ &+\iint_Q \delta\boldsymbol{v}\cdot\nabla y p dxdt + \iint_Q \boldsymbol{v}\cdot\nabla\delta y p dxdt + a_0\iint_Q p\delta y dxdt = 0.\end{aligned}$$

Integration by parts in time and application of Green's formula in space yield

$$\begin{aligned}&\int_\Omega p(T)\delta y(T)dx - \int_\Omega p(0)\delta y(0)dx\\ &+\iint_Q\Big[-\frac{\partial p}{\partial t} - \nu\nabla^2 p - \boldsymbol{v}\cdot\nabla p + a_0 p\Big]\delta y dxdt + \iint_Q \delta\boldsymbol{v}\cdot\nabla y p dxdt\\ &-\nu\iint_\Sigma(\frac{\partial\delta y}{\partial\boldsymbol{n}}p - \frac{\partial p}{\partial\boldsymbol{n}}\delta y)dxdt + \iint_\Sigma p\delta y\boldsymbol{v}\cdot\boldsymbol{n}dxdt = 0.\end{aligned}\tag{5.12}$$

where $\boldsymbol{n}$ is the unit outward normal vector at Γ.

Next, let us take the L^2 adjoint of the operator $\frac{\partial}{\partial t} - \nu\nabla^2 + \boldsymbol{v}\cdot\nabla + a_0$ in (5.11) and consider the following adjoint system

$$\begin{cases} -\dfrac{\partial p}{\partial t} - \nu\nabla^2 p - \boldsymbol{v}\cdot\nabla p + a_0 p = \alpha_1(y - y_d) \text{ in } Q,\\ \qquad\qquad p = 0 \qquad\qquad \text{on } \Sigma,\\ \qquad\qquad p(T) = \alpha_2(y(T) - y_T). \end{cases}\tag{5.13}$$

In (5.13), the term $\alpha_1(y - y_d)$ is the derivative of the integrand of the objective functional with respect to the state y, and the terminal time term $\alpha_2(y(T) - y_T)$ comes from differentiating the objective functional integrand term at T. Assume that the function p is the solution to (5.13). Then, it follows from (5.10), (5.11), (5.12) and (5.13) that

$$\delta J(\boldsymbol{v}) = \iint_Q(\boldsymbol{v} - p\nabla y)\cdot\delta\boldsymbol{v}dxdt,$$

which, together with (5.9), implies that

$$\begin{cases} DJ(\boldsymbol{v})\in\mathcal{U},\\ \iint_Q DJ(\boldsymbol{v})\cdot\boldsymbol{z}dxdt = \iint_Q(\boldsymbol{v} - p\nabla y)\cdot\boldsymbol{z}dxdt, \forall\boldsymbol{z}\in\mathcal{U}. \end{cases}\tag{5.14}$$

From the discussion above, the first-order optimality condition of (BCP) can be summarized as follows.

Theorem 5.2. *Let $\boldsymbol{u} \in \mathcal{U}$ be an optimal control of (BCP). Then, it satisfies the following optimality condition*

$$\iint_Q (\boldsymbol{u} - p\nabla y) \cdot \boldsymbol{z} dxdt = 0, \forall \boldsymbol{z} \in \mathcal{U},$$

where y and p are obtained from $\boldsymbol{u}$ via the solutions of the following two parabolic equations:

$$\begin{cases} \dfrac{\partial y}{\partial t} - \nu\nabla^2 y + \boldsymbol{u} \cdot \nabla y + a_0 y = f \quad in\ Q, \\ y = g \quad on\ \Sigma, \\ y(0) = \phi, \end{cases} \qquad \text{(state equation)}$$

and

$$\begin{cases} -\dfrac{\partial p}{\partial t} - \nu\nabla^2 p - \boldsymbol{u} \cdot \nabla p + a_0 p = \alpha_1(y - y_d) \quad in\ Q, \\ p = 0 \quad on\ \Sigma, \\ p(T) = \alpha_2(y(T) - y_T). \end{cases} \qquad \text{(adjoint equation)}$$

5.3 An Implementable Nested Conjugate Gradient Method

In this section, we discuss the application of a CG strategy to solve (BCP). In particular, we elaborate on the computation of the gradient and the stepsize at each CG iteration, and thus obtain an easily implementable algorithm.

5.3.1 A Generic Conjugate Gradient Method for (BCP)

Conceptually, implementing the CG method to (BCP), we readily obtain the following algorithm:

(a) Given $\boldsymbol{u}^0 \in \mathcal{U}$.

(b) Compute $\boldsymbol{g}^0 = DJ(\boldsymbol{u}^0)$. If $DJ(\boldsymbol{u}^0) = 0$, take $\boldsymbol{u} = \boldsymbol{u}^0$; otherwise set $\boldsymbol{w}^0 = \boldsymbol{g}^0$.

For $k \geq 0$, $\boldsymbol{u}^k, \boldsymbol{g}^k$ and $\boldsymbol{w}^k$ being known, with the last two different from 0, one computes $\boldsymbol{u}^{k+1}, \boldsymbol{g}^{k+1}$ and if necessary $\boldsymbol{w}^{k+1}$ as follows:

(c) Compute the stepsize ρ_k by solving the following optimization problem

$$\begin{cases} \rho_k \in \mathbb{R}, \\ J(\boldsymbol{u}^k - \rho_k \boldsymbol{w}^k) \leq J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k), \forall \rho \in \mathbb{R}. \end{cases} \tag{5.15}$$

(d) Update $\boldsymbol{u}^{k+1}$ and $\boldsymbol{g}^{k+1}$, respectively, by

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - \rho_k \boldsymbol{w}^k,$$

and

$$\boldsymbol{g}^{k+1} = DJ(\boldsymbol{u}^{k+1}).$$

If $DJ(\boldsymbol{u}^{k+1}) = 0$, take $\boldsymbol{u} = \boldsymbol{u}^{k+1}$; otherwise,

(e) Compute

$$\beta_k = \frac{\iint_Q |\boldsymbol{g}^{k+1}|^2 dxdt}{\iint_Q |\boldsymbol{g}^k|^2 dxdt},$$

and then update

$$\boldsymbol{w}^{k+1} = \boldsymbol{g}^{k+1} + \beta_k \boldsymbol{w}^k.$$

Do $k+1 \to k$ and return to (**c**).

The above iterative method looks very simple, but practically, the implementation of the CG method (**a**)–(**e**) for the solution of (BCP) is nontrivial. In particular, it is numerically challenging to compute $DJ(\boldsymbol{v})$, $\forall \boldsymbol{v} \in \mathcal{U}$ and ρ_k as illustrated below. We shall discuss how to address these two issues in the following part of this section.

5.3.2 Computation of gradient

It is clear that the implementation of the generic CG method (**a**)–(**e**) for the solution of (BCP) requires the knowledge of $DJ(\boldsymbol{v})$ for various $\boldsymbol{v} \in \mathcal{U}$, and this has been conceptually provided in (5.14). However, it is numerically challenging to compute $DJ(\boldsymbol{v})$ by (5.14) due to the restriction $\nabla \cdot DJ(\boldsymbol{v}) = 0$ which ensures that all iterates $\boldsymbol{u}^k$ of the CG method meet the additional divergence-free

constraint $\nabla \cdot \boldsymbol{u}^k = 0$. In this subsection, we show that equation (5.14) can be reformulated as a saddle point problem by introducing a Lagrange multiplier associated with the constraint $\nabla \cdot DJ(\boldsymbol{v}) = 0$. Then, a preconditioned CG method is proposed to solve this saddle point problem.

First of all, it follows from [125] that equation (5.14) can be reformulated as

$$\begin{cases} DJ(\boldsymbol{v})(t) \in \mathbb{S}, \text{ for a.e. } t \in (0,T), \\ \int_\Omega DJ(\boldsymbol{v})(t) \cdot \boldsymbol{z} dx = \int_\Omega (\boldsymbol{v}(t) - p(t)\nabla y(t)) \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in \mathbb{S}, \end{cases} \tag{5.16}$$

where

$$\mathbb{S} = \{\boldsymbol{z} | \boldsymbol{z} \in [L^2(\Omega)]^d, \nabla \cdot \boldsymbol{z} = 0\}.$$

Clearly, problem (5.16) is a particular case of

$$\begin{cases} \boldsymbol{g} \in \mathbb{S}, \\ \int_\Omega \boldsymbol{g} \cdot \boldsymbol{z} dx = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in \mathbb{S}, \end{cases} \tag{5.17}$$

with $\boldsymbol{f}$ given in $[L^2(\Omega)]^d$.

Introducing a Lagrange multiplier $\lambda \in H_0^1(\Omega)$ associated with the constraint $\nabla \cdot \boldsymbol{z} = 0$ and then it is clear that problem (5.17) is equivalent to the following saddle point problem

$$\begin{cases} (\boldsymbol{g}, \lambda) \in [L^2(\Omega)]^d \times H_0^1(\Omega), \\ \int_\Omega \boldsymbol{g} \cdot \boldsymbol{z} dx = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{z} dx + \int_\Omega \lambda \nabla \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in [L^2(\Omega)]^d, \\ \int_\Omega \nabla \cdot \boldsymbol{g} q dx = 0, \forall q \in H_0^1(\Omega), \end{cases} \tag{5.18}$$

which is actually a Stokes type problem.

For the solution of (5.18), we advocate a CG method inspired from [75, 78]. For this purpose, one has to specify the inner product to be used over $H_0^1(\Omega)$. As discussed in [75], the usual L^2-inner product, namely, $\{q, q'\} \to \int_\Omega qq' dx$ leads to a CG method with poor convergence properties. Indeed, using some arguments similar to those in [74, 75], we can show that the saddle point problem (5.18) can be reformulated as a linear variational problem in terms of the Lagrange multiplier λ. The corresponding coefficient matrix after space discretization with

mesh size h has a condition number of the order of h^{-2}, which is ill-conditioned especially for small h and makes the CG method converges fairly slow. Hence, preconditioning is necessary for solving problem (5.18). Efficient preconditioning strategies include, e.g., the augmented Lagrangian approach in [75, Section 4] or the grad-div stabilization approach for computational fluid dynamics in [140]. Here, we follow [75, Section 7] and choose $-\nabla \cdot \nabla$ as a preconditioner for problem (5.18), and the corresponding preconditioned CG method operates in the space $H_0^1(\Omega)$ equipped with the inner product $\{q, q'\} \to \int_\Omega \nabla q \cdot \nabla q' dx$ and the associated norm $\|q\|_{H_0^1(\Omega)} = (\int_\Omega |\nabla q|^2 dx)^{1/2}, \forall q, q' \in H_0^1(\Omega)$. The resulting algorithm reads as:

G1 Choose $\lambda^0 \in H_0^1(\Omega)$.

G2 Solve

$$\begin{cases} \boldsymbol{g}^0 \in [L^2(\Omega)]^d, \\ \int_\Omega \boldsymbol{g}^0 \cdot \boldsymbol{z} dx = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{z} dx + \int_\Omega \lambda^0 \nabla \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in [L^2(\Omega)]^d, \end{cases}$$

and

$$\begin{cases} r^0 \in H_0^1(\Omega), \\ \int_\Omega \nabla r^0 \cdot \nabla q dx = \int_\Omega \nabla \cdot \boldsymbol{g}^0 q dx, \forall q \in H_0^1(\Omega). \end{cases}$$

If $\frac{\int_\Omega |\nabla r^0|^2 dx}{\max\{1, \int_\Omega |\nabla \lambda^0|^2 dx\}} \leq tol_1$, take $\lambda = \lambda^0$ and $\boldsymbol{g} = \boldsymbol{g}^0$; otherwise set $w^0 = r^0$. For $k \geq 0$, $\lambda^k, \boldsymbol{g}^k, r^k$ and w^k being known, with the last two different from 0, we compute $\lambda^{k+1}, \boldsymbol{g}^{k+1}, r^{k+1}$ and if necessary w^{k+1} as follows:

G3 Solve

$$\begin{cases} \bar{\boldsymbol{g}}^k \in [L^2(\Omega)]^d, \\ \int_\Omega \bar{\boldsymbol{g}}^k \cdot \boldsymbol{z} dx = \int_\Omega w^k \nabla \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in [L^2(\Omega)]^d, \end{cases}$$

and

$$\begin{cases} \bar{r}^k \in H_0^1(\Omega), \\ \int_\Omega \nabla \bar{r}^k \cdot \nabla q dx = \int_\Omega \nabla \cdot \bar{\boldsymbol{g}}^k q dx, \forall q \in H_0^1(\Omega), \end{cases}$$

and compute the stepsize via

$$\eta_k = \frac{\int_\Omega |\nabla r^k|^2 dx}{\int_\Omega \nabla \bar{r}^k \cdot \nabla w^k dx}.$$

G4 Update $\lambda^k, \boldsymbol{g}^k$ and r^k via

$$\lambda^{k+1} = \lambda^k - \eta_k w^k, \boldsymbol{g}^{k+1} = \boldsymbol{g}^k - \eta_k \bar{\boldsymbol{g}}^k, \text{ and } r^{k+1} = r^k - \eta_k \bar{r}^k.$$

If $\frac{\int_\Omega |\nabla r^{k+1}|^2 dx}{\max\{1, \int_\Omega |\nabla r^0|^2 dx\}} \leq tol_1$, take $\lambda = \lambda^{k+1}$ and $\boldsymbol{g} = \boldsymbol{g}^{k+1}$; otherwise,

G5 Compute

$$\gamma_k = \frac{\int_\Omega |\nabla r^{k+1}|^2 dx}{\int_\Omega |\nabla r^k|^2 dx},$$

and update w^k via

$$w^{k+1} = r^{k+1} + \gamma_k w^k.$$

Do $k+1 \to k$ and return to **G3**.

Clearly, one only needs to solve two simple linear equations at each iteration of the preconditioned CG algorithm (**G1**)–(**G5**), which implies that the algorithm is easy and cheap to implement. Moreover, due to the well-chosen preconditioner $-\nabla \cdot \nabla$, one can expect the above preconditioned CG algorithm to have a fast convergence; this will be validated by the numerical experiments reported in Section 5.5.

5.3.3 Computation of the Stepsize ρ_k

Another crucial step to implement the CG method **(a)**–**(e)** is the computation of the stepsize ρ_k. It is the solution of the optimization problem (5.15) which is numerically expensive to be solved exactly or up to a high accuracy. For instance, to solve (5.15), one may consider the Newton method applied to the solution of

$$H_k'(\rho) = 0,$$

where

$$H_k(\rho) = J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k).$$

The Newton method requires the second-order derivative $H_k''(\rho)$ which can be computed via an iterated adjoint technique requiring the solution of *four* parabolic problems per Newton's iteration. Hence, the implementation of the Newton method is numerically expensive.

The high computational load for solving (5.15) motivates us to implement certain stepsize rule to determine an approximation of ρ_k. Here, we advocate the following procedure to compute an approximate stepsize $\hat{\rho}_k$.

