ON THE CONTROL OF SYSTEMS MODELLED BY PARTIAL DIFFERENTIAL EQUATIONS

A Dissertation Presented to
the Faculty of the Department of Mathematics
University of Houston

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

By
Rahul Kumar
Fall 2016

ON THE CONTROL OF SYSTEMS MODELLED BY PARTIAL DIFFERENTIAL EQUATIONS

Rahul Kumar

APPROVED:

Roland Glowinski, Chairman
Dept. of mathematics

Tsorng-Whay Pan
Dept. of Mathematics

Annalisa Quaini
Dept. of Mathematics

LieJune Shiau
Dept. of Mathematics and Statistics
University of Houston-Clear Lake

Dean, College of Natural Sciences and Mathematics

Acknowledgements

I would like to express my gratitude

To Dr. Roland Glowinski for his loving patience through the vicissitude of the creative process. My knowledge belongs to him.

To Dr. Dinesh Singh and all faculty at Mathematical Sciences Foundation, New Delhi for giving me chance to cultivate my mathematical acumen.

To IMA, University of Minnessota for making me long time visitor for the conference on control theory which helped me in writing this thesis.

To my mother, father, brother and sister for their love and inspiration.

I would like to thank the committee members of my PhD dissertation for their time and effort. I have had the extreme fortune of being guided by brilliant professors at University of Houston. My very respectful list of directly interacting professors includes Dr Ronald Hoppe, Dr Maxim Olshanshki, Dr Yuri Kuznetsov and Dr Daniel Onofrei. I specially thank Dr Daniel Onofrei for writing recommendation letters for me whenever I needed.

I want to thank John Haas and Andrew Faldt for being valuable friends during my university days, being my bike buddies for summer 2014 and for all mathematical and non mathematical discussions.

I wish to thank all my friends, my roommates over the years Siddhartha Jana, A. Agrawal, Satish Pandey, Eric Emeca and Deepak for all the support they gave from the moment I stepped in to Houston to the moment I will leave.

I also wish to thank several other wonderful people including Pooran Negi, Chandi Bhandari and their family. They are wonderful people who have graciously opened their doors for my visit, with whom I have had intellectual or personal conversations during the last four years- the list is certainly too long and I risk missing many names, but if you are reading this you are probably one of them!

Last, I offer my salutation to the lotus feet of my guru Swami Sadafaldeo Ji Maharaj for being with me virtually and guiding me all the way. I thank all the member of NAIVY (North America Institute of Vihangam yoga) for the Wednesday's meetings.

कर्मण्येवाधिकारस्ते मा फलेषु कदाचन।
मा कर्मफलहेतुर्भुर्मा ते सङ्गोऽस्त्वकर्मणि॥

ON THE CONTROL OF SYSTEMS MODELLED BY PARTIAL DIFFERENTIAL EQUATIONS

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Mathematics

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

By

Rahul Kumar

Fall 2016

Abstract

This work aims to discuss two kinds of numerical problems, namely the control of diffusion phenomenon on the surface of manifolds and the control of problems modelled by parabolic variational inequalities of the obstacle type.

The manifold problems treated are the control of diffusion phenomenon on the surface of a torus and sphere in $\mathbb{R}^3$. Discretization are constructed using finite element in space and an implicit integration scheme in time. The calculation of the optimal control was performed in two ways. The first approach discusses the control with mapping it to a standard domain and imposing periodic boundary condition over the mesh geometry for solving the state equation. The second approach is rather a direct approach which involves working with the mesh itself; it handles general two-dimensional manifolds in three-dimensional space. The solution of the state equation is numerically approximated using an isoparametric finite element method.

The control problem associated with the parabolic variational inequalities is transformed into a control problem with the state equation as a nonlinear parabolic equation using penalization. Adjoint equation techniques are employed to compute the optimal control for the obstacle type problem and a conjugate gradient algorithm is used to solve the non-linear minimization problem that appears. Optimality conditions for the control problem were found using perturbation analysis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The past two decades have seen an explosion of algorithms and models, opening up new fields of research within control theory and optimization. However, much is still left to be said about most classical settings in control theory. Such problems consist of optimizing an objective functional, e.g. maximizing the performance or minimizing the cost of a system subject to some conditions given by the laws of physics: for example, an aeronautical engineer will want to choose the best shape for a wing to optimize its performance ([37], [12], [20] and [22]). Inverse problems may also be treated as control problem where the goal is to infer some unobservable information from observable evidence ([10] and [21]). Optimal control of systems modelled by partial differential equations are known by epithets like *optimal control* problems, *inverse* problems, *parameter estimation*, and *optimal design* problems. The purpose of this introductory chapter is to give an overview of the topics treated in this thesis. We will give a motivation for the different numerical experiments performed and also briefly discuss the theories and ideas behind the models and methods to be used.

We are interested in numerical methods for the simulation and control of systems modelled by partial differential equations. The broad area we will explore in this dissertation covers:

- The control of diffusion phenomenon on manifolds.
- The control of distributed parameter systems modelled by parabolic variational inequalities.

These topics span the disciplines of partial differential equations, optimization, and control theory. Basically different numerical experiments have been performed. The first experiment aim to simulate the control of a diffusion phenomenon on a circle which we consider as a toy problem for the control of diffusion processes on manifolds. Finally, our interest will be in the application to the optimal control of distributed-parameter systems modelled by parabolic variational inequalities of the obstacle type. In the following sections, we will give a small overview of the problems presented in this thesis. We will not go into any greater detail but try to paint a large picture.

All the mathematical formulations we are going to consider are a-dimensional, a classical approach for the control of partial differential equations, popularized by J.L. Lions (see, e.g., [27]) (J.L. Lions is considered as the founder of this branch of mathematics concerned with the control of systems modelled by partial differential equations)

1.1 Control of diffusion phenomena on the surface of a torus

Diffusion problems are common when studying chemical reactions and heat transfer. For example, in controlling an unstable chemical reaction we may want to reduce the heat produced to avoid an explosion. Diffusion problems are also related to problems in fluid dynamics, and extension to our work could be of interest in studying the control of advection-diffusion phenomena on manifolds which is of particular importance in studying the identification of pollution sources [38]. This problem can be also be considered as an inverse source problems ([4] and [5]). In our work, we implement the classical approach discussed in [13], [9], and [28].

Variatonal formulation of the control problem:

Find $v \in L^2(\Sigma \times (0,T))$ such that the solution y of the following variational problem

$$\begin{cases} (\frac{\partial y}{\partial t}(t), z) + \mu(\nabla_\Sigma y(t), \nabla_\Sigma z) = \int_\omega v(t) z d\Sigma, \ \ \forall z \in H^1(\Sigma), \ \ a.e \ \ on \ \ (0,T), \\ y(0) = y_0, \end{cases}$$

$$satisfies$$

$$y(T) = y_T \qquad .$$

where

- ∇_Σ is the tangential gradient on Σ,
- $d\Sigma$ is the infinitesimal superficial(surfacic) measure,
- $H^1(\Sigma) = \{z : z \in L^2(\Sigma), \int_\Sigma |\nabla_\Sigma v|^2 d\Sigma < +\infty\}$

Unfortunately, this *Exact controllability* is possible only for a very restricted class of target functions, y_T. We formulate therefore the problem as an approximate controllability problem using a penalty argument ([13], [9], and [28]). To discuss the control of diffusion phenomena on manifolds, we first control the diffusion phenomena on the circle and use that as a toy problem. For the toy problem, we formulate the control problem on a unit circle and map it to an unit interval imposing periodic boundary condition at $x = 0$ and $x = 1$. Next, we attempt to solve the control problem on manifolds. Control problems on manifolds are fairly complicated problems from the analytical and computational point of view. To achieve this goal, we employ a methodology combining finite differences for the time discretization and finite elements for the space approximation. Controlling on the whole domain is not realistic in practice, so we seek to implement the control on small subsets of the domain a (vertical or horizontal strips in the case of a torus). Then we analyze convergence rates and controllability errors for these control domains.

1.2 Control of diffusion phenomena on manifolds (Torus and Sphere)

In Chapter 4 we will discuss an alternative approach to the one we use in Chapter 2 (Control of diffusion phenomena on the surface of a torus) by working directly with the mesh and implementing an optimization algorithm. The solution of the state equation which is a partial differential equation over the immersed manifolds is numerically approximated using an iso-parametric finite element method described in [39]. We work with already built meshes for torus and sphere. This experiment is motivated from the article by Bonito and Glowinski [7]. Also, we will present the finite element assembly details on manifold which serves the purpose of skipping related assembly details in Chapter 2 and Chapter 3. Finally, we will report our numerical results for different test cases . We will also report controllability errors and the time evolution of the L^2 norm of the controls obtained using the approach we sketched above.

1.3 Control of distributed parameter system modelled by parabolic variational inequalities of the obstacle type

In the last chapter we will study the control of distributed parameter systems modelled by parabolic variational inequalities of the obstacle type. This is a quite widely investigated problem(particularly in Germany) in recent years using interior penalty approach, see for instance ([18], [19], and [16]). Our goal here is (much) more modest: we want to discuss the numerical solution of control problems associated with parabolic inequalities of the obstacle type, using an exterior-penalty technique. Using this penalty approach, we will be able to approximate the parabolic variational inequalties by nonlinear parabolic equations and then apply to the related control problem the (fairly classical methods) discussed in [13].

The motivation of this topic is inspired by many interesting applications including, to name a few, Tripology and non-Newtonian fluid mechanics (visco-plasticity). The Tripology is the study in science and engineering of interacting surfaces in relative motion. It includes the study and application of the principles of friction, lubrication, and wear which is quite an important branch of material science and mechanical engineering.

1.4 Implementation in FEniCS

We will give a precise summary of FEniCS (python-based package) as explained in [3] for the numerical simulations of all the model problems. The FEniCS project is a collection of open-source software for the automation of mathematical problems based on differential equations. The most important aspect of FEniCS is the Unified Form Language (UFL, see Alnaes et al. (2013)) which allows the generation of assembly routines by providing only the variational equation. This is an enormous saving of human resources which would be spent if one implements and tests the assembly routines for bilinear forms by hand. Another big advantage of UFL is the possible use of automatic differentiation (AD) in order to avoid the manipulation of long formulas which are associated with errors in calculation and implementation. Besides that, FEniCS offers a wide range of high-end routines that allow the user to solve differential equations quite easily in few lines of code. The state equation in all our model problem is given in terms of a variational equation which can be easily fed into FEniCS. We will give a brief overview of the relatively easy coding style for our control problem. We will also give an overview of recent development of dolfin-adjoint package built on the top of FEniCS which is used in creating the tape out of the forward problem (state equation) and use that tape to derive the adjoint. We will discuss the optimization framework of dolfin adjoint in Chapter 4.

Chapter 2

Control of a diffusion phenomenon on a circle

Our goal in this chapter is to discuss the control of diffusion phenomena taking place on a time interval $(0, T)$ and a circle C using an actuator located on an open set ω (not necessarily connected) of C. From a practical point of view, we will map C to the interval $[0, 1)$ and impose periodic conditions at $x = 0$ and $x = 1$. First we will introduce the model and describe the solution method. In section 2.5 and 2.6, we will discuss the discretizations used not only in this chapter but also in chapter 3 and 5. Finally, we will present numerical results.

2.1 Introduction

The motivation for this study is the control of diffusion phenomena on manifolds particularly on the torus and the sphere. Ocean and atmosphere simulations crucially rely on the capability to solve equations over the sphere. The model presented in this chapter will serve as a toy model for the control of diffusion phenomena on manifolds. Schematically, a control problem consists of:

- An input-output process (Control system)
- Observations of the output of the controlled system
- An objective to be achieved

The controlled system is described by the following state equation

$$\frac{\partial y}{\partial t} + \mathcal{A}(y) = f + \mathcal{B}v \tag{2.1}$$

where in (2.1):

- y is the state function (a temperature, for example).
- $\mathcal{A}$ is the partial differential operator and f is a forcing term
- The control v belongs to a suitable control space $\mathcal{U}$, $\mathcal{B}$ being a linear operator acting on $\mathcal{U}$.

2.2 Problem formulation

The control problem which we want to solve computationally reads as

$$\begin{cases} \text{Find } u \in \mathcal{U} \text{ such that} \\ J(u) \leq J(v) \qquad\qquad \forall v \in \mathcal{U} \end{cases} \tag{2.2}$$

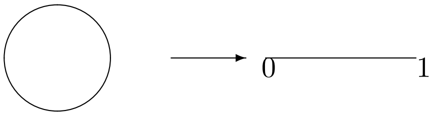


Figure 2.1: Mapping from circle to unit interval

where in (2.2)

$$\mathcal{U} = L^2(\omega \times (0,T)) \tag{2.3}$$

and (with $k > 0$)

$$J(v) = \frac{1}{2} \int\limits_{\omega\times(0,T)} |v(x,t)|^2 dxdt + \frac{k}{2} \int\limits_0^1 |y(x,T) - y_T(x)|^2 dx \tag{2.4}$$

The function y in (2.4) is the solution of the following parabolic problem (with $\mu > 0$).

$$\begin{cases} \frac{\partial y}{\partial t} - \mu \frac{\partial^2 y}{\partial x^2} = v\chi_\omega \ in \ \ (0,1) \times (0,T) \\ y(0,t) = y(1,t), \ \frac{\partial y}{\partial x}(0,t) = \frac{\partial y}{\partial x}(1,t), 0 < t < T, \\ y(0) = y_0. \end{cases} \tag{2.5}$$

In (2.4), (2.5) : (a) χ_ω is the characteristic function of the set ω, (b) y_T is the target function and (c) $y(t)$ denotes the function $x \to y(x,t)$. We will also use the following notation.

$$\begin{cases} (f,g) = \int\limits_0^1 f(x)g(x)dx, \forall f,g \in L^2(0,1), \\ ||f||_{L^2(0,1)} = (f,f)^{\frac{1}{2}}, \forall f \in L^2(0,1), \\ ||v||_{\mathcal{U}} = \sqrt{\int\limits_{\omega\times(0,T)} |v(x,t)|^2 dxdt}, \ \forall v \in \mathcal{U} \end{cases} \tag{2.6}$$

The closeness to the target y_T will be forced in a least square sense by penalty, k being the penalty parameter. The value of k determines the relative importance between the cost of the control and the distance to the target.

2.3 Generalities and synopsis

Our goals in the following sections are:

1. To describe a conjugate gradient algorithm operating in the space $\mathcal{U}$ for the solution of problem (2.2).
2. To investigate a space-time discretization of problem (2.2) obtained by combining a backward Euler scheme for the time discretization with a finite element approximation for the space discretization.
3. To use a discrete analogue of the algorithm in step (1) to solve the fully discrete control problem.

Now, a *variational formulation* of the state equation is given by

$$\begin{cases} y(t) \in V_0 \; a.e \; on \; (0,T), \\ (y_t, z) + \mu(y_x, z_x) = \int\limits_{\omega} vzdx, \forall z \in V_0, \\ y(0) = y_0, \end{cases} \tag{2.7}$$

where in (2.7)

$$V_0 = \{z | z \in H^1(0,1), z(0) = z(1)\}. \tag{2.8}$$

Given the target function y_T in $L^2(0,1)$, our goal is to find a control v so that the associated value of $y(T)$ is close to y_T at a minimal cost.

Remark 2.3.1. For the sake of rigour, we should replace, in (2.7), (y_t, z) by $\langle y_t, z\rangle$, where$\langle ., .\rangle$ denotes the duality pairing between V_0' (the dual space of V_0) and V_0

2.4 Gradient calculation

Most minimization algorithms use information on the gradient of the cost function. In the particular case of the functional (2.4), deriving DJ is easy. Indeed, this can be achieved via a very simple perturbation analysis. Let δv be a perturbation of v: we have then

$$\delta J(v) = \int\limits_{\omega\times(0,T)} DJ(v)\delta v dxdt = \int\limits_{\omega\times(0,T)} v\delta v dxdt + k\int\limits_0^1 (y(T) - y_T)\delta y(T)dx. \tag{2.9}$$

From(2.9), the perturbation δy induced by δv verifies

$$\begin{cases} \delta y(t) \in V_0 \; a.e \; on \; (0,T), \\ (\delta y_t, z) + \mu(\delta y_x, z_x) = \int\limits_\omega \delta v(t) z dx, \forall z \in V_0, \\ \delta y(0) = 0. \end{cases} \tag{2.10}$$

We introduce now a 'smooth' function p of x and t such that $p(t) \in V_0$, $\forall t \in (0,T)$. It follows then from (2.10) that

$$(\delta y_t, p(t)) + \mu(\delta y_x, p_x(t)) = \int\limits_\omega \delta v(t) p(t) dx \;\; , a.e \;\; on \;\; (0,T). \tag{2.11}$$