For a given $\boldsymbol{w}^k \in \mathcal{U}$, we replace the state $y = S(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ in $J(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ by

$$S(\boldsymbol{u}^k) - \rho S'(\boldsymbol{u}^k)\boldsymbol{w}^k,$$

which is indeed the linearization of the mapping $\rho \mapsto S(\boldsymbol{u}^k - \rho \boldsymbol{w}^k)$ at $\rho = 0$. We thus obtain the following quadratic approximation of $H_k(\rho)$:

$$\begin{aligned} Q_k(\rho) := \frac{1}{2}\iint_Q |\boldsymbol{u}^k - \rho \boldsymbol{w}^k|^2 dxdt + \frac{\alpha_1}{2}\iint_Q |y^k - \rho z^k - y_d|^2 dxdt \\ + \frac{\alpha_2}{2}\int_\Omega |y^k(T) - \rho z^k(T) - y_T|^2 dx, \end{aligned} \tag{5.19}$$

where $y^k = S(\boldsymbol{u}^k)$ is the solution of the state equation (1.15) associated with $\boldsymbol{u}^k$, and $z^k = S'(\boldsymbol{u}^k)\boldsymbol{w}^k$ satisfies the following linear parabolic problem

$$\begin{cases} \dfrac{\partial z^k}{\partial t} - \nu \nabla^2 z^k + \boldsymbol{w}^k \cdot \nabla y^k + \boldsymbol{u}^k \cdot \nabla z^k + a_0 z^k = 0 \quad \text{in} \quad Q, \\ z^k = 0 \quad \text{on} \quad \Sigma, \\ z^k(0) = 0. \end{cases} \tag{5.20}$$

Then, it is easy to show that the equation $Q_k'(\rho) = 0$ admits a unique solution

$$\hat{\rho}_k = \frac{\iint_Q \boldsymbol{g}^k \cdot \boldsymbol{w}^k dxdt}{\iint_Q |\boldsymbol{w}^k|^2 dxdt + \alpha_1 \iint_Q |z^k|^2 dxdt + \alpha_2 \int_\Omega |z^k(T)|^2 dx}, \tag{5.21}$$

and we take $\hat{\rho}_k$, which is clearly an approximation of ρ_k, as the stepsize in each CG iteration.

Altogether, with the stepsize given by (5.21), every iteration of the resulting CG algorithm requires solving only *three* parabolic problems, namely, the state equation (1.15) forward in time and the associated adjoint equation (5.13) backward in time for the computation of $\boldsymbol{g}^k$, and the linearized parabolic equation (5.20) forward in time for the stepsize $\hat{\rho}_k$. For comparison, if the Newton method is employed to compute the stepsize ρ_k by solving (5.15), at least *six* parabolic problems are required to be solved at each iteration of the CG method, which is much more expensive numerically.

Remark 5.1. *To find an appropriate stepsize, a natural idea is to employ some line search strategies, such as the backtracking strategy based on the Armijo–Goldstein condition or the Wolf condition, see e.g., [138]. It is worth noting that these line search strategies require the evaluation of $J(\boldsymbol{v})$ repeatedly, which is numerically expensive because every evaluation of $J(\boldsymbol{v})$ for a given $\boldsymbol{v}$ requires solving the state equation (1.15). Moreover, we have implemented the CG method for solving (BCP) with various line search strategies and observed from the numerical results that line search strategies always lead to tiny stepsizes making extremely slow the convergence of the CG method.*

5.3.4 A Nested CG Method for Solving (BCP)

Following Sections 5.3.2 and 5.3.3, we advocate the following nested CG method for solving (BCP):

I. Given $\boldsymbol{u}^0 \in \mathcal{U}$.

II. Compute y^0 and p^0 by solving the state equation (1.15) and the adjoint equation (5.13) corresponding to $\boldsymbol{u}^0$. Then, for a.e. $t \in (0,T)$, solve

$$\begin{cases} \boldsymbol{g}^0(t) \in \mathbb{S}, \\ \displaystyle\int_\Omega \boldsymbol{g}^0(t)\cdot\boldsymbol{z}dx = \int_\Omega (\boldsymbol{u}^0(t) - p^0(t)\nabla y^0(t))\cdot\boldsymbol{z}dx, \forall \boldsymbol{z}\in\mathbb{S}, \end{cases}$$

by the preconditioned CG algorithm (**G1**)–(**G5**); and set $\boldsymbol{w}^0 = \boldsymbol{g}^0$.

For $k \geq 0$, $\boldsymbol{u}^k, \boldsymbol{g}^k$ and $\boldsymbol{w}^k$ being known, with the last two different from 0, one computes $\boldsymbol{u}^{k+1}, \boldsymbol{g}^{k+1}$ and if necessary $\boldsymbol{w}^{k+1}$ as follows:

III. Compute the stepsize $\hat{\rho}_k$ by (5.21).

IV. Update $\boldsymbol{u}^{k+1}$ by

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - \hat{\rho}_k \boldsymbol{w}^k.$$

Compute y^{k+1} and p^{k+1} by solving the state equation (1.15) and the adjoint equation (5.13) corresponding to $\boldsymbol{u}^{k+1}$; and for a.e. $t \in (0,T)$, solve

$$\begin{cases} \boldsymbol{g}^{k+1}(t) \in \mathbb{S}, \\ \displaystyle\int_\Omega \boldsymbol{g}^{k+1}(t)\cdot\boldsymbol{z}dx = \int_\Omega (\boldsymbol{u}^{k+1}(t) - p^{k+1}(t)\nabla y^{k+1}(t))\cdot\boldsymbol{z}dx, \forall \boldsymbol{z}\in\mathbb{S}, \end{cases}$$

by the preconditioned CG algorithm (**G1**)–(**G5**).

If $\frac{\iint_Q |\boldsymbol{g}^{k+1}|^2 dxdt}{\iint_Q |\boldsymbol{g}^0|^2 dxdt} \leq tol$, take $\boldsymbol{u} = \boldsymbol{u}^{k+1}$; else

V. Compute

$$\beta_k = \frac{\iint_Q |\boldsymbol{g}^{k+1}|^2 dxdt}{\iint_Q |\boldsymbol{g}^k|^2 dxdt}, \text{ and } \boldsymbol{w}^{k+1} = \boldsymbol{g}^{k+1} + \beta_k \boldsymbol{w}^k.$$

Do $k+1 \to k$ and return to **III**.

5.4 Space and time discretizations

In this section, we discuss first the numerical discretization of the bilinear optimal control problem (BCP). We achieve the time discretization by a semi-implicit finite difference method and the space discretization by a piecewise linear finite element method. Then, we discuss an implementable nested CG method for solving the fully discrete bilinear optimal control problem.

5.4.1 Time Discretization of (BCP)

First, we define a time discretization step Δt by $\Delta t = T/N$, with N a positive integer. Then, we approximate the control space $\mathcal{U} = L^2(0,T;\mathbb{S})$ by $\mathcal{U}^{\Delta t} := (\mathbb{S})^N$; and equip $\mathcal{U}^{\Delta t}$ with the following inner product

$$(\boldsymbol{v}, \boldsymbol{w})_{\Delta t} = \Delta t \sum_{n=1}^{N} \int_\Omega \boldsymbol{v}_n \cdot \boldsymbol{w}_n dx, \quad \forall \boldsymbol{v} = \{\boldsymbol{v}_n\}_{n=1}^N, \boldsymbol{w} = \{\boldsymbol{w}_n\}_{n=1}^N \in \mathcal{U}^{\Delta t},$$

and the norm

$$\|\boldsymbol{v}\|_{\Delta t} = \left(\Delta t \sum_{n=1}^{N} \int_\Omega |\boldsymbol{v}_n|^2 dx \right)^{\frac{1}{2}}, \quad \forall \boldsymbol{v} = \{\boldsymbol{v}_n\}_{n=1}^N \in \mathcal{U}^{\Delta t}.$$

Then, (BCP) is approximated by the following semi-discrete bilinear control problem (BCP$^{\Delta t}$):

$$\begin{cases} \boldsymbol{u}^{\Delta t} \in \mathcal{U}^{\Delta t}, \\ J^{\Delta t}(\boldsymbol{u}^{\Delta t}) \leq J^{\Delta t}(\boldsymbol{v}), \forall \boldsymbol{v} = \{\boldsymbol{v}_n\}_{n=1}^N \in \mathcal{U}^{\Delta t}, \end{cases} \tag{BCP$^{\Delta t}$}$$

where the objective functional $J^{\Delta t}$ is defined by

$$J^{\Delta t}(\boldsymbol{v}) = \frac{1}{2}\Delta t \sum_{n=1}^{N} \int_{\Omega} |\boldsymbol{v}_n|^2 dx + \frac{\alpha_1}{2}\Delta t \sum_{n=1}^{N} \int_{\Omega} |y_n - y_d^n|^2 dx + \frac{\alpha_2}{2} \int_{\Omega} |y_N - y_T|^2 dx,$$

with $\{y_n\}_{n=1}^N$ the solution of the following semi-discrete state equation: $y_0 = \phi$; for $n = 1, \ldots, N$, with y_{n-1} being known, we obtain y_n from the solution of the following linear elliptic problem:

$$\begin{cases} \dfrac{y_n - y_{n-1}}{\Delta t} - \nu \nabla^2 y_n + \boldsymbol{v}_n \cdot \nabla y_{n-1} + a_0 y_{n-1} = f_n \quad \text{in} \quad \Omega, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad y_n = g_n \quad \text{on} \quad \Gamma. \end{cases} \tag{5.22}$$

Remark 5.2. *For simplicity, we have chosen a one-step semi-implicit scheme to discretize system (1.15). This scheme is first-order accurate and reasonably robust, once combined to an appropriate space discretization. The application of second-order accurate time discretization schemes to optimal control problems has been discussed in e.g., [29].*

Remark 5.3. *At each step of scheme (5.22), we only need to solve a simple linear elliptic problem to obtain y_n from y_{n-1}, and there is no particular difficulty in solving such a problem.*

Remark 5.4. *Note that usually a semi-implicit scheme is only conditionally stable and its restriction on the time stepsize Δt is problem-dependent. For the diffusion-dominated case, we will show in Section 5.5 that the semi-implicit scheme to be implemented works well empirically even with a relatively large time stepsize. For the advection-dominated case, however, as well studied in the literatures (e.g. [3, 12]), more restrictive conditions on the time stepsize are usually required.*

The existence of an optimal control to (BCP$^{\Delta t}$) can be proved in a similar way as what we have done for the continuous case. Let $\boldsymbol{u}^{\Delta t}$ be an optimal control of (BCP$^{\Delta t}$) and then it verifies the following first-order optimality condition:

$$DJ^{\Delta t}(\boldsymbol{u}^{\Delta t}) = 0,$$

where $DJ^{\Delta t}(\boldsymbol{v})$ is the first-order differential of the functional $J^{\Delta t}$ at $\boldsymbol{v} \in \mathcal{U}^{\Delta t}$.

Proceeding as in the continuous case, we can show that $DJ^{\Delta t}(\boldsymbol{v}) = \{\boldsymbol{g}_n\}_{n=1}^N \in \mathcal{U}^{\Delta t}$ where

$$\begin{cases} \boldsymbol{g}_n \in \mathbb{S}, \\ \int_\Omega \boldsymbol{g}_n \cdot \boldsymbol{w} dx = \int_\Omega (\boldsymbol{v}_n - p_n \nabla y_{n-1}) \cdot \boldsymbol{w} dx, \forall \boldsymbol{w} \in \mathbb{S}, \end{cases}$$

and the vector-valued function $\{p_n\}_{n=1}^N$ is the solution of the semi-discrete adjoint system below:

$$p_{N+1} = \alpha_2(y_N - y_T);$$

for $n = N$, solve

$$\begin{cases} \dfrac{p_N - p_{N+1}}{\Delta t} - \nu \nabla^2 p_N = \alpha_1(y_N - y_d^N) & \text{in} \quad \Omega, \\ p_N = 0 & \text{on} \quad \Gamma, \end{cases}$$

and for $n = N-1, \cdots, 1$, solve

$$\begin{cases} \dfrac{p_n - p_{n+1}}{\Delta t} - \nu \nabla^2 p_n - \boldsymbol{v}_{n+1} \cdot \nabla p_{n+1} + a_0 p_{n+1} = \alpha_1(y_n - y_d^n) & \text{in} \quad \Omega, \\ p_n = 0 & \text{on} \quad \Gamma. \end{cases}$$

5.4.2 Space Discretization of (BCP$^{\Delta t}$)

In this subsection, we discuss the space discretization of (BCP$^{\Delta t}$), obtaining thus a full space-time discretization of (BCP). For simplicity, we suppose from now on that Ω is a polygonal domain of $\mathbb{R}^2$ (or has been approximated by a family of such domains).

Let $\mathcal{T}_H$ be a classical triangulation of Ω, with H the largest length of the edges of the triangles of $\mathcal{T}_H$. From $\mathcal{T}_H$ we construct $\mathcal{T}_h$ with $h = H/2$ by joining the mid-points of the edges of the triangles of $\mathcal{T}_H$.

We first consider the finite element space V_h defined by

$$V_h = \{\varphi_h | \varphi_h \in C^0(\bar{\Omega}); \varphi_h|_{\mathbb{T}} \in P_1, \forall\, \mathbb{T} \in \mathcal{T}_h\}$$

with P_1 the space of the polynomials of two variables of degree ≤ 1. Two useful sub-spaces of V_h are

$$V_{0h} = \{\varphi_h | \varphi_h \in V_h, \varphi_h|_\Gamma = 0\} := V_h \cap H_0^1(\Omega),$$

and (assuming that $g(t) \in C^0(\Gamma)$)

$$V_{gh}(t) = \{\varphi_h | \varphi_h \in V_h, \varphi_h(Q) = g(Q,t), \forall\, Q \text{ vertex of } \mathcal{T}_h \text{ located on } \Gamma\}.$$

In order to construct the discrete control space, we introduce first

$$\Lambda_H = \{\varphi_H | \varphi_H \in C^0(\bar{\Omega}); \varphi_H|_{\mathbb{T}} \in P_1, \forall\, \mathbb{T} \in \mathcal{T}_H\}, \text{ and } \Lambda_{0H} = \{\varphi_H | \varphi_H \in \Lambda_H, \varphi_H|_\Gamma = 0\}.$$

Then, the discrete control space $\mathcal{U}_h^{\Delta t}$ is defined by

$$\mathcal{U}_h^{\Delta t} = (\mathbb{S}_h)^N,$$

with

$$\mathbb{S}_h = \{\boldsymbol{v}_h | \boldsymbol{v}_h \in V_h \times V_h, \int_\Omega \nabla \cdot \boldsymbol{v}_h q_H dx \left(= -\int_\Omega \boldsymbol{v}_h \cdot \nabla q_H dx \right) = 0, \forall q_H \in \Lambda_{0H}\}.$$

With the above finite element spaces, we approximate (BCP) and (BCP$^{\Delta t}$) by (BCP$_h^{\Delta t}$)defined by

$$\begin{cases} \boldsymbol{u}_h^{\Delta t} \in \mathcal{U}_h^{\Delta t}, \\ J_h^{\Delta t}(\boldsymbol{u}_h^{\Delta t}) \leq J_h^{\Delta t}(\boldsymbol{v}_h^{\Delta t}), \forall \boldsymbol{v}_h^{\Delta t} = \{\boldsymbol{v}_{n,h}\}_{n=1}^N \in \mathcal{U}_h^{\Delta t}, \end{cases} \tag{BCP$_h^{\Delta t}$}$$

where the fully discrete objective functional $J_h^{\Delta t}$ is defined by

$$J_h^{\Delta t}(\boldsymbol{v}_h^{\Delta t}) = \frac{1}{2}\Delta t \sum_{n=1}^N \int_\Omega |\boldsymbol{v}_{n,h}|^2 dx + \frac{\alpha_1}{2}\Delta t \sum_{n=1}^N \int_\Omega |y_{n,h} - y_d^n|^2 dx + \frac{\alpha_2}{2}\int_\Omega |y_{N,h} - y_T|^2 dx \tag{5.23}$$

with $\{y_{n,h}\}_{n=1}^N$ the solution of the following fully discrete state equation: $y_{0,h} = \phi_h \in V_h$, where ϕ_h verifies

$$\phi_h \in V_h, \forall\, h > 0, \text{ and } \lim_{h \to 0} \phi_h = \phi, \text{ in } L^2(\Omega),$$

and for $n = 1, \ldots, N$, with $y_{n-1,h}$ being known, we obtain $y_{n,h} \in V_{gh}(n\Delta t)$ from the solution of the following linear variational problem:

$$\begin{aligned} \int_\Omega \frac{y_{n,h} - y_{n-1,h}}{\Delta t}\varphi dx + \nu \int_\Omega \nabla y_{n,h} \cdot \nabla \varphi dx + \int_\Omega \boldsymbol{v}_{n,h} \cdot \nabla y_{n-1,h} \varphi dx \\ + \int_\Omega a_0 y_{n-1,h} \varphi dx = \int_\Omega f_n \varphi dx, \forall \varphi \in V_{0h}. \end{aligned} \tag{5.24}$$

In the following discussion, the subscript h in all variables will be omitted for simplicity.

In a similar way as what we have done in the continuous case, one can show that the first-order differential of $J_h^{\Delta t}$ at $\boldsymbol{v} \in \mathcal{U}_h^{\Delta t}$ is $DJ_h^{\Delta t}(\boldsymbol{v}) = \{\boldsymbol{g}_n\}_{n=1}^N \in (\mathbb{S}_h)^N$ where

$$\begin{cases} \boldsymbol{g}_n \in \mathbb{S}_h, \\ \displaystyle\int_\Omega \boldsymbol{g}_n \cdot \boldsymbol{z} dx = \int_\Omega (\boldsymbol{v}_n - p_n \nabla y_{n-1}) \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in \mathbb{S}_h, \end{cases} \tag{5.25}$$

and the vector-valued function $\{p_n\}_{n=1}^N$ is the solution of the following fully discrete adjoint system:

$$p_{N+1} = \alpha_2(y_N - y_T); \tag{5.26}$$

for $n = N$, solve

$$\begin{cases} p_N \in V_{0h}, \\ \displaystyle\int_\Omega \frac{p_N - p_{N+1}}{\Delta t} \varphi dx + \nu \int_\Omega \nabla p_N \cdot \nabla \varphi dx = \int_\Omega \alpha_1 (y_N - y_d^N) \varphi dx, \forall \varphi \in V_{0h}, \end{cases} \tag{5.27}$$

then, for $n = N-1, \cdots, 1$, solve

$$\begin{cases} p_n \in V_{0h}, \\ \displaystyle\int_\Omega \frac{p_n - p_{n+1}}{\Delta t} \varphi dx + \nu \int_\Omega \nabla p_n \cdot \nabla \varphi dx + \int_\Omega p_{n+1} \boldsymbol{v}_{n+1} \cdot \nabla \varphi dx \\ \qquad\qquad + a_0 \displaystyle\int_\Omega p_{n+1} \varphi dx = \int_\Omega \alpha_1 (y_n - y_d^n) \varphi dx, \forall \varphi \in V_{0h}. \end{cases} \tag{5.28}$$

It is worth mentioning that the so-called discretize-then-optimize approach is employed here, which implies that we first discretize (BCP), and to compute the gradient in a discrete setting, the fully discrete adjoint equation (5.26)–(5.28) has been derived from the fully discrete objective functional $J_h^{\Delta t}$ (5.23) and the fully discrete state equation (5.24). This implies that the fully discrete state equation (5.24) and the fully discrete adjoint equation (5.26)–(5.28) are strictly in duality. This fact guarantees that $-DJ_h^{\Delta t}(\boldsymbol{v})$ is a descent direction of the fully discrete bilinear optimal control problem ($\mathrm{BCP}_h^{\Delta t}$).

Remark 5.5. *A natural alternative has been advocated in the literature: (i) Derive the adjoint equation to compute the first-order differential of the objective functional in a continuous setting; (ii) Discretize the state and adjoint state equations by certain numerical schemes; (iii) Use the resulting discrete analogs of y and p to compute a discretization of the differential of the objective functional.*

The main problem with this optimize-then-discretize approach is that it may not preserve a strict duality between the discrete state equation and the discrete adjoint equation. This fact implies in turn that the resulting discretization of the continuous gradient may not be a gradient of a discrete optimal control problem. As a consequence, the resulting algorithm is not a descent algorithm and divergence may take place. We refer to, e.g. [79, 89, 197], for more discussions on the difference between the discretize-then-optimize and optimize-then-discretize approaches.

5.4.3 A Nested CG Method for Solving the Fully Discrete Problem ($\mathbf{BCP}_h^{\Delta t}$)

In this subsection, we propose a nested CG method for solving the fully discrete problem ($\mathrm{BCP}_h^{\Delta t}$). As discussed in Section 5.3, the implementation of CG method requires the knowledge of $DJ_h^{\Delta t}(\boldsymbol{v})$ and an appropriate stepsize. In the following discussion, we address these two issues by extending the results for the continuous case in Sections 5.3.2 and 5.3.3 to the fully discrete settings; and derive the corresponding CG algorithm.

First, it is clear that one can compute $DJ_h^{\Delta t}(\boldsymbol{v})$ via the solution of the N linear variational problems encountered in (5.25). For this purpose, we introduce a Lagrange multiplier $\lambda \in \Lambda_{0H}$ associated with the divergence-free constraint. Then, problem (5.25) is equivalent to the following saddle point system

$$\begin{cases} (\boldsymbol{g}_n, \lambda) \in (V_h \times V_h) \times \Lambda_{0H}, \\ \displaystyle\int_\Omega \boldsymbol{g}_n \cdot \boldsymbol{z} dx = \int_\Omega (\boldsymbol{v}_n - p_n \nabla y_{n-1}) \cdot \boldsymbol{z} dx + \int_\Omega \lambda \nabla \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in V_h \times V_h, \\ \displaystyle\int_\Omega \nabla \cdot \boldsymbol{g}_n q dx = 0, \forall q \in \Lambda_{0H}. \end{cases} \tag{5.29}$$

As discussed in Section 5.3.2, problem (5.29) can be solved by the following preconditioned CG algorithm, which is actually a discrete analogue of (**G1**)–(**G5**).

DG1 Choose $\lambda^0 \in \Lambda_{0H}$.

DG2 Solve

$$\begin{cases} \boldsymbol{g}_n^0 \in V_h \times V_h, \\ \int_\Omega \boldsymbol{g}_n^0 \cdot \boldsymbol{z} dx = \int_\Omega (\boldsymbol{v}_n - p_n \nabla y_{n-1}) \cdot \boldsymbol{z} dx + \int_\Omega \lambda^0 \nabla \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in V_h \times V_h, \end{cases}$$

and

$$\begin{cases} r^0 \in \Lambda_{0H}, \\ \int_\Omega \nabla r^0 \cdot \nabla q dx = \int_\Omega \nabla \cdot \boldsymbol{g}_n^0 q dx, \forall q \in \Lambda_{0H}. \end{cases}$$

If $\frac{\int_\Omega |\nabla r^0|^2 dx}{\max\{1, \int_\Omega |\nabla \lambda^0|^2 dx\}} \leq tol_1$, take $\lambda = \lambda^0$ and $\boldsymbol{g}_n = \boldsymbol{g}_n^0$; otherwise set $w^0 = r^0$. For $k \geq 0$, $\lambda^k, \boldsymbol{g}_n^k, r^k$ and w^k being known, with the last two different from 0, we compute $\lambda^{k+1}, \boldsymbol{g}_n^{k+1}, r^{k+1}$ and if necessary w^{k+1} as follows:

DG3 Solve

$$\begin{cases} \bar{\boldsymbol{g}}_n^k \in V_h \times V_h, \\ \int_\Omega \bar{\boldsymbol{g}}_n^k \cdot \boldsymbol{z} dx = \int_\Omega w^k \nabla \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in V_h \times V_h, \end{cases}$$

and

$$\begin{cases} \bar{r}^k \in \Lambda_{0H}, \\ \int_\Omega \nabla \bar{r}^k \cdot \nabla q dx = \int_\Omega \nabla \cdot \bar{\boldsymbol{g}}_n^k q dx, \forall q \in \Lambda_{0H}, \end{cases}$$

and compute

$$\eta_k = \frac{\int_\Omega |\nabla r^k|^2 dx}{\int_\Omega \nabla \bar{r}^k \cdot \nabla w^k dx}.$$

DG4 Update $\lambda^k, \boldsymbol{g}_n^k$ and r^k via

$$\lambda^{k+1} = \lambda^k - \eta_k w^k, \boldsymbol{g}_n^{k+1} = \boldsymbol{g}_n^k - \eta_k \bar{\boldsymbol{g}}_n^k, \text{ and } r^{k+1} = r^k - \eta_k \bar{r}^k.$$

If $\frac{\int_\Omega |\nabla r^{k+1}|^2 dx}{\max\{1, \int_\Omega |\nabla r^0|^2 dx\}} \leq tol_1$, take $\lambda = \lambda^{k+1}$ and $\boldsymbol{g}_n = \boldsymbol{g}_n^{k+1}$; otherwise,

DG5 Compute

$$\gamma_k = \frac{\int_\Omega |\nabla r^{k+1}|^2 dx}{\int_\Omega |\nabla r^k|^2 dx},$$

and update w^k via

$$w^{k+1} = r^{k+1} + \gamma_k w^k.$$

Do $k+1 \to k$ and return to **DG3**.

Remark 5.6. *Similar as the continuous case discussed in Section 3, we employ the preconditioned CG algorithm (**DG1**)–(**DG5**) to compute the gradient $DJ_h^{\Delta t}(\boldsymbol{v})$ in order that all iterates of the CG method satisfy the divergence-free constraint. To ensure the divergence-free constraint, other choices include some exactly divergence-free finite element methods for space discretization such as the H_0^1-conforming finite element methods based on the Scott–Vogelius elements [88], some $H(div)$-conforming finite element methods based on the Raviart–Thomas elements [151] or the Brezzi–Douglas–Marini elements [23].*

To find an appropriate stepsize in the CG iteration for the solution of $(\mathrm{BCP}_h^{\Delta t})$, we note that, for any $\{\boldsymbol{w}_n\}_{n=1}^N \in (\mathbb{S}_h)^N$, the fully discrete analogue of $Q_k(\rho)$ in (5.19) reads as

$$\begin{aligned} Q_h^{\Delta t}(\rho) =& \frac{1}{2}\Delta t \sum_{n=1}^{N} \int_\Omega |\boldsymbol{u}_n - \rho \boldsymbol{w}_n|^2 dx \\ &+ \frac{\alpha_1}{2}\Delta t \sum_{n=1}^{N} \int_\Omega |y_n - \rho z_n - y_d^n|^2 dx + \frac{\alpha_2}{2}\int_\Omega |y_N - \rho z_N - y_T|^2 dx, \end{aligned}$$

where the vector-valued function $\{z_n\}_{n=1}^N$ is obtained as follows: $z_0 = 0$; for $n = 1, \ldots, N$, with z_{n-1} being known, z_n is obtained from the solution of the linear variational problem

$$\begin{cases} z_n \in V_{0h}, \\ \displaystyle\int_\Omega \frac{z_n - z_{n-1}}{\Delta t}\varphi dx + \nu \int_\Omega \nabla z_n \cdot \nabla \varphi dx + \int_\Omega \boldsymbol{w}_n \cdot \nabla y_n \varphi dx \\ \qquad\qquad + \displaystyle\int_\Omega \boldsymbol{u}_n \cdot \nabla z_{n-1}\varphi dx + a_0 \int_\Omega z_{n-1}\varphi dx = 0, \forall \varphi \in V_{0h}. \end{cases}$$

As discussed in Section 5.3.3 for the continuous case, we take the unique solution of $Q_h^{\Delta t'}(\rho) = 0$ as the stepsize in each CG iteration, that is

$$\hat{\rho}_h^{\Delta t} = \frac{\Delta t \sum_{n=1}^N \int_\Omega \boldsymbol{g}_n \cdot \boldsymbol{w}_n dx}{\Delta t \sum_{n=1}^N \int_\Omega |\boldsymbol{w}_n|^2 dx + \alpha_1 \Delta t \sum_{n=1}^N \int_\Omega |z_n|^2 dx + \alpha_2 \int_\Omega |z_N|^2 dx}. \tag{5.30}$$

Finally, with above preparations, we propose the following nested CG method for the solution of the fully discrete problem $(\mathrm{BCP}_h^{\Delta t})$.

DI. Given $\boldsymbol{u}^0 := \{\boldsymbol{u}_n^0\}_{n=1}^N \in (\mathbb{S}_h)^N$.

DII. Compute $\{y_n^0\}_{n=0}^N$ and $\{p_n^0\}_{n=1}^{N+1}$ by solving the fully discrete state equation (5.24) and the fully discrete adjoint equation (5.26)–(5.28) corresponding to $\boldsymbol{u}^0$. Then, for $n = 1, \cdots, N$ solve

$$\begin{cases} \boldsymbol{g}_n^0 \in \mathbb{S}_h, \\ \int_\Omega \boldsymbol{g}_n^0 \cdot \boldsymbol{z} dx = \int_\Omega (\boldsymbol{u}_n^0 - p_n^0 \nabla y_{n-1}^0) \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in \mathbb{S}_h, \end{cases}$$

by the preconditioned CG algorithm (**DG1**)–(**DG5**), and set $\boldsymbol{w}_n^0 = \boldsymbol{g}_n^0$.

For $k \geq 0$, $\boldsymbol{u}^k, \boldsymbol{g}^k$ and $\boldsymbol{w}^k$ being known, with the last two different from $\mathbf{0}$, one computes $\boldsymbol{u}^{k+1}, \boldsymbol{g}^{k+1}$ and if necessary $\boldsymbol{w}^{k+1}$ as follows:

DIII. Compute the stepsize $\hat{\rho}_k$ by (5.30).

DIV. Update $\boldsymbol{u}^{k+1}$ by

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k - \hat{\rho}_k \boldsymbol{w}^k.$$

Compute $\{y_n^{k+1}\}_{n=0}^N$ and $\{p_n^{k+1}\}_{n=1}^{N+1}$ by solving the fully discrete state equation (5.24) and the fully discrete adjoint equation (5.26)–(5.28) corresponding to $\boldsymbol{u}^{k+1}$. Then, for $n = 1, \cdots, N$, solve

$$\begin{cases} \boldsymbol{g}_n^{k+1} \in \mathbb{S}_h, \\ \int_\Omega \boldsymbol{g}_n^{k+1} \cdot \boldsymbol{z} dx = \int_\Omega (\boldsymbol{u}_n^{k+1} - p_n^{k+1} \nabla y_{n-1}^{k+1}) \cdot \boldsymbol{z} dx, \forall \boldsymbol{z} \in \mathbb{S}_h, \end{cases} \tag{5.31}$$

by the preconditioned CG algorithm (**DG1**)–(**DG5**).