Integrating by parts equation (2.11) over (0, T), we obtain

$$\begin{cases} \int\limits_0^1 p(T)\delta y(T)dx - \int\limits_0^1 p(0)\delta y(0)dx - \int\limits_{(0,1)\times(0,T)} p_t \delta y dxdt + \\ \mu \int\limits_0^T dt \int\limits_0^1 p_x \delta y_x dx = \int\limits_{\omega\times(0,T)} \delta v p dxdt. \end{cases} \tag{2.12}$$

Let us assume now that p is the solution(necessarily unique) of the following *backward in time* initial value problem (the adjoint equation):

$$\begin{cases} p(t) \in V_0 \text{ , } a.e \text{ } on \text{ } (0,T), \\ -(p_t, z) + \mu(p_x, z_x) = 0 \, \forall \, z \in V_0, \\ p(T) = k(y(T) - y_T). \end{cases} \tag{2.13}$$

It follows then from (2.9), (2.10), (2.12) and (2.13) that

$$\int\limits_{\omega\times(0,T)} DJ(v)\delta v dxdt = \int\limits_{\omega\times(0,T)} (v+p)\delta v dxdt \tag{2.14}$$

i.e

$$\boxed{DJ(v) = v + p|_{\omega\times(0,T)}} \tag{2.15}$$

Remark 2.4.1. If large values of the parameter k are used to force the closeness of $y(T)$ and y_T, the minimization problem (2.2) may be badly conditioned, requiring an accurate expression of DJ. The safest way to obtain it is to discretize first the control problem and then to derive the discrete analogue of the state equation (2.5). The direct discretization of the adjoint equation will produce, in general an approximation of DJ which is not a gradient, a fact which may have disastrous consequences on the convergence of the iterative methods that we intend to employ to solve problem (2.2).

2.5 The Semi Discrete Control problem

The Time Discretization is being considered first; we divide thus the interval $(0,T)$ into $N(> 1)$ sub-intervals of equal length $\triangle t = \frac{T}{N}$. We then approximate the control space $\mathcal{U}(= L^2(\omega \times (0,T)))$ by

$$\mathcal{U}^{\triangle t} = (L^2(\omega))^N \tag{2.16}$$

and the control problem (2.2) by

$$\begin{cases} \mathbf{u}^{\triangle t} \in \mathcal{U}^{\triangle t}, \\ J^{\triangle t}(\mathbf{u}^{\triangle t}) \leq J^{\triangle t}(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}^{\triangle t}, \end{cases} \tag{2.17}$$

where

$$J^{\triangle t}(\mathbf{v}) = \frac{1}{2}\triangle t \sum_{n=1}^{N} \int_{\omega} |v_n|^2 dx + \frac{k}{2} \int_{0}^{1} |y^N - y_T|^2 dx, \forall \mathbf{v} = (v_n)_{n=1}^{N} \in \mathcal{U}^{\triangle t}, \tag{2.18}$$

with y^N obtained from y_0 and $\mathbf{v}$ via the solution of the following discrete parabolic problem

$$\begin{cases} y^0 = y_0. \\ For\ n = 1, \ldots, N,\ (y^{n-1}, v^n) \rightarrow y^n\ via\ the\ solution\ of \\ \frac{y^n - y^{n-1}}{\triangle t} - \mu \frac{d^2 y^n}{dx^2} = v^n \chi_\omega\ in\ (0,1), \\ y^n(0) = y^n(1), \frac{dy^n}{dx}(0) = \frac{dy^n}{dx}(1). \end{cases} \tag{2.19}$$

A most important step still has to be addressed, namely the computation of $DJ^{\triangle t}$; assuming that $\mathcal{U}^{\triangle t}$ is equipped with following inner product

$$\boxed{(\mathbf{v}, \mathbf{w})_{\triangle t} = \triangle t \sum_{n=1}^{N} \int_{\omega} v^n w^n dx} \quad , \tag{2.20}$$

we can easily show (using perturbation analysis) that $DJ^{\triangle t}(v)$ is the element of $\mathcal{U}^{\triangle t}$ defined as follows

$$\boxed{DJ^{\triangle t}(\mathbf{v}) = \{v^n + p^n|_\omega\}_{n=1}^{N}} \tag{2.21}$$

with $\{p^n\}_{n=1}^{N+1}$ defined from $\mathbf{v}$ and $\{y^n\}_{n=0}^{N}$ via the solution of the following discrete adjoint equation

$$\begin{cases} p^{N+1} = k(y^N - y_T). \\ For\ n = N, \ldots, 1,\ \ p^{n+1} \rightarrow p^n\ via\ the\ \ solution\ of \\ \frac{p^n - p^{n+1}}{\triangle t} - \mu \frac{d^2 p^n}{dx^2} = 0\ \ in\ (0,1), \\ p^n(0) = p^n(1), \frac{dp^n}{dx}(0) = \frac{dp^n}{dx}(1). \end{cases} \tag{2.22}$$

2.6 Full discretization

For the space discretization, we introduce an integer I such that $h = \frac{1}{I}$ with $x_i = ih$ for $i = 0, \dots, I$. We denote by k_i, the interval $[x_{i-1}, x_i]$, $i = 1, \dots, I$, and we will approximate $H^1(\Omega)$ using finite element approximations, since they are well-suited to the fact that the state and co-state equations have been given directly in variational forms. We consider finite element partitions $\mathcal{K}_h$ of $\Omega = (0, 1)$ with the following properties:

- $\mathcal{K}_h$ is a finite collection of closed intervals k_i contained in Ω, with h denoting the common length of the interval k_i.
- $\Omega = \bigcup\limits_{k_i \in \mathcal{K}_h} \bar{k_i}$
- If $\mathcal{K}$ and $\mathcal{K}'$ belong to $\mathcal{K}_h$ with $\mathcal{K} \neq \mathcal{K}'$, we have either $\mathcal{K} \cap \mathcal{K}' = \emptyset$ or $\mathcal{K}$ and $\mathcal{K}'$ have only one point in common

We approximate then V_0 by

$$V_{0h} = \{z | z \in C^0[0,1], z|_{k_i} \in P_1, \forall k_i \in \mathcal{K}_h, i = 1, \dots, I, z(0) = z(1)\}, \tag{2.23}$$

where P_1 is the space of the one variable polynomials of degree ≤ 1. As a vector basis for V_{0h}, we consider $B_h = \{\phi_i\}_{i=0}^{I-1}$, where ϕ_i, , if $i = 1, \dots, I-1$, is the usual "hat" functions defined by

$$\begin{cases} \phi_i \in V_{0h} \\ \phi_i(x_j) = \delta_{ij} \ \forall \ j = 1, \dots, I-1 \end{cases} \tag{2.24}$$

and for $i = 0$, we have

$$\begin{cases} \phi_0(0) = \phi_0(1) = 1 \\ and \\ \phi_0(x_j) = 0 \;\; \forall j = 0, \dots, I \end{cases} \tag{2.25}$$

The discrete control problem considered in this section is

$$\begin{cases} \mathbf{u}_h^{\triangle t} \in V_{0h}, \\ J_h(\mathbf{u}_h^{\triangle t}) \leq J_h(v), \forall v \in \mathcal{U}_h^{\triangle t} \end{cases} \tag{2.26}$$

where

$$J_h(\mathbf{v}) = \frac{\triangle t}{2} \sum_{n=1}^{N} \int\limits_{\omega} |v^n|^2 dx + \frac{k}{2} \int\limits_{0}^{1} |y^N - y_T|^2 dx \tag{2.27}$$

and where y_h^n is obtained from the solution of the fully discretized state equation

$$\begin{cases} y_h^0 \in V_{0h}, \;\; such \;\; that \\ (y_h^0, \phi_i) = (y_0, \phi_i) \;, \forall i = 0, \dots, I-1; \end{cases} \tag{2.28}$$

$$\begin{cases} For \;\; n = 1, \dots, N, \\ y_h^n \in V_{0h}; \forall i = 0, \dots, I-1, we \;\; have \\ (\frac{y_h^n - y_h^{n-1}}{\triangle t}, \phi_i) + \mu(\frac{dy_h^n}{dx}, \frac{d\phi_i}{dx}) = (v^n \chi_\omega, \phi_i) \end{cases} \tag{2.29}$$

The fully discrete adjoint system corresponding to (2.22) is given by

$$\begin{cases} p_h^{N+1} \in V_{0h}, such \;\; that \\ (p_h^{N+1}, \phi_i) = k(y_h^N - y_T, \phi_i), \forall i = 0, \dots, I-1, \end{cases} \tag{2.30}$$

for $n = N, \dots, 1$, compute p_h^n from the solution of

$$\begin{cases} p_h^n \in V_{0h}; \forall i = 0, \dots, I-1, we \;\; have \\ (\frac{p_h^n - p_h^{n+1}}{\triangle t}, \phi_i) + \mu(\frac{dp_h^n}{dx}, \frac{d\phi_i}{dx}) = 0. \end{cases} \tag{2.31}$$

Then the derivative of $J_h^{\triangle t}$ at $\mathbf{v}$ in the $\mathbf{w}$ direction will be

$$\boxed{(DJ^{\triangle t}(\mathbf{v}),\mathbf{w})_{\triangle t} = \Delta t \sum_{n=1}^{n=N} \int_{\omega} (v_h^n + p_h^n|_{\omega}) w^n dx} \tag{2.32}$$

2.7 Implementation details

Description of the algorithm

Through out $(.,.)$ stands for $\int_{\Omega\times(0,T)} vwdxdt$

$$\mathbf{v}_0 \ \ is \ \ given \ \ in \ \ (\mathbf{u}_h^{\triangle t}) \ \ . \tag{2.33}$$

Assuming that y^0 is known, solve first:

$$\begin{cases} (y^0,\phi_i) = (y_{0h},\phi_i), \ \ \forall i = 0,\dots,I-1 \\ For \ \ n = 1,\dots,N \ \ solve \\ y_h^n \in V_{0h}, \\ (\frac{y_h^n - y_h^{n-1}}{\triangle t},\phi_i) + \mu(\frac{dy_h^n}{dx},\frac{d\phi_i}{dx}) = (v_0^n\chi_\omega,\phi_i), \forall i = 0,\dots,I \ \ . \end{cases} \tag{2.34}$$

Solve next

$$\begin{cases} p_h^{N+1} \in V_{0h}, \ \ such \ \ that \\ (p_h^{N+1},\phi_i) = k(y_h^N - y_{T_h},\phi_i), \ \ \forall i = 0,\dots,I-1 \end{cases} \tag{2.35}$$

For $n = N,\dots,1$, compute p_h^n from the solution of

$$\begin{cases} p_h^n \in V_{0h}; \forall i = 0,\dots,I-1, we \ \ have \\ (\frac{p_h^n - p_h^{n+1}}{\triangle t},\phi_i) + \mu(\frac{dp_h^n}{dx},\frac{d\phi_i}{dx}) = 0, \end{cases} \tag{2.36}$$

and compute

$$g_0 = \{v_0^n + p_0^n|_\omega\}_{n=1}^N \tag{2.37}$$

and set $w_0 = g_0$.

Then for $k \geq 0$, assuming that $\mathbf{u}_k$, $\mathbf{g}_k$ and $\mathbf{w}_k$ are known, the last two different from $\mathbf{0}$, compute $\mathbf{u}_{k+1}, \mathbf{p}_{k+1}$ and if necessary $\mathbf{w}_{k+1}$ as follows:

$$\bar{y}_k^0 = 0.$$

For $n = 1, \dots, N$, solve

$$\begin{cases} \bar{y_k}^n \in V_{0h}, \\ (\frac{\bar{y_k}^n - \bar{y_k}^{n-1}}{\triangle t}, \phi_i) + \mu(\frac{d\bar{y}_k^n}{dx}, \frac{d\phi_i}{dx}) = (w_k^n \chi_\omega, \phi_i), \forall i = 0, \dots, I-1. \end{cases} \tag{2.38}$$

Take $\bar{p_k}^{N+1} = k\bar{y}_h^N$ and for $n = N, \dots, 1$, solve

$$\begin{cases} \bar{p_k}^n \in V_{0h}, \\ (\frac{\bar{p_k}^n - \bar{p_k}^{n+1}}{\Delta t}, \phi_i) + \mu(\frac{d\bar{p}_k^n}{dx}, \frac{d\phi_i}{dx}) = 0, \forall i = 0, \dots, I-1. \end{cases} \tag{2.39}$$

Compute

$$\begin{cases} \bar{\mathbf{g}}_k = \{w_k^n + \bar{p}_k^n|_\omega\}_{n=1}^N, \\ \rho_k = \frac{(\mathbf{g}_k, \mathbf{g}_k)_{\Delta t}}{(\bar{\mathbf{g}}_k, \mathbf{w}_k)_{\Delta t}} \end{cases} \tag{2.40}$$

and then,

$$\begin{cases} \mathbf{u}_{k+1} = \mathbf{u}_k - \rho_k \mathbf{w}_k, \\ \mathbf{g}_{k+1} = \mathbf{g}_k - \rho_k \bar{\mathbf{g}_k} \ . \end{cases} \tag{2.41}$$

If $\frac{(\mathbf{g}_{k+1},\mathbf{g}_{k+1})}{(\mathbf{g}_0,\mathbf{g}_0)} \leq \epsilon^2$ take $\mathbf{u}_h^{\triangle t} = \mathbf{u}_{k+1}$, else compute

$$\begin{cases} \gamma_k = \frac{(\mathbf{g}_{k+1},\mathbf{g}_{k+1})_{\triangle t}}{(\mathbf{g}_k,\mathbf{g}_k)_{\triangle t}}, \\ and\ \ then \\ \mathbf{w}_{k+1} = \mathbf{g}_{k+1} + \gamma_k \mathbf{w}_k. \end{cases} \tag{2.42}$$

Do $k = k+1$ and return to step (2.35).

2.8 Numerical experiments

2.8.1 A first test problem

In the numerical experiments, the initial condition y_0, the target y_T, have been for simplicity replaced by piecewise linear approximation y_{0h}, y_{Th}. The state, y_h^n, and the adjoint state p_h^n are expanded in the basis ϕ_h, and the integrals obtained are computed using the Simpson rule. This gives a system of equations with a sparse matrix on the left hand side. At each step, solve routine in (FEniCS[39]) was called which solves the linear system using LU factorization. We choose the data mentioned in Table 2.1 to perform our numerical experiments and in Table 2.2 and Table 2.3, we will present the performance of the conjugate gradient algorithm. Our control space is $\mathcal{U} = L^2(\omega \times (0,T))$ and we choose $\omega = \Omega$ for the first experiment.

Our target funtion y_T for the first experiment is

$$y_T : x \to cos(2\pi x), \tag{2.43}$$

$y_0(x) = 0$ being our initial condition for the state equation. Our initial control is chosen as $v_0 = 0$. The computed state $y^c(T)$ obtained from the simulation as well as the difference

Physical Parameters	μ	1
	T	0.5
	Ω	(0,1)
Penalty parameter	k	$10^2, 10^4$
Time Discretization parameter	$\triangle t$	10^{-2}

Table 2.1: Data used in experiments to evaluate the performance.

between the target function y_T and the computed state at t= T are shown in Figure 2.2. We have shown also the evolution of the L^2 norm of the control. We will present numerical results for the control supported by the whole domain. We will present a more detailed description of the effect of the domain on the control behaviour in the next chapter but for now we will stick only to the simple case of $\omega = (0, 1)$.