If $\frac{\Delta t \sum_{n=1}^N \int_\Omega |\boldsymbol{g}_n^{k+1}|^2 dx}{\Delta t \sum_{n=1}^N \int_\Omega |\boldsymbol{g}_n^0|^2 dx} \leq tol$, take $\boldsymbol{u} = \boldsymbol{u}^{k+1}$; else

DV. Compute

$$\beta_k = \frac{\Delta t \sum_{n=1}^N \int_\Omega |\boldsymbol{g}_n^{k+1}|^2 dx}{\Delta t \sum_{n=1}^N \int_\Omega |\boldsymbol{g}_n^k|^2 dx}, \text{ and } \boldsymbol{w}^{k+1} = \boldsymbol{g}^{k+1} + \beta_k \boldsymbol{w}^k.$$

Do $k + 1 \to k$ and return to **DIII**.

Despite its apparent complexity, the nested CG method (**DI**)–(**DV**) is easy to implement. Actually, one of the main computational difficulties in the implementation of the above algorithm seems to be the solution of N linear systems (5.31), which is time-consuming. However, it is worth noting that the linear systems (5.31) are separable with respect to different n and they can be solved in

parallel. As a consequent, one can compute the gradient $\{\boldsymbol{g}_n^k\}_{n=1}^N$ simultaneously and the computation time can be reduced significantly.

Moreover, it is clear that the computation of $\{\boldsymbol{g}_n^k\}_{n=1}^N$ requires the storage of the solutions of (5.24) and (5.26)–(5.28) at all points in space and time. For large scale problems, especially in three space dimensions, it will be very memory demanding and maybe even impossible to store the full sets $\{y_n^k\}_{n=0}^N$ and $\{p_n^k\}_{n=1}^{N+1}$ simultaneously. To tackle this issue, one can employ the strategy described in e.g., [83, Section 1.12] that can drastically reduce the storage requirements at the expense of a small CPU increase.

5.5 Numerical experiments

In this section, we report some preliminary numerical results validating the efficiency of the proposed CG algorithm (**DI**)–(**DV**) for (BCP). All codes were written in MATLAB R2016b and numerical experiments were conducted on a Surface Pro 5 laptop with 64-bit Windows 10.0 operation system, Intel(R) Core(TM) i7-7660U CPU (2.50 GHz), and 16 GB RAM.

Example 1. We consider the bilinear optimal control problem (BCP) on the domain $Q = \Omega \times (0, T)$ with $\Omega = (0, 1)^2$ and $T = 1$. In particular, we take the control $\boldsymbol{v}(x, t)$ in a finite-dimensional space, i.e. $\boldsymbol{v} \in L^2(0, T; \mathbb{R}^2)$. In addition, we set $\alpha_2 = 0$ in (5.1) and consider the following tracking-type bilinear optimal control problem:

$$\min_{\boldsymbol{v} \in L^2(0,T;\mathbb{R}^2)} J(\boldsymbol{v}) = \frac{1}{2} \int_0^T |\boldsymbol{v}(t)|^2 dt + \frac{\alpha_1}{2} \iint_Q |y - y_d|^2 dx dt, \tag{5.32}$$

where $|\boldsymbol{v}(t)| = \sqrt{\boldsymbol{v}_1(t)^2 + \boldsymbol{v}_2(t)^2}$ is the canonical 2-norm, and y is obtained from $\boldsymbol{v}$ via the solution of the state equation (5.2).

Since the control $\boldsymbol{v}$ is considered in a finite-dimensional space, the divergence-free constraint $\nabla \cdot \boldsymbol{v} = 0$ is verified automatically. As a consequence, the first-order differential $DJ(\boldsymbol{v})$ can be easily computed. Indeed, it is easy to show

that

$$DJ(\boldsymbol{v}) = \left\{ \boldsymbol{v}_i(t) + \int_\Omega y(t) \frac{\partial p(t)}{\partial x_i} dx \right\}_{i=1}^{2}, \text{ a.e. on } (0,T), \forall \boldsymbol{v} \in L^2(0,T;\mathbb{R}^2), \tag{5.33}$$

where $p(t)$ is the solution of the adjoint equation (5.13). The inner preconditioned CG algorithm (**DG1**)-(**DG5**) for the computation of the gradient $\{\boldsymbol{g}_n\}_{n=1}^N$ is thus avoided.

In order to examine the efficiency of the proposed CG algorithm (**DI**)–(**DV**), we construct an example with a known exact solution. To this end, we set $\nu = 1$ and $a_0 = 1$ in (5.2), and

$$y = e^t(-3\sin(2\pi x_1)\sin(\pi x_2) + 1.5\sin(\pi x_1)\sin(2\pi x_2)), \quad p = (T-t)\sin \pi x_1 \sin \pi x_2.$$

Substituting these two functions into the optimality condition $DJ(\boldsymbol{u}(t)) = 0$, we have

$$\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2)^\top = (2e^t(T-t), -e^t(T-t))^\top.$$

We further set

$$\begin{aligned} f &= \frac{\partial y}{\partial t} - \nabla^2 y + \boldsymbol{u} \cdot \nabla y + y, \\ \phi &= -3\sin(2\pi x_1)\sin(\pi x_2) + 1.5\sin(\pi x_1)\sin(2\pi x_2), \\ y_d &= y - \frac{1}{\alpha_1}\left(-\frac{\partial p}{\partial t} - \nabla^2 p - \boldsymbol{u} \cdot \nabla p + p \right), \quad g = 0. \end{aligned}$$

Then, it is easy to verify that $\boldsymbol{u}$ is a solution point of the problem (5.32). We display the solution $\boldsymbol{u}$ and the target function y_d at different instants of time in Figure 5.1 and Figure 5.2, respectively.

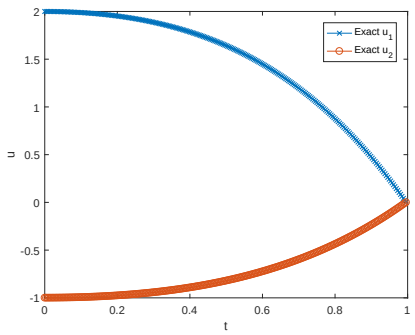


Figure 5.1: The exact optimal control $\boldsymbol{u}$ for Example 1.

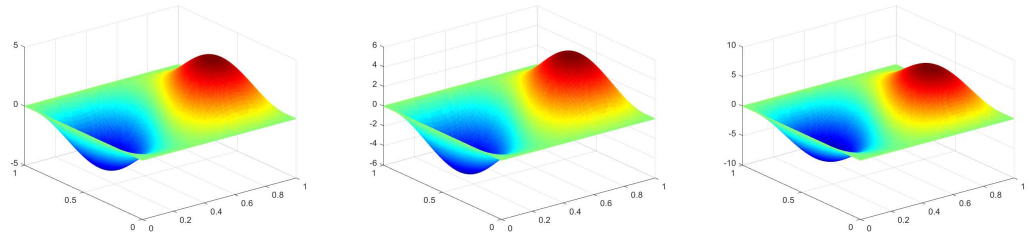


Figure 5.2: The target function y_d at $t = 0.25, 0.5$ and 0.75 (from left to right) for Example 1.

The stopping criterion of the CG algorithm (**DI**)–(**DV**) is set as

$$\frac{\Delta t \sum_{n=1}^{N} |\boldsymbol{g}_n^{k+1}|^2}{\Delta t \sum_{n=1}^{N} |\boldsymbol{g}_n^{0}|^2} \leq 10^{-5}.$$

The initial value is chosen as $\boldsymbol{u}^0 = (0,0)^\top$; and we denote by $\boldsymbol{u}^{\Delta t}$ and $y_h^{\Delta t}$ the computed control and state, respectively.

First, we take $h = \frac{1}{2^i}, i = 5, 6, 7, 8$, $\Delta t = \frac{h}{2}$ and $\alpha_1 = 10^6$, and implement the proposed CG algorithm (**DI**)–(**DV**) for solving the problem (5.32). The numerical results reported in Table 5.1 show that the CG algorithm converges fairly fast and is robust with respect to different mesh sizes. We also observe that the target function y_d has been reached within a good accuracy. Similar comments hold for the approximation of the optimal control $\boldsymbol{u}$ and of the state y of problem (5.32). By taking $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$, the computed state $y_h^{\Delta t}$ and $y_h^{\Delta t} - y_d$ at $t = 0.25, 0.5$ and 0.75 are reported in Figures 5.3, 5.4 and 5.5, respectively; and the computed control $\boldsymbol{u}^{\Delta t}$ and error $\boldsymbol{u}^{\Delta t} - \boldsymbol{u}$ are visualized in Figure 5.6.

Table 5.1: Results of (**DI**)–(**DV**) with different h and Δt for Example 1.

Mesh sizes	*Iter*	$\|\boldsymbol{u}^{\Delta t} - \boldsymbol{u}\|_{L^2(0,T;\mathbb{R}^2)}$	$\|y_h^{\Delta t} - y\|_{L^2(Q)}$	$\frac{\|y_h^{\Delta t} - y_d\|_{L^2(Q)}}{\|y_d\|_{L^2(Q)}}$
$h = 1/2^5, \Delta t = 1/2^6$	117	2.8820×10^{-2}	1.1569×10^{-2}	3.8433×10^{-3}
$h = 1/2^6, \Delta t = 1/2^7$	48	1.3912×10^{-2}	2.5739×10^{-3}	8.5623×10^{-4}
$h = 1/2^7, \Delta t = 1/2^8$	48	6.9095×10^{-3}	4.8574×10^{-4}	1.6516×10^{-4}
$h = 1/2^8, \Delta t = 1/2^9$	31	3.4845×10^{-3}	6.6231×10^{-5}	2.2196×10^{-5}

Furthermore, we tested the proposed CG algorithm (**DI**)–(**DV**) with $h = \frac{1}{2^6}$ and $\Delta t = \frac{1}{2^7}$ for different penalty parameter α_1. The results reported in Table 5.2

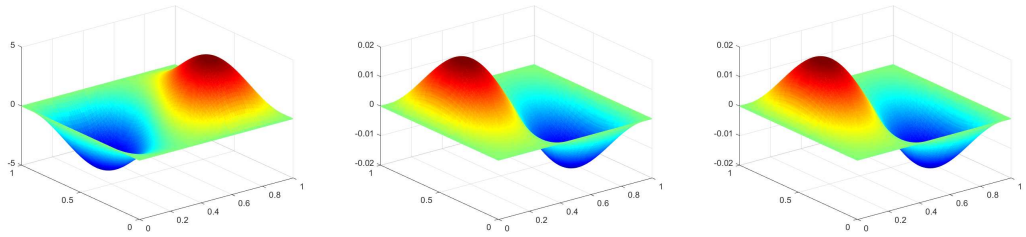


Figure 5.3: Computed state $y_h^{\Delta t}$, error $y_h^{\Delta t} - y$ and $y_h^{\Delta t} - y_d$ (from left to right) at $t = 0.25$ for Example 1.

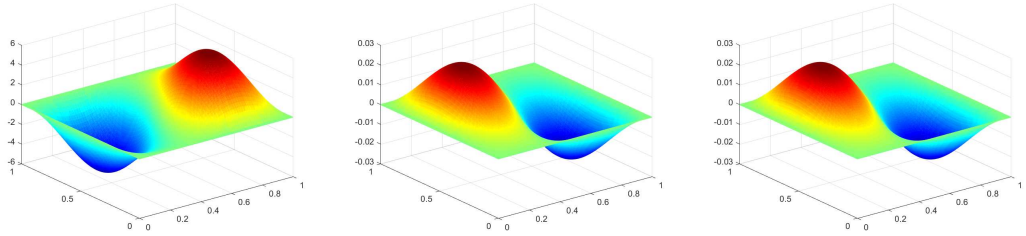


Figure 5.4: Computed state $y_h^{\Delta t}$, error $y_h^{\Delta t} - y$ and $y_h^{\Delta t} - y_d$ (from left to right) at $t = 0.5$ for Example 1.

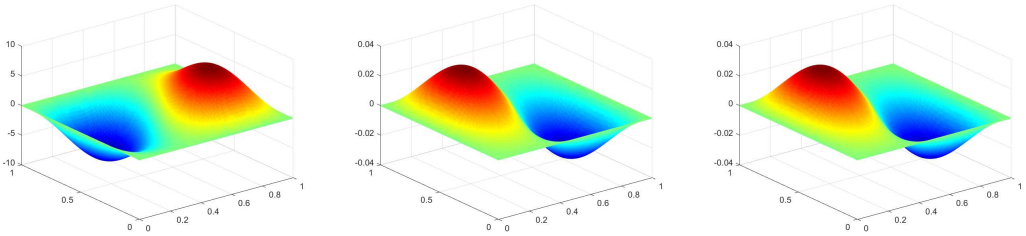


Figure 5.5: Computed state $y_h^{\Delta t}$, error $y_h^{\Delta t} - y$ and $y_h^{\Delta t} - y_d$ (from left to right) at $t = 0.75$ for Example 1.

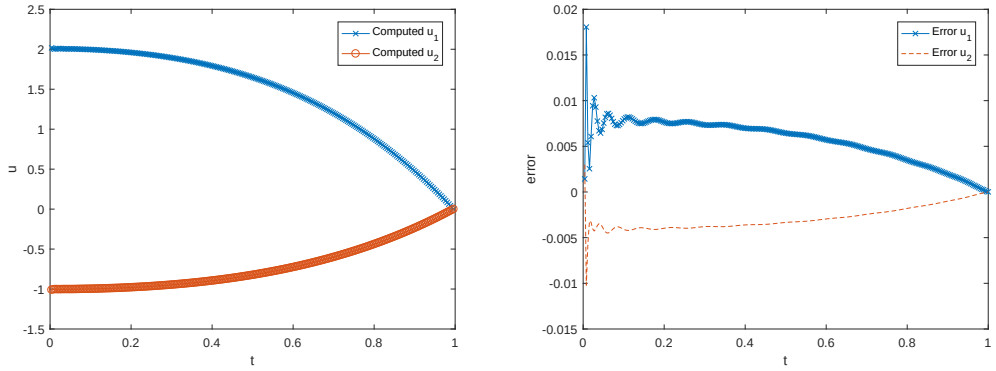


Figure 5.6: Computed optimal control $\boldsymbol{u}^{\Delta t}$ and error $\boldsymbol{u}^{\Delta t} - \boldsymbol{u}$ for Example 1.

show that the performance of the proposed CG algorithm is robust with respect to the penalty parameter, at least for the example being considered. We also observe that as α_1 increases, the value of $\frac{\|y_h^{\Delta t}-y_d\|_{L^2(Q)}}{\|y_d\|_{L^2(Q)}}$ decreases. This implies that, as expected, the computed state $y_h^{\Delta t}$ is closer to the target function y_d when the penalty parameter gets larger.

Table 5.2: Results of (**DI**)–(**DV**) with different α_1 for Example 1.