Table 2.2 shows the effect of different values of the penalty parameter k on the control problem. CGIters denotes the number of iterations necessary to achieve convergence of the conjugate gradient algorithm for the tolerance $tol = 10^{-5}$. We choose relatively large tolerance for the algorithm since it does not have good convergence properties for small stopping tolerances. The Norm u^c is denoted by $\|u^c\|_{L^2(\omega\times(0,T))}$, the relative controllability error being defined by $\frac{\|y_T - y^c(T)\|_{L^2(\Omega)}}{\|y_T\|_{L^2(\Omega)}}$. Table 2.3 shows that the different values of the

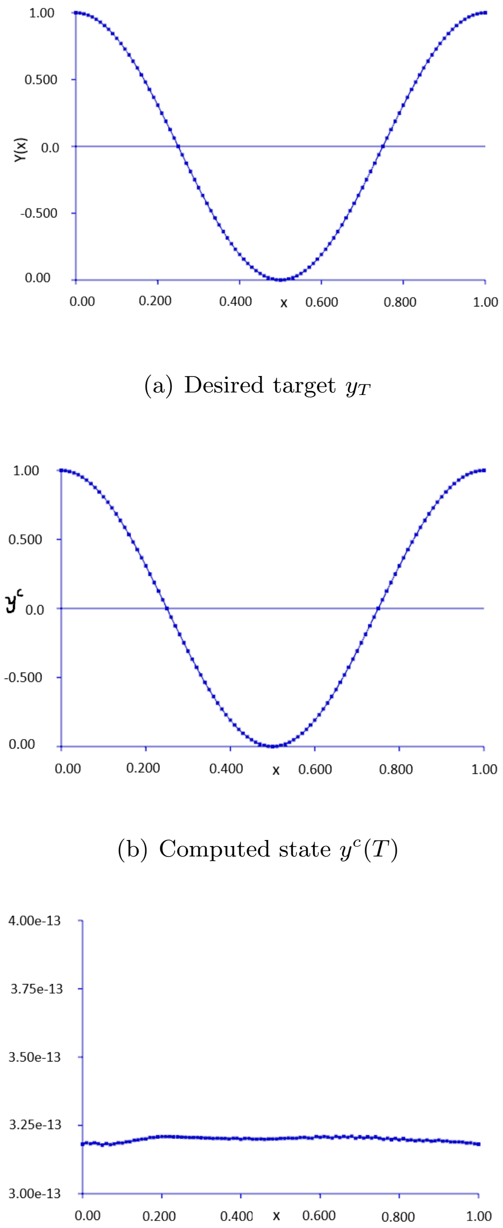
Table 2.2: Summary of convergence for $\triangle t = 0.01$, $T = 0.5$, h=$\frac{1}{64}$, and μ=1.

ω	k	CGIters	Norm u^c	Rel. Error
$(0, 1)$	10^2	5	6.8785	0.00086
	10^4	5	6.8785	0.00086

discretization parameter that we have considered lead to very close results. First we want to state that the numerical results do not depend on the discretization used. This suggests that the control-theoretic properties, like controllability and observability depend only on the location of sensors and actuators and not on the discretization used (for this test problem at least). There are no dramatic differences when the discretization is refined. This also suggests the stability behaviour of the algorithm for this test problem. The control we are using here is a robust feedback control. We observe in Table 2.3 that only Rel. Error has a significant change as we move from $h = \frac{1}{16}$ to $h = \frac{1}{128}$ which is desirable because of the increase in the number of discretization points.

Table 2.3: Summary of convergence for $T = 0.5$, $\mu = 1$, and $k = 10^4$.

h	CGiters	Norm u^c	Rel. Error
$\frac{1}{16}$	8	6.8835	0.01758
$\frac{1}{32}$	5	6.8772	0.01381
$\frac{1}{64}$	5	6.8758	0.00088
$\frac{1}{128}$	5	6.8755	0.00021



(a) Desired target y_T

(b) Computed state $y^c(T)$

(c) Difference between the computed state $y^c(T)$ and the desired target y_T

Figure 2.2: Visualization of the numerical results for $T = 0.5$, $\triangle t$=0.01, $h = \frac{1}{64}$, $k = 10^2$, and $\mu = 1$.

2.8.2 A second test problem

We choose for this second test problem the following target function

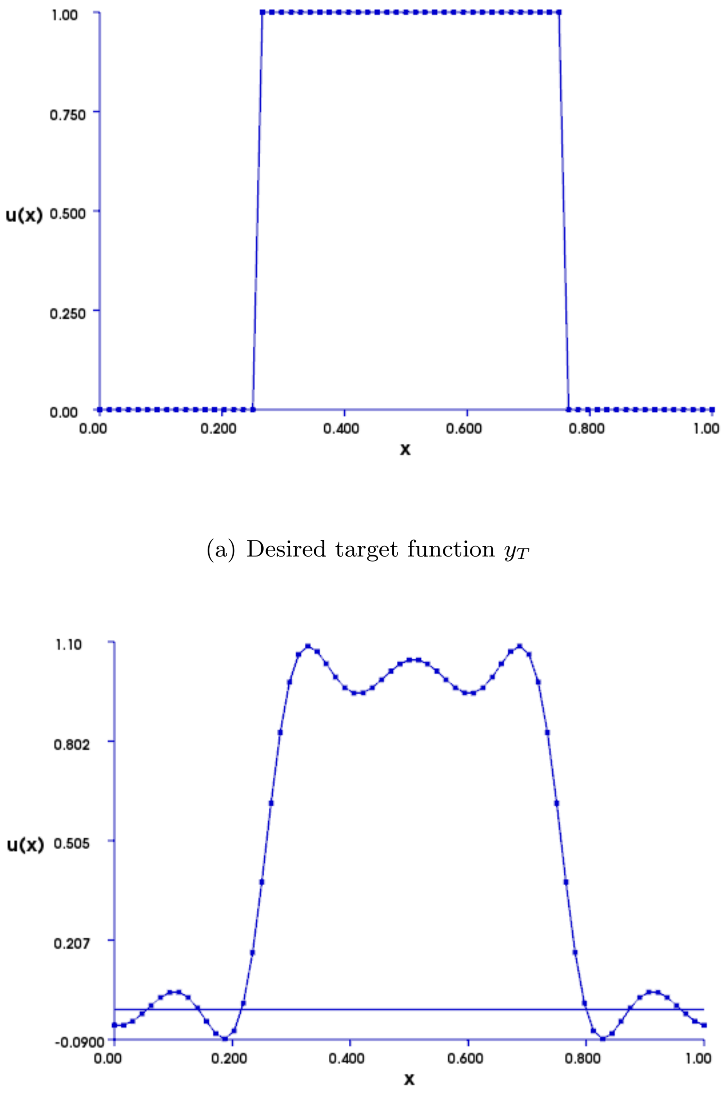
$$y_T(x) = \begin{cases} 1, & x \in [\frac{1}{4}, \frac{3}{4}] \\ 0, & \text{elsewhere} \end{cases} \tag{2.44}$$

with discontinuity at $x = \frac{1}{4}$ and $x = \frac{3}{4}$. We observe that our algorithm is not as efficient as in the first case and is not able to reconstruct the target function accurately near the points of discontinuity. Table 2.4 presents the summary of the convergence of the conjugate gradient algorithm. Here we analyse the convergence for different values of k. We observe that the convergence is better when the value of the penalty parameter is $k = 10^8$.

Table 2.4: Summary of convergence for T=0.5, μ= 1, $k = 10^4$, $\triangle t$= 0.01, and h=$\frac{1}{64}$.

k	CGIters	$\|u\|_{L^2(\omega\times(0,T))}$	Rel.error
10^2	28	14.9358	0.1890
10^4	29	81.7874	0.1709
10^6	29	200.225	0.1708
10^8	16	200.225	0.1688

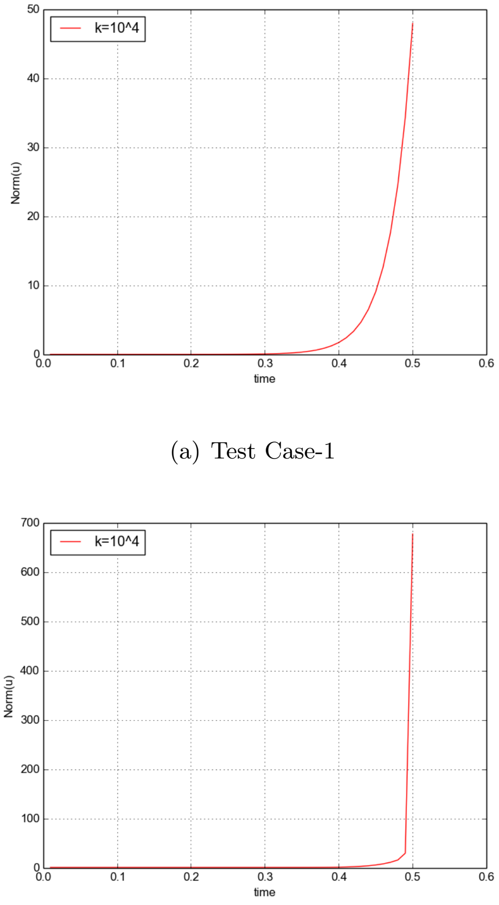
We have visualized on Figure 2.4, for $k = 10^4$ and $T = 0.5$, the evolution of the L^2 norm of the control for the first and second experiments i.e., the cost of the control required to achieve the desired target. We observe that the cost of the control starts from zero and increases as the time increases.



(a) Desired target function y_T

(b) Computed state $y^c(T)$

Figure 2.3: Visualization of the numerical results for T=0.5, $k = 10^2$, $\triangle t$=0.01, $\mu = 1$, and $h = \frac{1}{64}$.



(a) Test Case-1

(b) Test Case-2

Figure 2.4: Visualization of the L^2 norm of the optimal control for $T = 0.5$, $\triangle t$=0.01, $k = 10^4$, $\mu = 1$, and $h = \frac{1}{64}$.

Finally, we have visualized on Figure 2.5 the value of the cost function $J_h^{\triangle t}$, associated with the second test problem, versus the conjugate gradient iteration number. The algorithm starts with a zero value for the control i.e $\mathbf{v}_0 = \mathbf{0}$. The corresponding functional value $J_h^{\triangle t}(0)$ is 24.4791 which is close to the value of the functional in the continuous case. The final value of the objective functional is 0.3158.

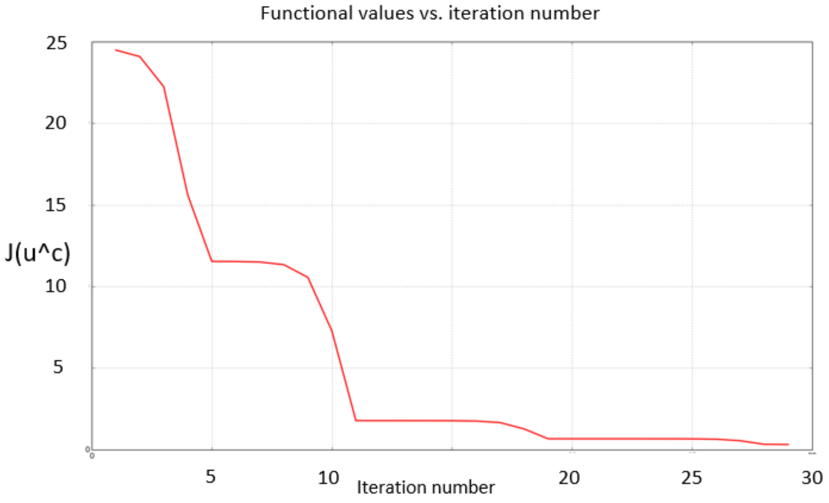


Figure 2.5: Functional values vs. iteration number for $T = 0.5$, $\triangle t= 0.01$, $k = 10^2$, $\mu = 1$, and $h = \frac{1}{64}$.

Chapter 3

Control of diffusion phenomena on the surface of a manifold

In this chapter, we will discuss the numerical solution of controllability problems associated with a diffusion process taking place on the surface of immersed manifolds (torus, sphere) in $\mathbb{R}^3$. To achieve this goal, we employ a methodology combining finite difference in time and finite element in space, and a conjugate gradient algorithm for the iterative solution of the discrete control problems. We will also investigate the null controllability properties of the diffusion model on these manifolds.

3.1 Introduction:

A large number of physical phenomena take place on surfaces. Many of these are modelled by partial differential equations, a typical example being provided by elastic shells. Differential equations posed over immersed manifolds are of particular importance in studying geophysical flows, for instance, since, ocean and atmosphere simulations crucially rely on the capability to solve such equations over a sphere.

3.2 A model problem

3.2.1 Generalities and synopsis

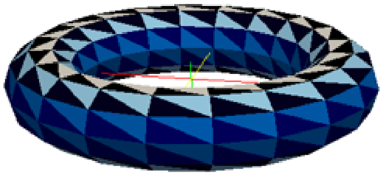


Figure 3.1: Surface of a torus

Our goals in the following sections are:

1. To transfer the ideas built in Chapter 2 to model the approximate controllability of diffusion phenomena on manifolds.

2. To discuss the optimality condition associated with the control problem.

3. To investigate a space-time discretization of the control problem by combining a backward Euler scheme for the time discretization with a finite element approximation for the space discretization.

4. To use a discrete analogue of the algorithm in Step 3 to solve the fully discrete control problem.

Let ω be an open subset of Σ. Our control problem is defined by

Find $v \in L^2(\Sigma \times (0,T))$ such that the solution y of the following variational problem

$$\begin{cases} (\frac{\partial y}{\partial t}(t), z) + \mu(\nabla_\Sigma y(t), \nabla_\Sigma z) = \int_\omega v(t) z d\Sigma, \ \ \forall z \in H^1(\Sigma), \ \ a.e \ \ on \ \ (0,T), \\ y(0) = y_0, \end{cases} \tag{3.1}$$

satisfies

$$y(T) = y_T. \tag{3.2}$$

called *exact controllability.* In general, the above problem has no solution. However, we have an approximate controllability because of the density of the set $\{y(T;v)\}_{v \in L^2(\Sigma\times(0,T))}$ in $L^2(\Sigma)$ which was proven in [13].

Above we have:

- ∇_Σ is the tangential gradient on Σ,
- $d\Sigma$ is the infinitesimal superficial (surfacic) measure,
- $H^1(\Sigma) = \{z : z \in L^2(\Sigma), \int_\Sigma |\nabla_\Sigma z|^2 d\Sigma < \infty\}$

The above controllability problem has no solution, in general. However, the density result provides us an approximately controllable variant of the above exact controllability problem.

Remark 3.2.1. Suppose that $y_T = 0$; it follows then from [32] that, $\forall y_0 \in L^2(\Sigma)$, there exists a control $v \in L^2(\Sigma \times (0,T))$, such that the solution of the initial value problem (3.1) verifies $y(T) = 0$. This property is called null controllability property and holds for sufficiently smooth surfaces of $\mathbb{R}^d$. We will also discuss the approximate null controllability properties of the diffusion model.

3.2.2 Approximate controllability problem

Taking advantage of the density results mentioned in Section 3.2.1, we approximate the exact controllability problem introduced above by the following one (of the approximation by penalty type).

$$\begin{cases} u_k \in \mathcal{U}, \\ J_k(u_k) \leq J_k(v), \ \ \forall v \in \mathcal{U}, \end{cases} \tag{3.3}$$

where:

a. $\mathcal{U}$ $=L^2(\Sigma \times (0,T))$.

b. The cost functional $J^k : \mathcal{U} \to R$ is defined by

$$J^k(v) = \frac{1}{2}\int_\omega |v|^2 d\Sigma dt + \frac{k}{2}\int_\Sigma |y(T) - y_T|^2 d\Sigma \ \ , \tag{3.4}$$

with $k > 0$ and the function y obtained from the control v via the solution of the initial value problem (3.1). It follows from, e.g., [13] that the approximate controllability problem (3.3) has a unique solution, characterized by

$$DJ_k(u_k) = 0 \tag{3.5}$$

where $DJ_k(u_k)$ is the differential of J_k at u_k.

Remark 3.2.2: Using results from [13], one can show the fact that if one denotes by y_k, the solution of (3.1) associated with u_k, then

$$\lim_{k\to\infty} y_k(T) = y_T \in L^2(\Sigma), \tag{3.6}$$

justifying thus taking (3.3) as approximate controllability problem.

3.3 Computation of $DJ_k(v)$: Optimality conditions

Using a similar methodology (perturbation analysis) as in Chapter 2, we get the differential of J_k as

$$\boxed{DJ_k(v) = \{v + p|_{\omega\times(0,T)}\}} \tag{3.7}$$

where p is the solution of the following adjoint equation

$$\begin{cases} -(\frac{\partial p}{\partial t}(t), z) + \mu(\nabla_\Sigma p(t), \nabla_\Sigma z) = 0 \ \ \forall z \in H^1(\Sigma), \ \ a.e \ \ on \ \ (0,T), \\ p(T) = k(y(T) - y_T). \end{cases} \tag{3.8}$$

Optimality conditions

Let u_k be the unique solution of the control problem (3.3) and denote by y_k and p_k the associated solution of (3.9) and (3.10), respectively. The control u_k will satisfy the following optimality system:

$$u_k + p_k|_{\omega\times(0,T)} = 0, \tag{3.9}$$

with

$$\begin{cases} (\frac{\partial y_k}{\partial t}, z) + \mu(\nabla y_k(t), \nabla z) = \int_\omega u_k(t) z d\Sigma, \forall z \in H^1(\Sigma), a.e \ \ on \ \ (0,T), \\ y_k(0) = y_0, \end{cases} \tag{3.10}$$

and

$$\begin{cases} -(\frac{\partial p_k}{\partial t}, z) + \mu(\nabla_\Sigma p_k(t), \nabla_\Sigma z) = 0, \forall z \in H^1(\Sigma), \ \ a.e \ \ on \ \ (0,T), \\ p_k(T) = k(y_k(T) - y_T). \end{cases} \tag{3.11}$$

3.4 Discretization of the control problem (3.3) and iterative solution of the fully discrete problem.

3.4.1 Time discretization of the control problem (3.3)

With N a positive integer we chose the time discretization step $\triangle t$ as $\triangle t = \frac{T}{N}$. Next, we approximate the control problem (3.3) by

$$\begin{cases} \mathbf{u}_k^{\triangle t} \in \mathcal{U}_k^{\triangle t}, \\ J_k^{\triangle t}(\mathbf{u}_k^{\triangle t}) \leq J_k^{\triangle t}(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}^{\triangle t}, \end{cases} \tag{3.12}$$

where

- $\mathcal{U}^{\triangle t} = (L^2(\omega))^N$.