α_1	$Iter$	$CPU(s)$	$\|\boldsymbol{u}^{\Delta t}-\boldsymbol{u}\|_{L^2(0,T;\mathbb{R}^2)}$	$\|y_h^{\Delta t}-y\|_{L^2(Q)}$	$\frac{\|y_h^{\Delta t}-y_d\|_{L^2(Q)}}{\|y_d\|_{L^2(Q)}}$
10^4	46	126.0666	1.3872×10^{-2}	2.5739×10^{-3}	8.7666×10^{-4}
10^5	48	126.4185	1.3908×10^{-2}	2.5739×10^{-3}	8.6596×10^{-4}
10^6	48	128.2346	1.3912×10^{-2}	2.5739×10^{-3}	8.5623×10^{-4}
10^7	48	127.1858	1.3912×10^{-2}	2.5739×10^{-3}	8.5612×10^{-4}
10^8	48	124.1160	1.3912×10^{-2}	2.5739×10^{-3}	8.5610×10^{-4}

Example 2. We consider the bilinear optimal control problem (BCP) on the domain $Q=\Omega\times(0,T)$ with $\Omega=(0,1)^2$ and $T=1$. Different from Example 1, the control $\boldsymbol{v}(x,t)$ of Example 2 is taken in the infinite-dimensional space $\mathcal{U}=\{\boldsymbol{v}|\boldsymbol{v}\in[L^2(Q)]^2,\nabla\cdot\boldsymbol{v}=0\}$. We set $\alpha_2=0$ in (5.1), $\nu=1$ and $a_0=1$ in (5.2), and consider the following tracking-type bilinear optimal control problem:

$$\min_{\boldsymbol{v}\in\mathcal{U}} J(\boldsymbol{v})=\frac{1}{2}\iint_Q|\boldsymbol{v}|^2dxdt+\frac{\alpha_1}{2}\iint_Q|y-y_d|^2dxdt, \tag{5.34}$$

where y is obtained from $\boldsymbol{v}$ via the solution of the state equation (5.2).

First, we let

$$\begin{aligned}
&y=e^t(-3\sin(2\pi x_1)\sin(\pi x_2)+1.5\sin(\pi x_1)\sin(2\pi x_2)),\\
&p=(T-t)\sin\pi x_1\sin\pi x_2,\ \text{and}\ \boldsymbol{u}=P_{\mathcal{U}}(p\nabla y),
\end{aligned}$$

where $P_{\mathcal{U}}(\cdot)$ is the projection onto the set $\mathcal{U}$.

We further set

$$\begin{aligned}
&f=\frac{\partial y}{\partial t}-\nabla^2y+\boldsymbol{u}\cdot\nabla y+y,\\
&\phi=-3\sin(2\pi x_1)\sin(\pi x_2)+1.5\sin(\pi x_1)\sin(2\pi x_2),\\
&y_d=y-\frac{1}{\alpha_1}\left(-\frac{\partial p}{\partial t}-\nabla^2p-\boldsymbol{u}\cdot\nabla p+p\right),\quad g=0.
\end{aligned}$$

Then, it is easy to show that $\boldsymbol{u}$ is a solution point of the problem (5.34). We note that $\boldsymbol{u} = P_{\mathcal{U}}(p\nabla y)$ has no analytical solution and it can only be solved numerically. Here, we solve $\boldsymbol{u} = P_{\mathcal{U}}(p\nabla y)$ by the preconditioned CG algorithm (**DG1**)–(**DG5**) with $h = \frac{1}{2^9}$ and $\Delta t = \frac{1}{2^{10}}$, and use the resulting control $\boldsymbol{u}$ as a reference solution for the example we considered.

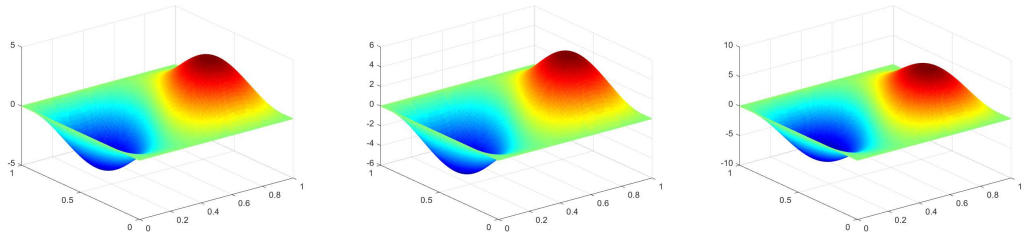


Figure 5.7: The target function y_d with $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$ at $t = 0.25, 0.5$ and 0.75 (from left to right) for Example 2.

The stopping criteria of the outer CG algorithm (**DI**)–(**DV**) and the inner preconditioned CG algorithm (**DG1**)–(**DG5**) are respectively set as

$$\frac{\Delta t \sum_{n=1}^{N} \int_\Omega |\boldsymbol{g}_n^{k+1}|^2 dx}{\Delta t \sum_{n=1}^{N} \int_\Omega |\boldsymbol{g}_n^{0}|^2 dx} \leq 5 \times 10^{-8}, \text{ and } \frac{\int_\Omega |\nabla r^{k+1}|^2 dx}{\max\{1, \int_\Omega |\nabla r^0|^2 dx\}} \leq 10^{-8}.$$

The initial values are chosen as $\boldsymbol{u}^0 = (0,0)^\top$ and $\lambda^0 = 0$; and we denote by $\boldsymbol{u}_h^{\Delta t}$ and $y_h^{\Delta t}$ the computed control and state, respectively.

First, we take $h = \frac{1}{2^i}, i = 6, 7, 8$, $\Delta t = \frac{h}{2}$, $\alpha_1 = 10^6$, and implement the proposed nested CG algorithm (**DI**)–(**DV**) for solving the problem (5.34). The numerical results reported in Table 5.3 show that the CG algorithm converges fast and is robust with respect to different mesh sizes. In addition, the preconditioned CG algorithm (**DG1**)–(**DG5**) converges within 10 iterations for all cases and thus is efficient for computing the gradient $\{\boldsymbol{g}_n\}_{n=1}^N$. We also observe that the target function y_d has been reached within a good accuracy. Similar comments hold for the approximation of the optimal control $\boldsymbol{u}$ and of the state y of problem (5.34).

Taking $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$, the computed state $y_h^{\Delta t}$, the error $y_h^{\Delta t} - y$ and $y_h^{\Delta t} - y_d$ at $t = 0.25, 0.5, 0.75$ are reported in Figures 5.8, 5.9 and 5.10, respectively; and the computed control $\boldsymbol{u}_h^{\Delta t}$, the exact control $\boldsymbol{u}$, and the error

Table 5.3: Results of (**DI**)–(**DV**) with different h and Δt for Example 2.

Mesh sizes	$Iter_{CG}$	PCG	$\|\boldsymbol{u}_h^{\Delta t} - \boldsymbol{u}\|_{L^2(Q)}$	$\|y_h^{\Delta t} - y\|_{L^2(Q)}$	$\frac{\|y_h^{\Delta t} - y_d\|_{L^2(Q)}}{\|y_d\|_{L^2(Q)}}$
$h = 1/2^6, \Delta t = 1/2^7$	443	9	3.7450×10^{-3}	9.7930×10^{-5}	1.0906×10^{-6}
$h = 1/2^7, \Delta t = 1/2^8$	410	9	1.8990×10^{-3}	1.7423×10^{-5}	3.3863×10^{-7}
$h = 1/2^8, \Delta t = 1/2^9$	405	8	1.1223×10^{-3}	4.4003×10^{-6}	1.0378×10^{-7}

$\boldsymbol{u}_h^{\Delta t} - \boldsymbol{u}$ at $t = 0.25, 0.5, 0.75$ are presented in Figures 5.11, 5.12 and 5.13.

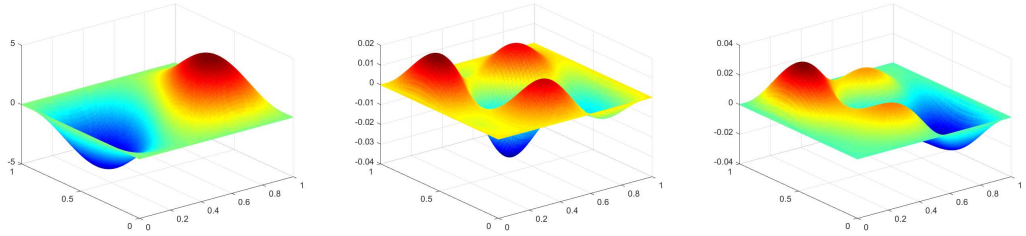


Figure 5.8: Computed state $y_h^{\Delta t}$, error $y_h^{\Delta t} - y$ and $y_h^{\Delta t} - y_d$ with $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$ (from left to right) at $t = 0.25$ for Example 2.

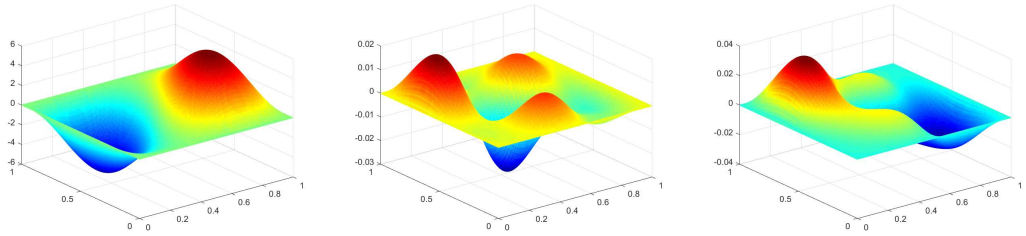


Figure 5.9: Computed state $y_h^{\Delta t}$, error $y_h^{\Delta t} - y$ and $y_h^{\Delta t} - y_d$ with $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$ (from left to right) at $t = 0.5$ for Example 2.

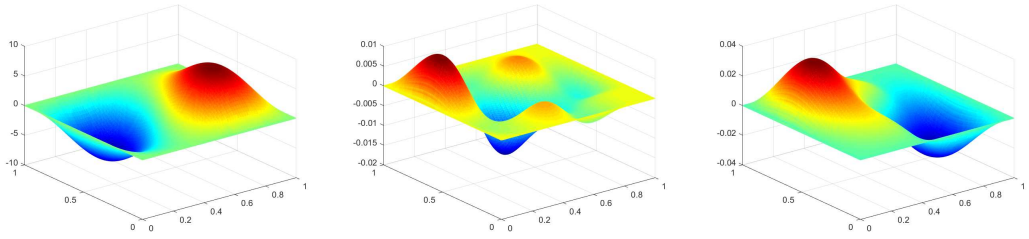


Figure 5.10: Computed state $y_h^{\Delta t}$, error $y_h^{\Delta t} - y$ and $y_h^{\Delta t} - y_d$ with $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$ (from left to right) at $t = 0.75$ for Example 2.

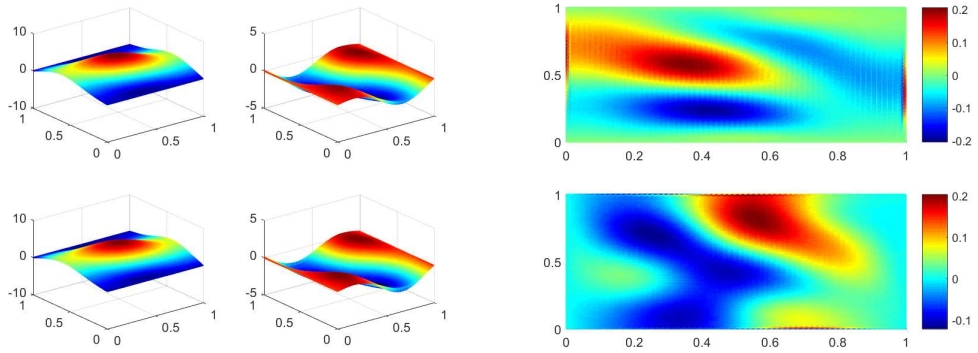


Figure 5.11: Computed control $\boldsymbol{u}_h^{\Delta t}$ and exact control $\boldsymbol{u}$ (left, from top to bottom) and the error $\boldsymbol{u}_h^{\Delta t} - \boldsymbol{u}$ (right) with $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$ at $t = 0.25$ for Example 2.

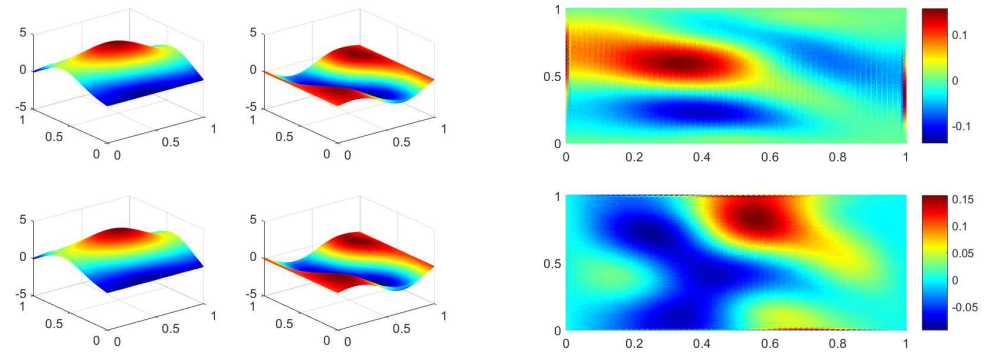


Figure 5.12: Computed control $\boldsymbol{u}_h^{\Delta t}$ and exact control $\boldsymbol{u}$ (left, from top to bottom) and the error $\boldsymbol{u}_h^{\Delta t} - \boldsymbol{u}$ (right) with $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$ at $t = 0.5$ for Example 2.

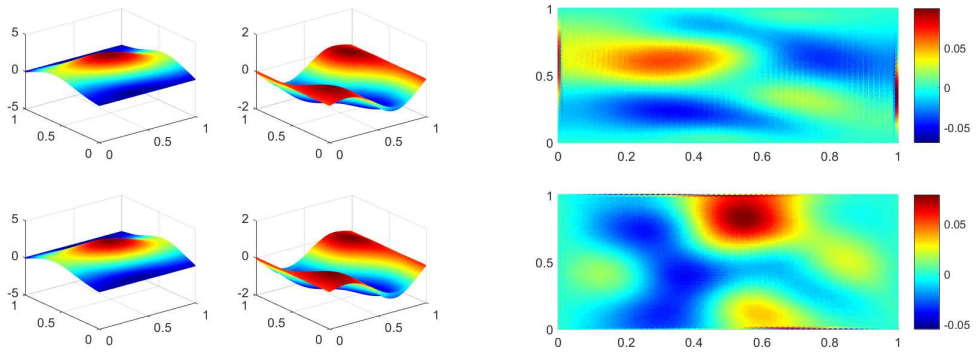


Figure 5.13: Computed control $\boldsymbol{u}_h^{\Delta t}$ and exact control $\boldsymbol{u}$ (left, from top to bottom) and the error $\boldsymbol{u}_h^{\Delta t} - \boldsymbol{u}$ (right) with $h = \frac{1}{2^7}$ and $\Delta t = \frac{1}{2^8}$ at $t = 0.75$ for Example 2.

Chapter 6

Conclusions and future works

In this chapter, we make some conclusions of this thesis and present some directions for future works, which are summarized as below.