- The cost functional $J_k^{\triangle t}$ is defined by

$$J_k^{\triangle t}(\mathbf{v}) = \frac{\triangle t}{2}\int_{\Sigma} |v^n|^2 d\Sigma + \frac{k}{2}\int_{\Sigma} (y^N - y_T)^2 d\Sigma \tag{3.13}$$

with y^n defined from v and y_0 from the solution of :

$$y^0 = y_0, \tag{3.14a}$$

$$\begin{cases} for \ \ n = 1, \dots, N \\ y^n \in H^1(\Sigma), \\ \int_\Sigma \frac{y^n - y^{n-1}}{\triangle t} z d\Sigma + \mu \int \nabla_\Sigma y^n . \nabla_\Sigma z d\Sigma = \int_\omega v^n z d\Sigma \ \ , \forall z \in H^1(\Sigma). \end{cases} \tag{3.14b}$$

The above problem is not associated with any boundary condition since Σ is a closed manifold, that is a manifold without a boundary. More specifically, the domain is a 2D manifold embedded in $\mathbb{R}^3$. Using classical convexity arguments, one can easily show that the discrete problem (3.12) has a unique solution, characterized by

$$DJ_k^{\triangle t}(\mathbf{u}_k^{\triangle t}) = 0, \tag{3.15}$$

where $DJ_k^{\triangle t}$ denotes the differential of $J_k^{\triangle t}$. Taking $\{\mathbf{v}, \mathbf{w}\} \rightarrow \triangle t \sum_{n=1}^{n=N} \int_\omega v^n w^n = (\mathbf{v}, \mathbf{w})_{\triangle t}$ as inner product over $\mathcal{U}^{\triangle t}$, and using a time discrete variant of the perturbation method used in Chapter 2, we can show that

$$(DJ_k^{\triangle t}(\mathbf{v}), \mathbf{w})_{\triangle t} = \triangle t \sum_{n=1}^{n=N} \int_\Sigma (v^n + p^n|_\omega) w^n d\Sigma, \tag{3.16}$$

where $\{p^n\}_{n=1}^N$ is obtained from $\mathbf{v}$ via the solution of (3.14) and of the following backward in time discrete initial-value problem (the associated adjoint system):

$$p^{N+1} = k(y^N - y_T), \tag{3.17}$$

for $n = N, \dots, 1$

$$\begin{cases} p^n \in H^1(\Sigma), \\ \int_\Sigma \frac{p^n - p^{n+1}}{\triangle t} z d\Sigma + \mu \int_\Sigma \nabla_\Sigma p^n \cdot \nabla_\Sigma z d\Sigma = 0, \forall z \in H^1(\Sigma). \end{cases} \tag{3.18}$$

Now one can easily derive the time-discrete analogue of the conjugate gradient algorithm, in order to solve the discrete control problem (3.12) via the optimality condition (3.15) using the methodologies discussed in Chapter 2. To avoid repetition, we will only describe the fully discrete analogue of the conjugate gradient algorithm after discussing the space discretizations of the control problem (3.3).

3.4.2 Full discretization of the control problem (3.3) for Σ a torus.

Concerning the space-discretization, we will use the parametrization based approach as explained in [40] which is depicted in Figure 3.2.

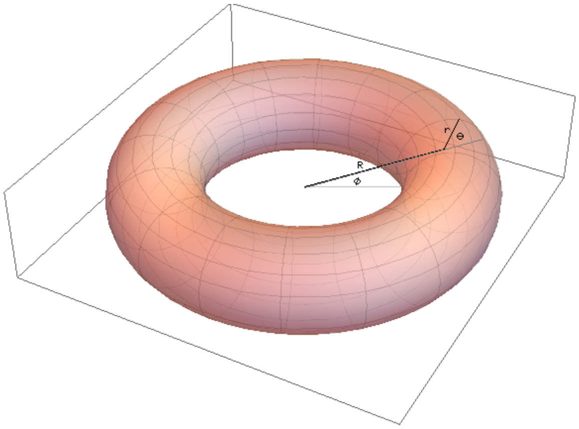


Figure 3.2: Torus surface view from top showing the major radius R and angle ϕ and the minor radius r and angle θ.

$$\begin{cases} x_1 = (R + rcos(\theta))cos(\phi) \\ x_2 = (R + rcos(\theta))sin(\phi) \\ x_3 = rsin(\theta), \end{cases} \tag{3.19}$$

to map Σ over the square $\hat{\Omega} = (0, 2\pi) \times (0, 2\pi)$ of the plane (ϕ, θ), periodic boundary conditions being used to take into account the fact that Σ is without boundary. However in Chapter 4, we will use the second option of approximating Σ by a polyhedral surface as mentioned in [7] to approximate the elliptic problems encountered in (3.14), (3.18). The above problem being formulated on a planar domain, one can easily space-approximate it, using a finite element method. Using the transformation (3.20), we have

$$d\Sigma = r(R + cos(\theta))d\phi d\theta, \tag{3.20}$$

and

$$\nabla_\Sigma y = [\frac{1}{(R + rcos(\theta))}\frac{\partial y}{\partial\theta}, \frac{1}{r}\frac{\partial y}{\partial\theta}] \tag{3.21}$$

Let denote by $\hat{\omega}$, the image of ω, by the geometrical transformation defined by (3.20); the time-discrete control problem can be then reformulated as:

$$\begin{cases} \mathbf{u}_k^{\triangle t} \in \mathcal{U}^{\triangle t}, \\ J_k^{\triangle t}(\mathbf{u}_k^{\triangle t}) \leq J_k^{\triangle t}(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}^{\triangle t} \end{cases} \tag{3.22}$$

with

- $\mathcal{U}^{\triangle t} = (L^2(\omega))^N$ equipped with the following inner product

$$(\mathbf{v}, \mathbf{w})_{\triangle t} = r \triangle t \sum \int_{\hat{\omega}} v^n w^n (R + rcos(\theta))d\phi d\theta \tag{3.23}$$

- The cost functional $J_k^{\triangle t}$ is defined by

$$\begin{cases} J_k^{\triangle t}(\mathbf{v}) = \frac{r}{2}\triangle t \int_{\hat{\Omega}} |v^n|^2 (R + rcos(\theta))d\phi d\theta + \\ \frac{k}{2} r \int_{\hat{\Omega}} (y^N - y_T)^2 (R + rcos(\theta))d\phi d\theta. \end{cases} \tag{3.24}$$

with $\{y^n\}_{n=1}^N$ obtained from y_0 and $\mathbf{v}$ via

$$y^0 = y_0, \tag{3.25}$$

for $n = 1, \ldots, N$ solve

$$\begin{cases} y^n \in H^1_p(\hat{\Omega}), \\ r\int_{\hat{\Omega}} \frac{y^n - y^{n-1}}{\triangle t} z(R + rcos(\theta))d\phi d\theta + \\ \mu\int_{\hat{\Omega}}(\frac{r}{R+rcos(\theta)}\frac{\partial y^n}{\partial\phi}\frac{\partial z}{\partial\phi} + \frac{R+rcos(\theta)}{r}\frac{\partial y^n}{\partial\theta}\frac{\partial z}{\partial\theta})d\phi d\theta = \\ r\int_{\hat{\omega}} v^n z(R + rcos(\theta))d\phi d\theta, \forall z \in H^1_p(\Omega) \end{cases} \tag{3.26}$$

where

$$\boxed{H^1_p(\hat{\Omega}) = \{z|z \in H^1(\hat{\Omega}), z(\phi, 0) = z(\phi, 2\pi) \;\; a.e \;\; on \;\; (0, 2\pi), z(0, \theta) = z(2\pi, \theta) \;\; a.e \;\; on \;\; (0, 2\pi)\}} \tag{3.27}$$