- In Chapter 2, we considered a class of application-driven nonlinear saddle point problems and proposed an algorithmic framework based on some known inexact Uzawa methods. The convergence and linear convergence rate of the algorithmic framework were uniformly analyzed. Then, we focused on an elliptic optimal control problem with control constraints and discussed how to choose appropriate preconditioners to specify the algorithmic framework as an easily implementable and efficient algorithm. Preliminary numerical results were reported to verify all the theoretical assertions including the convergence and linear convergence rate of our proposed inexact Uzawa type algorithm, as well as the convergence order of the finite element discretization.

  Our philosophy in algorithmic design and techniques for theoretical analysis and numerical implementation can be easily extended to other problems such as the optimal control problems constrained by the convection-diffusion equation [9] or Stokes equation [155], or elliptic variational inequalities of the second kind [78]. Moreover, our experiments empirically revealed that the mesh size of finite element discretization does not effect the performance of the proposed inexact Uzawa type algorithm. It will be

interesting to fathom the theory behind.

- In Chapter 3, we focused on the implementation of the well-known alternating direction method of multipliers (ADMM) to parabolic optimal control problems with control constraints. Direct implementation of ADMM decouples the control constraint and the parabolic state equation at each iteration, while the resulting unconstrained parabolic optimal control subproblems should be solved inexactly. Hence, only inexact versions of the ADMM are implementable for these problems. We proposed an easily implementable inexactness criterion for these subproblems; and obtained an inexact version of the ADMM whose execution consists of two-layer nested iterations. The strong global convergence of the resulting inexact ADMM was proved rigorously in an infinite-dimensional Hilbert space; and the worst-case convergence rate measured by the iteration complexity was also established. We illustrated by the CG method how to execute the inexactness criterion, and showed the efficiency of the resulting ADMM–CG iterative scheme numerically. In particular, our numerical results validate that usually a few internal CG iterations are sufficient to guarantee the overall convergence of the ADMM–CG; hence there is no need to solve the unconstrained parabolic optimal control problem at each iteration up to a high precision. This fact significantly saves computation and contributes to the efficiency of the ADMM–CG. As mentioned in Remark 3.1, the new inexactness criterion possesses a variety of features that are software-friendly and hence important for softwarization and industrialization. In this sense, we follow the fundamental concept of trustworthiness in software engineering (also in artificial intelligence) and call the proposed inexact ADMM, or more concretely Algorithm 3.3, a trustworthy algorithm.

  Our philosophy in algorithmic design can be easily extended to other optimal control problems; hence the proposed inexact ADMM can be deliberately specified as various algorithms for a wide range of optimal control problems. For some challenging problems whose numerical study is limited (such as the general case of (3.1)–(3.2) or (3.50)–(3.51) where $\omega \subsetneq \Omega$ and $d \geq 2$), the algorithms specified from the inexact ADMM are attractive in senses of numerical performance and easiness of coding. It is interesting

and much more challenging to design operator splitting type algorithms for optimal control problems constrained by some nonlinear PDEs in the future.

- In Chapter 4, we have discussed the sparse initial source identification for diffusion-advection equations. More precisely, we have designed an algorithm capable of recovering the unique initial configuration leading the solution of our model to match with a prescribed final target in a given time horizon T. Our main interest being to identify moving pollution sources traveling in either a compressible or incompressible fluid, we assumed that the initial condition is a linear combination of Dirac measures indicating the location of the sources, with their weights representing the intensity of the sources. The algorithm we proposed to solve the source identification problem is comprised of two steps. Firstly, we formulated an optimal control problem with a suitable functional consists of three terms:

  1. A first term seeking for an initial condition u_0 such that the corresponding solution, at time $t = T$, is as close as possible to the desired target.
  2. A second term, involving the L^1 norm of the initial datum to detect sparsity.
  3. A third Tikhonov regularization term, introduced to guarantee the well-posedness of the problem while improving the conditioning of the optimal control problem.

  We introduced a generalized PDHG-based prediction-correction algorithmic framework to obtain the location information of the sources by solving the resulting optimal control problem. Secondly, an optimization problem in terms of the locations and a least squares fitting corresponding to the intensities are considered to find the optimal locations and intensities of the sources, respectively. In our numerical simulations, we have considered several test cases where the algorithm identifies the initial sources from a reachable target or noisy observation very successfully even for some heterogeneous materials or coupled models.

Nevertheless, our work left several unaddressed key aspects of the proposed source identification problem, which will be to subject of future investigation.

1. First of all, our simulations have shown that the proposed algorithm is capable of accurately recover the initial source configuration in short time horizons. On the other hand, when the time horizon is too large, the source identification for (4.1) is unsuccessful, as it can be appreciated in Figure 6.1. This issue, highly related to the diffusivity parameter and the ill-posedness of the backward heat problem, has already been mentioned in previous research works (see e.g., [134, 135]). It would then be interesting to estimate the maximum final time at which the recovery is still feasible and to design novel and efficient algorithms allowing to address the source identification problem in large time horizons. To the best of our knowledge, this important issue is still open in the literature.
2. Another future direction would be to consider the case of strong jumps in the material coefficients. When the diffusivities of the coupled materials are very different from each other, we can observe in Figure 6.2 that the recovered initial source is not correct. A possible solution would be to combine our approach with some splitting methods in order to parallelize the computations in each of the subdomains.
3. Finally, it would be of interest to extend our proposed numerical approach to some more complicated geometries or nonlinear models.

- In Chapter 5, we studied the bilinear control of an advection-reaction-diffusion system, where the control variable enters the model as a velocity field of the advection term. Mathematically, we proved the existence of optimal controls and derived the associated first-order optimality conditions. Computationally, the conjugate gradient (CG) method was suggested and its implementation is nontrivial. In particular, an additional divergence-free constraint on the control variable leads to a projection subproblem to compute the gradient; and the computation of a stepsize at each CG iteration requires solving the state equation repeatedly due to the nonlin-

Figure 6.1: Numerical results for the case of $v = (1, 2)^T$, $T = 0.5$ and $d = 0.05$.

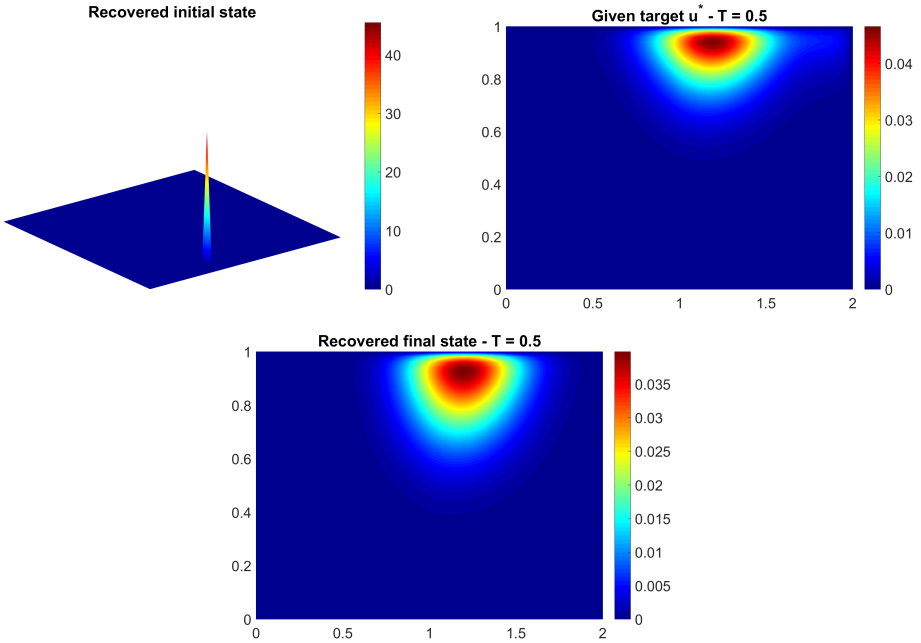
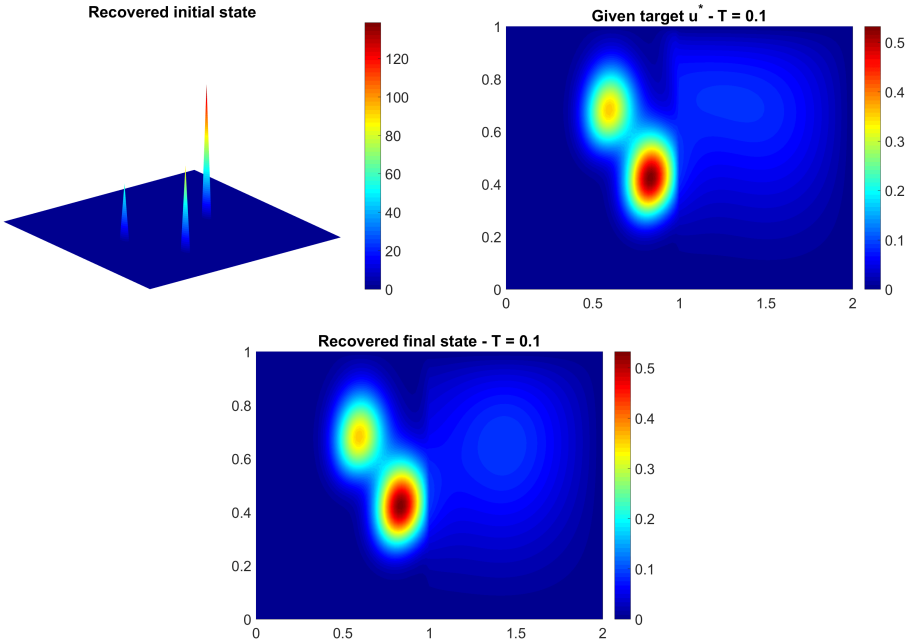


Figure 6.2: Numerical results for the case of $v = (1, 2)^T$, $T = 0.1$, $d = 0.05$ in $\Omega_1 = (0, 1) \times (0, 1)$ and $d = 0.5$ in $\Omega_2 = (1, 2) \times (0, 1)$.



ear relation between the state and control variables. To resolve the above issues, we reformulated the gradient computation as a Stokes-type problem and proposed a fast preconditioned CG method to solve it. We also proposed an efficient inexactness strategy to determine the stepsize, which only requires the solution of one linear parabolic equation. An easily implementable nested CG method was thus proposed. For the numerical discretization, we employed the standard piecewise linear finite element method and the Bercovier-Pironneau finite element method for the space discretizations of the bilinear optimal control and the Stokes-type problem, respectively, and a semi-implicit finite difference method for the time discretization. The resulting algorithm was shown to be numerically efficient by some preliminary numerical experiments.

We focused in this chapter on an advection-reaction-diffusion system controlled by a general form velocity field. In a real physical system, the velocity field may be determined by some partial differential equations (PDEs), such as the Navier-Stokes equations. As a result, we meet some bilinear optimal control problems constrained by coupled PDE systems. Moreover, instead of (1.14), one can also consider other types of objective functionals in the bilinear optimal control of an advection-reaction-diffusion system. For instance, one can incorporate $\iint_Q |\nabla \boldsymbol{v}|^2 dxdt$ and $\iint_Q |\frac{\partial \boldsymbol{v}}{\partial t}|^2 dxdt$ into the objective functional to promote that the optimal velocity field has the least rotation and is almost steady, respectively, which are essential in e.g., mixing enhancement for different flows [128]. All these problems are of practical interest but more challenging from algorithmic design perspectives, and they have not been well-addressed numerically in the literature. Our current work has laid a solid foundation for solving these problems and we leave them in the future.

Bibliography

[1] S. G. Andrade and A. Borzì, *Multigrid second-order accurate solution of parabolic control-constrained problems*, Computational Optimization and Applications, 51 (2012), pp. 835–866.

[2] K.J. Arrow, L. Hurwicz and H. Uzawa, *Studies in linear and non-linear programming*, Stanford Mathematical Studies in the Social Sciences, 1958.

[3] U. M. Ascher, S. J. Ruuth and B. T. Wetton, *Implicit-explicit methods for time-dependent partial differential equations*, SIAM Journal on Numerical Analysis, 32 (1995), pp. 797–823.

[4] H. Attouch and M. Soueycatt, *Augmented Lagrangian and proximal alternating direction methods of multipliers in Hilbert spaces: applications to games, PDE's and control*, Pacific Journal of Optimization, 5 (2008), pp. 17–37.

[5] C. Bacuta, *A unified approach for Uzawa algorithms*, SIAM Journal on Numerical Analysis, 44 (2006), pp. 2633–2649.

[6] R.E. Bank, B.D. Welfert and H. Yserentant, *A class of iterative methods for solving saddle point problems*, Numerische Mathematik, 56 (1989), pp. 645–666.

[7] A. T. Barker and M. Stoll, *Domain decomposition in time for PDE-constrained optimization*, Computer Physics Communications, 197 (2015), pp. 136–143.

[8] S. Bartels, *Total variation minimization with finite elements: convergence and iterative solution*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 1162–1180.

[9] R. Becker and B. Vexler, *Optimal control of the convection-diffusion equation using stabilized finite element methods*, Numerische Mathematik, 106 (2007), pp. 349–367.

[10] M. Benzi, G.H. Golub and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[11] M. Bercovier and O. Pironneau, *Error estimates for finite element method solution of the Stokes problem in the primitive variables*. Numerische Mathematik, 33 (1979), pp. 211–224.

[12] M. Berggren and R. Glowinski, *Controllability issues for flow-related models: a computational approach*, Technical report TR94-47, Rice University, Houston, TX, 1994.

[13] M. Berggren, R. Glowinski and J. L. Lions, *A computational approach to controllability issues for flow-related models. (I): pointwise control of the viscous Burgers equation*, International Journal of Computational Fluid Dynamics, 7 (1996), pp. 237–252.

[14] M. Bergounioux, K. Ito and K. Kunisch, *Primal-dual strategy for constrained optimal control problems*, SIAM Journal on Control and Optimization, 37 (1999), pp. 1176–1194.

[15] D. Boffi, F. Brezzi and M. Fortin, *Mixed Finite Element Methods and Applications*, vol. 44, Springer, 2013.

[16] A. Borzì, *Multigrid methods for parabolic distributed optimal control problems*, Journal of Computational and Applied Mathematics, 157 (2003), pp. 365–382.

[17] A. Borzì, E.-J. Park and M. Vallejos Lass, *Multigrid optimization methods for the optimal control of convection-diffusion problems with bilinear control*, Journal of Optimization Theory and Applications, 168 (2016), pp. 510–533.

[18] A. Borzì and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, vol. 8, SIAM, 2011.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends ® in Machine learning, 3 (2011), pp. 1–122.

[20] J.H. Bramble and J.E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Mathematics of Computation, 50 (181), pp. 1–17.

[21] J.H. Bramble, J.E. Pasciak and A.T. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM Journal on Numerical Analysis, 34 (1997), pp. 1072–1092.

[22] K. Bredies and H. Sun, *Preconditioned Douglas-Rachford splitting methods for convex-concave saddle-point problems*, SIAM Journal on Numerical Analysis, 53 (2015), pp. 421–444.

[23] F. Brezzi, J. Douglas, and L. D. Marini, *Two families of mixed elements for second order elliptic problems*, Numerische Mathematik, 47 (1985), pp. 217–235.

[24] L.M. Bregman, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR Comput. Math. Math. Phys. 7, 3 (1967), pp. 200–217.

[25] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, vol. 15, Springer Science & Business Media, 2012.

[26] X.L. Briggs, V.E. Henson and S.F. McCormick, *A Multigrid Tutorial*, vol. 72, SIAM, 2000.

[27] W. Bu, Y. Tang and J. Yang, *Galerkin Finite Element Method for Two-dimensional Riesz Space Fractional Diffusion Equations*, Journal of Computational Physics, 276, pp. 26–38, 2014.

[28] P. Cannarsa, G. Floridia and A.Y. Khapalov, *Multiplicative controllability for semilinear reactiondiffusion equations with finitely many changes of sign*, Journal de Mathématiques Pures et Appliquées, 108 (2017), pp. 425–458.

[29] C. Carthel, R. Glowinski and J.L. Lions, *On exact and approximate boundary controllabilities for the heat equation: a numerical approach*, Journal of Optimization Theory and Applications, 82 (1994), pp. 429–484.

[30] E. Casas, *Using piecewise linear functions in the numerical approximation of semilinear elliptic control problems*, Advances in Computational Mathematics, 26 (2007), pp. 137–153.

[31] E. Casas, *A review on sparse solutions in optimal control of partial differential equations*, SeMA Journal, 74(2017), pp. 319–344.

[32] E. Casas, C. Clason and K. Kunisch, *Approximation of elliptic control problems in measure spaces with sparse solutions*, SIAM Journal on Control and Optimization, 50 (2012), pp. 1735–1752.

[33] E. Casas, C. Clason and K. Kunisch, *Parabolic control problems in measure spaces with sparse solutions*, SIAM Journal on Control and Optimization, 51 (2013), pp. 28–63.

[34] E. Casas and K. Kunisch, *Using sparse control methods to identify sources in linear diffusion-convection equations*, Inverse Problem, 35 (2019), pp. 114002.

[35] E. Casas, B. Vexler and E. Zuazua, *Sparse initial data identification for parabolic PDE and its finite element approximations*, AIMS, 5 (2015), pp. 377–399.

[36] E. Casas and E. Zuazua, *Spike controls for elliptic and parabolic PDEs*, Systems & Control Letters, 62 (2013), pp. 311–318.

[37] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145.

[38] T.F. Chan and R. Glowinski, *Finite element approximation and iterative solution of a class of mildly non-linear elliptic equations*, Stanford Computer Science Report STAN-CS-78-674, Stanford University, Stanford, CA (1978).

[39] L. Chen, *iFEM: an innovative finite element methods package in MATLAB*, (2008).

[40] X. Chen, *Global and superlinear convergence of inexact Uzawa methods for saddle point problems with nondifferentiable mappings*, SIAM Journal on Numerical Analysis, 35 (1998), pp. 1130–1148.

[41] Z. Chen, Q. Du and J. Zou, *Finite element methods with matching and nonmatching meshes for Maxwell equations with discontinuous coefficients*, SIAM Journal on Numerical Analysis, 37 (2000), pp. 1542–1570.

[42] Z. Chen and J. Zou, *An augmented Lagrangian method for identifying discontinuous parameters in elliptic systems*, SIAM Journal on Control and Optimization, 37 (1999), pp. 892–910.

[43] X. Cheng and W. Han, *Inexact Uzawa algorithms for variational inequalities of the second kind*, Computer Methods in Applied Mechanics and Engineering, 192 (2003), pp. 1451–1462.

[44] G. Ciaramella and A. Borzì, *A LONE code for the sparse control of quantum systems*, Computer Physics Communications, 200 (2016), pp. 312–323.

[45] F.H. Clarke, *Optimization and Nonsmooth Analysis*, vol. 5, SIAM, 1990.

[46] C. Clason and K. Kunisch, *A duality-based approach to elliptic control problems in non-reflexive Banach spaces*, ESAIM: Control, Optimisation and Calculus of Variations, 17 (2011), pp. 243–266.

[47] C. Clason and T. Valkonen, *Primal-dual extragradient methods for nonlinear nonsmooth PDE-constrained optimization*, SIAM Journal on Optimization, 27 (2017), pp. 1314–1339.

[48] L. Dede' and A. Quarteroni, *Optimal control and numerical adaptivity for advection-diffusion equations*, ESAIM: Mathematical Modelling and Numerical Analysis, 39 (2005), pp. 1019–1040.

[49] J.C. De los Reyes, *Numerical PDE-constrained Optimization*, Springer, 2015.

[50] P. Destuynder and T. Nevers, *Some numerical aspects of mixed finite elements for bending plates*, Computer Methods in Applied Mechanics and Engineering, 78 (1990), pp. 73–87.

[51] J. Douglas and H.H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*. Transactions of the American mathematical Society, 82 (1956), pp. 421–439.

[52] J. Eckstein and D. Bertsekas, *On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Mathematical Programming, 55 (1992), pp. 293–318.

[53] J. Eckstein and W. Yao, *Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results*, RUTCOR Research Reports, 32 (2012): 44.

[54] J. Eckstein and W. Yao, *Relative-error approximate versions of Douglas–Rachford splitting and special cases of the ADMM*, Mathematical Programming, 170 (2018), pp. 417–444.

[55] I. Ekeland and R. Temam, *Convex Analysis and Variational Problems*, SIAM, 1999.

[56] A. El Badia, T. Ha-Duong and A. Hamdi, *Identification of a point source in a linear advection-dispersion-reaction equation: application to a pollution source problem*, Inverse Problem, 21 (2005), pp. 1121.

[57] H.C. Elman and G.H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM Journal on Numerical Analysis, 31 (1994), pp. 1645–1661.

[58] H.C. Elman, D.J. Silvester and A.J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2014.

[59] H.W. Engl, A.K. Louis and W. Rundell, *Inverse problems in medical imaging and nondestructive testing: proceedings of the conference in Oberwolfach, Germany, February 4-10, 1996*. Springer Science & Business Media, 2012.

[60] E. Esser, X. Zhang and T.F. Chan, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 1015–1046.

[61] R.E. Ewing and M.F. Wheeler, *Computational aspects of mixed finite element methods*, Numerical Methods for Scientific Computing, 1983, pp. 163–172.

[62] F. Facchinei and J.S. Pang, *Finite-dimensional Variational Inequalities and Complementarity Problems*, Springer Science & Business Media, 2007.

[63] R. D. Falgout, *An introduction to algebraic multigrid*, Computing in Science and Engineering, 8 (2006), pp. 24–33.

[64] R.S. Falk, *Approximation of a class of optimal control problems with order of convergence estimates*, Journal of Mathematical Analysis and Applications, 44 (1973), pp. 28–47.

[65] R.S. Falk, *An analysis of the finite element method using Lagrange multipliers for the stationary stokes equation*, Mathematics of Computation, 30 (1976), pp. 241–249.

[66] A. Fleig and R. Guglielmi, *Optimal control of the Fokker–Planck equation with space-dependent controls*, Journal of Optimization Theory and Applications, 174 (2017), pp. 408–427.

[67] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-value Problems*, North-Holland, Amsterdam, 1983.

[68] M. Fortin and R. Glowinski, *Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems*, vol. 15, Elsevier, 2000.

[69] D. Gabay and B. Mercier, *A dual algorithm for the solution of non linear variational problems via finite element approximation.* Computers & Mathematics with Applications, 2 (1976): pp. 17–40.

[70] M.J. Gander and F. Kwok, *Schwarz methods for the time-parallel solution of parabolic control problems*, Domain Decomposition Methods in Science and Engineering XXII, Springer, 2016, pp. 207–216.

[71] R. Gilbert, Z. Lin and J. Buchanan, *Direct and inverse problems in ocean acoustics*, Nonlinear Analysis: Theory, Methods and Applications, 30 (1997), pp. 1535–1546.

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

[73] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer-Verlag, 1984.

[74] R. Glowinski, *Ensuring well-posedness by analogy; Stokes problem and boundary control for the wave equation*, Journal of Computational Physics, 103 (1992), pp. 189–221.

[75] R. Glowinski, *Finite Element Methods for Incompressible Viscous Flow*, Handbook of Numerical Analysis, 9 (2003), pp. 3–1176.

[76] R. Glowinski, *Lectures on Numerical Methods for Non-linear Variational Problems*, Springer Science & Business Media, 2008.

[77] R. Glowinski, *On alternating direction methods of multipliers: a historical perspective*, in Modeling, Simulation and Optimization for Science and Technology, Springer, 2014, pp. 59–82.

[78] R. Glowinski, *Variational Methods for the Numerical Solution of Nonlinear Elliptic Problems*, SIAM, 2015.

[79] R. Glowinski and J. He, *On shape optimization and related issues*, In Computational Methods for Optimal Design and Control, J. Borggaard, J. Burns, E. Cliff & S. Schreck (eds.), Birkhuser, Boston, MA, 1998, pp. 151–179.

[80] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics*, vol. 9, SIAM, 1989.

[81] R. Glowinski and J.L. Lions, *Exact and approximate controllability for distributed parameter systems, Part I*, Acta Numerica, 3 (1994), pp. 269–378.

[82] R. Glowinski and J.L. Lions, *Exact and approximate controllability for distributed parameter systems, Part II*, Acta Numerica, 4 (1995), pp. 159–328.

[83] R. Glowinski, J.L. Lions and J. He, *Exact and Approximate Controllability for Distributed Parameter Systems: A Numerical Approach (Encyclopedia of Mathematics and its Applications)*, Cambridge University Press, 2008.

[84] R. Glowinski, Y. Song and X. Yuan, *An ADMM numerical approach to linear parabolic state constrained optimal control problems*, Numerische Mathematik, 144 (2020), pp. 931–966.

[85] T. Goldstein, M. Li and X. Yuan, *Adaptive primal-dual splitting methods for statistical learning and image processing*, In Advances in neural information processing systems (2015), pp. 2089–2097.

[86] C. Gräser and R. Kornhuber, *On preconditioned Uzawa-type iterations for a saddle point problem with inequality constraints*, in Domain decomposition methods in science and engineering XVI, Springer, 2007, pp. 91–102.

[87] G. Gurarslan and H. Karahan, *Solving inverse problems of groundwater-pollution-source identification using a differential evolution algorithm*, Hydrogeol Journal, 23 (2015), pp. 1109–1119.

[88] J. Guzmán and L. Scott, *The Scott-Vogelius finite elements revisited*, Mathematics of Computation, 88 (2019), pp. 515–529.

[89] W. W. Hager, *Runge–Kutta methods in optimal control and the transformed adjoint system*, Numerische Mathematik, 87 (2000), pp. 247–282.

[90] N. Handagama and S. Lenhart, *Optimal control of a PDE/ODE system modeling a gas-phase bioreactor*, In Mathematical Models in Medical and Health Sciences, M. A. Horn, G. Simonett, and G. Webb (eds.), Vanderbilt University Press, Nashville, TN, 1998.

[91] Y. Hao, X. Wang, H. Song and K. Zhang, *An alternating direction method of multipliers for the optimization problem constrained with a stationary Maxwell system*, Communications in Computational Physics, 24 (2018), pp. 1435–1454.

[92] B. He, L. Z. Liao, D. Han and H. Yang, *A new inexact alternating directions method for monotone variational inequalities*, Mathematical Programming, 92 (2002), pp. 103–118.

[93] B. He, F. Ma and X. Yuan, *An algorithmic framework of generalized primal-dual hybrid gradient methods for saddle point problems*, Journal of Mathematical Imaging and Vision, 58 (2017), pp. 279–293.

[94] B. He, Y. You and X. Yuan, *On the convergence of primal-dual hybrid gradient algorithm*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 2526–2537.

[95] B. He and X. Yuan, *Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective*, SIAM Journal on Imaging Science, 5 (2012), pp. 119–149.

[96] B. He and X. Yuan, *On the O(1/n) convergence rate of the Douglas–Rachford alternating direction method*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 700–709.

[97] B. He and X. Yuan, *On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers*, Numerische Mathematik, 130 (2015), pp. 567–577.

[98] M. Heinkenschloss, *A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems*, Journal of Computational and Applied Mathematics, 173 (2005), pp. 169–198.

[99] R. Herzog and E. Sachs, *Preconditioned conjugate gradient method for optimal control problems with control and state constraints*, SIAM Journal on Matrix Analysis and Application, 31 (2010), pp. 2291–2317.

[100] M.R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization Theory and Applications, 4 (1969), pp. 303–320.

[101] M. Hintermüller, K. Ito and K. Kunisch, *The primal-dual active set strategy as a semismooth newton method*, SIAM Journal on Optimization, 13 (2002), pp. 865–888.

[102] M. Hintermüller, I. Kopacka and S. Volkwein, *Mesh-independence and preconditioning for solving parabolic control problems with mixed control-state constraints*, ESAIM: Control, Optimisation and Calculus of Variations 15 (2009), pp. 626–652.

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

[104] M. Hinze and M. Vierling, *The semi-smooth Newton method for variationally discretized control constrained elliptic optimal control problems; implementation, convergence and globalization*, Optimization Methods and Software, 27 (2012), pp. 933–950.

[105] Q. Hu and J. Zou, *Two new variants of nonlinear inexact Uzawa algorithms for saddle-point problems*, Numerische Mathematik, 93 (2002), pp. 333–359.

[106] Q. Hu and J. Zou, *Nonlinear inexact Uzawa algorithms for linear and nonlinear saddle-point problems*, SIAM Journal on Optimization, 16 (2006), pp. 798–825.

[107] V. Isakov, *Inverse Problems for Partial Differential Equations*, vol. 127 of Applied Mathematical Sciences, Springer, Cham, third ed., 2017.

[108] K. Ito and K. Kunisch, *Optimal bilinear control of an abstract Schrödinger equation*, SIAM Journal on Control and Optimization, 46 (2007), pp. 274–287.

[109] H.R. Joshi, *Optimal control of the convective velocity coefficient in a parabolic problem*, Nonlinear Analysis: Theory, Methods & Applications, 63 (2005), pp. e1383–e1390.

[110] L. Justen and R. Ramlau, *A general framework for soft-shrinkage with applications to blind deconvolution and wavelet denoising*, Applied and Computational Harmonic Analysis, 26 (2009), 43–63.

[111] A.Y. Khapalov, *Controllability of the semilinear parabolic equation governed by a multiplicative control in the reaction term: a qualitative approach*, SIAM Journal on Control and Optimization, 41 (2003), pp. 1886–1900.

[112] A.Y. Khapalov, *Controllability of Partial Differential Equations Governed by Multiplicative Controls*, Springer, 2010.

[113] G.M. Korpelevič, An extragradient method for finding saddle points and for other problems, Ekonom. ‘ i Mat. Metody, 12 (1976), pp. 747–756.

[114] A. Kröner, K. Kunisch and B. Vexler, *Semismooth Newton methods for optimal control of the wave equation with control constraints*, SIAM Journal on Control and Optimization 49 (2011), pp. 830–858.

[115] A. Kröner and B. Vexler, *A priori error estimates for elliptic optimal control problems with a bilinear state equation*, Journal of Computational and Applied Mathematics, 230 (2009), pp. 781–802.

[116] K. Kunisch, K. Pieper and B. Vexler, *Measure valued directional sparsity for parabolic optimal control problems*, SIAM Journal on Control and Optimization, 52 (2014), pp. 3078–3108.

[117] K. Kunisch and A. Rösch, *Primal-dual active set strategy for a general class of constrained optimal control problems*, SIAM Journal on Optimization, 13 (2002), pp. 321–334.

[118] E. Laitinen, A. Lapin and S. Lapin, *Iterative solution methods for variational inequalities with nonlinear main operator and constraints to gradient of solution*, Lobachevskii Journal of Mathematics, 33 (2012), pp. 341–352.

[119] A. Lapin, *Preconditioned Uzawa-type methods for finite-dimensional constrained saddle point problems*, Lobachevskii Journal of Mathematics, 31 (2010), pp. 309–322.

[120] D. Leykekhman, B. Vexler and D. Walter, *Numerical analysis of sparse initial data identification for parabolic problems*, ESAIM: Mathematical Modeling and Numerical Analysis, 54 (2020), pp. 1139–1180.

[121] S. Lenhart, *Optimal control of a convective-diffusive fluid problem*, Mathematical Models and Methods in Applied Sciences, 5 (1995), pp. 225–237.

[122] B. Li, J. Liu and M. Xiao, *A fast and stable preconditioned iterative method for optimal control problem of wave equations*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2508–A2534.

[123] Y. Li, S. Osher and R. Tsai, *Heat source identification based on ℓ^1 constrained minimization*, Inverse Problems & Imaging, 8 (2014), pp. 199–221.

[124] G. Li, Y. Tan, J. Cheng and X. Wang, *Determining magnitude of groundwater pollution sources by data compatibility analysis*, Inverse Problems in Science and Engineering, 14 (2006), pp. 287–300.

[125] J.L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations (Grundlehren der Mathematischen Wissenschaften)*, Vol. 170, Springer Berlin, 1971.

[126] J.L. Lions, *Exact controllability, stabilization and perturbations for distributed systems*, SIAM Review, 30 (1988), pp. 1–68.

[127] P.L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM Journal on Numerical Analysis, 16 (1979), pp. 964 979.

[128] W. Liu, *Mixing enhancement by optimal flow advection*, SIAM Journal on Control and Optimization, 47 (2008), pp. 624–638.

[129] J. Liu and J. W. Pearson, *Parameter-robust preconditioning for the optimal control of the wave equation*, Numerical Algorithms, 83 (2020), pp. 1171–1203.

[130] Y. Liu, X. Yuan, S. Zeng and J. Zhang, *Partial error bound conditions and the linear convergence rate of the alternating direction method of multipliers*, SIAM Journal on Numerical Analysis, 56 (2018), pp. 2095–2123.

[131] T.P. Mathew, M. Sarkis and C.E. Schaerer, *Analysis of block parareal preconditioners for parabolic optimal control problems*, SIAM Journal on Scientific Computing, 32 (2010), pp. 1180–1200.

[132] E. McDonald, *All-at-once solution of time-dependent PDE problems*, PhD thesis, University of Oxford, 2016.

[133] D. Meidner and B. Vexler, *A priori error estimates for space-time finite element discretization of parabolic optimal control problems part II: problems with control constraints*, SIAM Journal on Control and Optimization, 47 (2008), pp. 1301–1329.

[134] A. Monge and E. Zuazua, *Sparse source identification of linear diffusion-advection equations by adjoint methods*, Systems & Control Letters, 145 (2020), pp. 104801.

[135] A. Münch and E. Zuazua, *Numerical approximation of null controls for the heat equation through transmutation*, Inverse Problem, 26 (2010), pp. 085018.

[136] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer Science & Bussines Media, New York, 2004.

[137] M. Ng, F. Wang and X. Yuan, *Inexact alternating direction methods for image recovery*, SIAM Journal on Scientific Computing, 33 (2011), pp. 1643–1668.

[138] J. Nocedal and S.J. Wright, *Numerical Optimization*, Second Edition, Springer, 2006.

[139] B. Nour-Omid and P. Wriggers, *A two-level iteration method for solution of contact problems*, Computer Methods in Applied Mechanics and Engineering, 54 (1986), pp. 131–144.

[140] M. Olshanskii and A. Reusken, *Grad-div stablilization for Stokes equations*, Mathematics of Computation, 73 (2004), pp. 1699–1718.

[141] C.C. Paige and M.A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629.

[142] D.W. Peaceman and H.H. Rachford, Jr. *The numerical solution of parabolic and elliptic differential equations*, Journal of the Society for Industrial and Applied Mathematics, 3 (1955), pp. 28–41.

[143] J.W. Pearson and J. Gondzio, *Fast interior point solution of quadratic programming problems arising from PDE-constrained optimization*, Numerische Mathematik, 137 (2017), pp. 959–999.

[144] J.W. Pearson, M. Stoll and A.J. Wathen, *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 1126–1152.

[145] J.W. Pearson and A.J. Wathen, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, Numerical Linear Algebra with Applications, 19 (2012), pp. 816–829.

[146] J.W. Pearson and A.J. Wathen, *Fast iterative solvers for convection-diffusion control problems*, Electronic Transactions on Numerical Analysis, 40 (2013), pp. 294–310.

[147] M. Porcelli, V. Simoncini and M. Tani, *Preconditioning of active-set Newton methods for PDE-constrained optimal control problems*, SIAM Journal on Scientific Computing, 37 (2015), pp. S472–S502.

[148] M.J. Powell, *A method for nonlinear constraints in minimization problems*, Optimization, (1969), pp. 283–298.

[149] M. Prato and L. Zanni, *Inverse problems in machine learning: an application to brain activity interpretation*, Journal of Physics: Conference Series 135 (2008), 012085.

[150] W. Queck, *The convergence factor of preconditioned algorithms of the Arrow-Hurwicz type*, SIAM Journal on Numerical Analysis, 26 (1989), pp. 1016–1030.

[151] R. A. Raviart and J. M. Thomas, *A mixed finite element method for 2nd order elliptic problems*, In Mathematical Aspects of Finite Element Methods, I. Galligani and E. Magenes (eds.), Lecture Notes in Math. 606, Springer-Verlag, New York, 1977, pp. 292–315.

[152] M.P. Robichaud, P.A. Tanguy and M. Fortin, *An iterative implementation of the Uzawa algorithm for 3-d fluid flow problems*, International Journal for Numerical Methods in Fluids, 10 (1990), pp. 429–442.

[153] R.T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898.

[154] R.T. Rockafellar and R. Wets, *Variational Analysis*, Springer Science & Business Media, 2009.

[155] A. Rösch and B. Vexler, *Optimal control of the Stokes equations: A priori error analysis for finite element discretization with postprocessing,* SIAM Journal on Numerical Analysis, 44 (2006), pp. 1903–1920.

[156] T. Rusten and R. Winther, *A preconditioned iterative method for saddle point problems*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 887–904.

[157] B. Rynne and M.A. Youngson, *Linear Functional Analysis*, Springer Science & Business Media, 2013.

[158] Y. Saad, *Iterative Methods for Sparse Linear Systems*, vol. 82, SIAM, 2003.

[159] M.A. Saunders, *Cholesky-based methods for sparse least squares: the benefits of regularization.* Linear and nonlinear conjugate gradient-related methods 100 (1996), pp: 92–100.

[160] A. Schiela and S. Ulbrich, *Operator preconditioning for a class of inequality constrained optimal control problems*, SIAM Journal on Optimization, 24 (2014), pp. 435–466.

[161] A. Schindele and A. Borzì, *Proximal schemes for parabolic optimal control problems with sparsity promoting cost functionals*, International Journal of Control, 90 (2017), pp. 2349–2367.

[162] X. Song and B. Yu, *A two-phase strategy for control constrained elliptic optimal control problems*, Numerical Linear Algebra with Applications, (2018), pp. e2138.

[163] Y. Song, X. Yuan and H. Yue, *Implementation of the ADMM to parabolic optimal control problems with control constraints and beyond*, arXiv preprint, arXiv:2005.01582, (2020).

[164] G. Stadler, *Elliptic optimal control problems with L^1-control cost and applications for the placement of control devices*, Computational Optimization and Applications, 44 (2009), pp. 159.

[165] M. Stoll, *One-shot solution of a time-dependent time-periodic PDE-constrained optimization problem*, IMA Journal of Numerical Analysis, 34 (2013), pp. 1554–1577.

[166] M. Stoll and A. Wathen, *Preconditioning for partial differential equation constrained optimization with control constraints*, Numerical Linear Algebra with Applications 19 (2012), pp. 53–71.

[167] W.C. Thacker, *Oceanographic inverse problems*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 16–37.

[168] W. Tian and X. Yuan, *Linearized primal-dual methods for linear inverse problems with total variation regularization and finite element discretization*, Inverse Problems, 32 (2016), pp. 115011.

[169] W. Tian and X. Yuan, *Convergence analysis of primal-dual based methods for total variation minimization with finite element approximation*, Journal of Scientific Computing, 76 (2018), pp. 243–274.

[170] W. Tian and X. Yuan, *An accelerated primal-dual iterative scheme for the L^2-TV regularized model of linear inverse problems*, Inverse Problems, 35 (2019), pp. 035002.

[171] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods, and Applications*, vol. 112, American Mathematical Society, 2010.

[172] U. Trottenberg, C. Oosterlee and A. Schüller, *Multigrid*, Elsevier, 1987.

[173] P. Tseng, *Applications of a splitting algorithm to decomposition in convex programming and variational inequalities*, SIAM Journal on Control and Optimization, 29 (1991), pp. 119–138.

[174] M. Ulbrich, *Semismooth newton methods for operator equations in function spaces*, SIAM Journal on Optimization, 13 (2002), pp. 805–841.

[175] M. Ulbrich, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems in Function Spaces*, vol. 11, SIAM, 2011.

[176] S. Ulbrich, *Generalized SQP methods with parareal time-domain decomposition for time-dependent PDE-constrained optimization*, Real-time PDE-constrained optimization, SIAM, 2007, pp. 145–168.

[177] S. Ulbrich, *Preconditioners based on parareal time-domain decomposition for time-dependent PDE-constrained optimization*, Multiple Shooting and Time Domain Decomposition Methods, Springer, 2015, pp. 203–232.

[178] T. Valkonen, *A primal–dual hybrid gradient method for nonlinear operators with applications to MRI*, Inverse Problems, 30 (2014), pp. 055012.

[179] H.A. Van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, vol. 13, Cambridge University Press, 2003.

[180] G. Wachsmuth and D. Wachsmuth, *Convergence and regularization results for optimal control problems with sparsity functional*, ESAIM: Control, Optimisation and Calculus of Variations, 17 (2011), pp. 858–886.

[181] C. Wagner, *Introduction to algebraic multigrid*, Course Notes of an Algebraic Multigrid Course, University of Heidelberg, 1999.

[182] X. Wang, J.J. Ye, X. Yuan, S. Zeng and J. Zhang, *Perturbation techniques for convergence analysis of proximal gradient method and other first-order algorithms via variational analysis*, Set-Valued and Variational Analysis, (2021), pp. 1–41.

[183] X. Wang and X. Yuan, *The linearized alternating direction method of multipliers for dantzig selector*, SIAM Journal on Scientific Computing, 34 (2012), pp. A2792–A2811.

[184] A.J. Wathen, *Realistic eigenvalue bounds for the Galerkin mass matrix*, IMA Journal of Numerical Analysis, 7 (1987), pp. 449–457.

[185] J. Xu and L. Zikatanov, *Algebraic multigrid methods*, Acta Numerica, 26 (2017), pp. 591–721.

[186] J. Yang and X. Yuan, *Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization*, Mathematics of Computation, 82 (2013), pp. 301–329.

[187] J.J. Ye and X.Y. Ye, *Necessary optimality conditions for optimization problems with variational inequality constraints*, Mathematics of Operations Research, 22 (1997), pp. 977–997.

[188] J.J. Ye, X. Yuan, S. Zeng and J. Zhang, *Variational analysis perspective on linear convergence of some first order methods for nonsmooth convex optimization problems*, optimization-online, 2018.

[189] X. Yuan, *The improvement with relative errors of He et al.'s inexact alternating direction method for monotone variational inequalities*, Mathematical and Computer Modelling, 42 (2005), pp. 1225–1236.

[190] H. Yue, Q. Yang, X. Wang and X. Yuan, *Implementing the alternating direction method of multipliers for big datasets: A case study of least absolute shrinkage and selection operator*, SIAM Journal on Scientific Computing, 40 (2018), pp. A3121–A3156.

[191] X. Zhang, M. Burger, X. Bresson and S. Osher, *Bregmanized nonlocal regularization for deconvolution and sparse reconstruction*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 253–276.

[192] X. Zhang, M. Burger, and S. Osher, *A unified primal-dual algorithm framework based on bregman iteration*, Journal of Scientific Computing, 46 (2011), pp. 20–46.

[193] M. Zhu and T. Chan, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration*, UCLA CAM Report, 34, 2008.

[194] E. Zuazua, *Propagation, observation, and control of waves approximated by finite difference methods*, SIAM Review, 47 (2005), pp. 197–243.

[195] E. Zuazua, *Controllability of Partial Differential Equations*, 3rd cycle, Castro Urdiales (Espagne), 2006, pp.311. cel-00392196.

[196] E. Zuazua, *Controllability and observability of partial differential equations: some results and open problems*, Handbook of differential equations: evolutionary equations, vol. 3, North-Holland, 2007, pp. 527–621.

[197] E. Zuazua, *Numerics for the control of partial differential equations*, In Encyclopedia of Applied and Computational Mathematics, B. Engquist (eds.), Springer, Berlin, Heidelberg, 2015.

Curriculum Vitae

Education:

2018 – present	The University of Hong Kong, Hong Kong, China, Ph.D Candidate.
2016 – 2018	Hong Kong Baptist University, Hong Kong, China, M.Phil.
2013 – 2016	Jilin University, Jilin, China, M.Phil.
2009 – 2013	Jilin University, Jilin, China, B.Sc.

Publications:

1. U. Biccari, **Y. Song**, X. Yuan, and E. Zuazua.
   "An optimal control based two-stage numerical approach for the sparse initial source identification of diffusion-advection equations", preprint.

2. R. Glowinski, **Y. Song**, X. Yuan, and H. Yue.
   "Bilinear optimal control of an advection-reaction-diffusion system", SIAM Review, to appear.

3. S. Gong, **Y. Song**, X. Yuan, and H. Yue.
   "Fast primal dual hybrid gradient methods for a general class of optimal control problems with PDE constraints", preprint.

4. R. Glowinski, **Y. Song**, X. Yuan, and H. Yue.
   "Implementation of the ADMM to parabolic optimal control problems with control constraints and beyond", preprint.

5. Y. Gao, J. Li, **Y. Song**, C. Wang, and K. Zhang.
"Alternating direction based method for optimal control problem constrained by Stokes equation", accepted by Journal of Inverse and Ill-posed Problems, (2020).

6. R. Glowinski, **Y. Song**, and X. Yuan.
"An ADMM numerical approach to linear parabolic state constrained optimal control problems", Numerische Mathematik, 144 (2020), pp. 931–966.

7. **Y. Song**, X. Yuan, and H. Yue.
"An inexact Uzawa algorithmic framework for nonlinear saddle point problems with applications to elliptic optimal control problem", SIAM Journal on Numerical Analysis, 57 (2019), pp. 2656–2684.

8. K. Zhang, J. Li , **Y. Song**, and X. Wang.
"An alternating direction method of multipliers for elliptic equation constrained optimization problem", Science China Mathematics, 60 (2017), pp. 1–18.