The differential $DJ^{\triangle t}_k(\mathbf{v})$ of $J^{\triangle t}_k$ at $\mathbf{v} \in \mathcal{U}^{\triangle t}$ is given by

$$(DJ^{\triangle t}_k(\mathbf{v}), \mathbf{w})_{\triangle t} = r \triangle t \sum_{n=1}^{N} \int_{\hat{\omega}} (v^n + p^n)w^n(R + rcos(\theta))d\phi d\theta, \;\; \forall \mathbf{w} \in \mathcal{U}^{\triangle t}, \tag{3.28}$$

with $\{p^n\}_{n=1}^{n=N}$ obtained from y^N via the solution of the following time-discrete adjoint system:

$$p^{N+1} = k(y^N - y_T), \tag{3.29}$$

for $n = N, \ldots 1$, solve

$$\begin{cases} p^n \in H^1_p(\hat{\Omega}), \\ r\int_{\hat{\Omega}} \frac{p^n - p^{n+1}}{\triangle t} z(R + rcos(\theta))d\phi d\theta + \\ \mu\int_{\hat{\Omega}}(\frac{r}{r+cos(\theta)}\frac{\partial p^n}{\partial\phi}\frac{\partial z}{\partial\phi} + \frac{R+rcos(\theta)}{r}\frac{\partial p^n}{\partial\theta}\frac{\partial z}{\partial\theta})d\phi d\theta = 0, \\ \forall z \in H^1_p(\hat{\Omega}). \end{cases} \tag{3.30}$$

The solution of problem (3.22) is characterized by $DJ_k^{\triangle t}(\mathbf{u}_k^{\triangle t}) = \mathbf{0}$.

Now, we can address the full discretization of the control problem (3.3). For that, we will first approximate $H_p^1(\hat{\Omega})$ using finite element approximations, since they are well-suited to the fact that state and co-state equations have been directly given in variational forms. We, therefore consider finite element triangulation $\mathcal{T}_h$ of $\hat{\Omega}$ with the following properties:

- $\mathcal{T}_h$ is a finite collection of closed triangles $\mathcal{K}$ contained in $\bar{\hat{\Omega}}$, with h denoting the length of the largest edges of $\mathcal{T}_h$.
- $\hat{\Omega} = \bigcup_{\mathcal{K} \in \mathcal{T}_h} \mathcal{K}$.
- If $\mathcal{K}$ and $\mathcal{K}'$ belong to $\mathcal{T}_h$ with $\mathcal{K} \neq \mathcal{K}$', we have either $\mathcal{K} \cap \mathcal{K}$' $= \emptyset$ or $\mathcal{K}$ and $\mathcal{K}$' have only one vertex in common .
- The function map (a 2π translation) maps a coordinate x in domain $\{\{\phi, \theta\} | \phi = 0, \theta \in [0, 2\pi]\}$ to domain $\{\{\phi, \theta\} | \phi = 2\pi, \theta \in [0, 2\pi]\}$, so that the right boundary is mapped to the left boundary and similarly for the other two edges of $\bar{\hat{\Omega}}$.

Figure 3.3 verifies the above assumptions

We approximate then $H_p^1(\hat{\Omega})$ by the following function space

$$\begin{cases} V_h = \{z | z \in C^0(\hat{\Omega}), z|_{\mathcal{K}} \in \mathbb{P}_1, \forall \mathcal{K} \in \mathcal{T}_h, \\ z(0,\theta) = z(2\pi,\theta), z(\phi,0) = z(\phi,2\pi), \forall \{\phi,\theta\} \in [0,2\pi]^2\} \end{cases} \tag{3.31}$$

where P_1 is the space of the two variable polynomials of degree ≤ 1. Also, we will assume that $\hat{\omega}$ is the union of triangles of $\mathcal{T}_h$ which allows us to formulate the fully approximate control problem as

$$\begin{cases} \mathbf{u}_{kh}^{\triangle t} \in \mathcal{U}_h^{\triangle t} \\ J_{kh}^{\triangle t}(\mathbf{u}_{kh}^{\triangle t}) \leq J_{kh}^{\triangle t}(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}_h^{\triangle t} \end{cases} \tag{3.32}$$

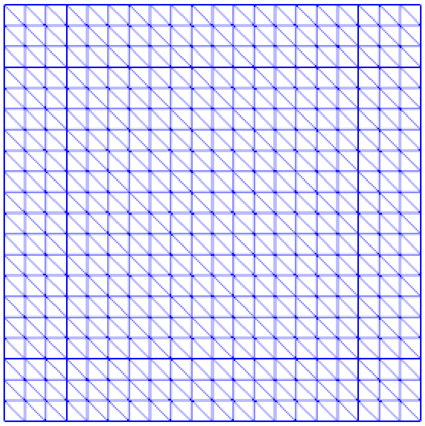


Figure 3.3: Triangulation of $\hat{\Omega}$

where in (3.33)

- The fully discrete control space $\mathcal{U}_h^{\triangle t}$ is defined by

$$\mathcal{U}_h^{\triangle t} = \{\mathbf{v}|\mathbf{v} = \{v^n\}_{n=1}^N, v^n \in V_h|_{\hat{\omega}}\} \tag{3.33}$$

- The space $\mathcal{U}_h^{\triangle t}$ is equipped with the following inner product.

$$\{v, w\} \to r \triangle t \sum_{n=1}^{N} \int_{\hat{\omega}} v^n w^n (R + rcos(\theta)) d\phi d\theta \tag{3.34}$$

that we will denote by $(.,.)_h^{\triangle t}$. We denote by $\mathcal{T}_{\hat{\omega}h}$ the subset of $\mathcal{T}_h$ defined by

$$\mathcal{T}_{\hat{\omega}h} = \{\mathcal{K}|\mathcal{K} \in \mathcal{T}_h, K \subset \bar{\hat{\omega}}\} \tag{3.35}$$

- The cost functional $J_{kh}^{\triangle t}$ is defined by

$$\begin{cases} J_{kh}^{\triangle t}(\mathbf{v}) = \frac{r}{2}\sum_{n=1}^{N}\int_{\hat{\omega}}|v^n|^2(R+rcos(\theta))d\phi d\theta + \\ \frac{k}{2}r\int_{\hat{\Omega}}|y^N-y_T|^2(R+rcos(\theta))d\phi d\theta, \end{cases} \tag{3.36}$$

with $\{y^n\}_{n=1}^{N}$ obtained from y_0 and $\mathbf{v}$ via the solution of

$$y^0 = y_{0h}(\in V_h). \tag{3.37}$$

For $n = 1, \dots, N$, solve

$$\begin{cases} y^n \in V_h, \\ r\int_{\hat{\Omega}}\frac{y^n-y^{n-1}}{\triangle t}z(R+rcos(\theta))d\phi d\theta + \\ \mu\int_{\hat{\Omega}}(\frac{r}{R+rcos(\theta)}\frac{\partial y^n}{\partial\phi}\frac{\partial z}{\partial\phi} + \frac{R+rcos(\theta)}{r}\frac{\partial y^n}{\partial\theta}\frac{\partial z}{\partial\theta})d\theta d\phi = \\ r\int_{\hat{\omega}}v^n z(R+rcos(\theta))d\phi d\theta, \forall z \in V_h \end{cases} \tag{3.38}$$

where y_{0h} is an approximation of y_0 belonging to V_h.

Proceeding as in Chapter 2, we can show that the differential $DJ_{kh}^{\triangle t}(\mathbf{v})$ of $J_{kh}^{\triangle t}$ at $\mathbf{v}$ is defined by

$$(DJ_h^{\triangle t}(\mathbf{v}),\mathbf{w})_h^{\triangle t} = r\triangle t\sum_{n=1}^{N}\int_{\hat{\omega}}(v^n+w^n)w^n(R+rcos(\theta))d\phi d\theta, \forall\mathbf{w}\in\mathcal{U}_h^{\triangle t}, \tag{3.39}$$

with $\{p^n\}_{n=1}^{N}$ obtained from the solution of the fully discrete adjoint problem

$$p^{N+1} = k(y^N - y_{T_h}); \tag{3.40}$$

for $n = N, \dots, 1$, solve

$$\begin{cases} p^n \in V_h, \\ r\int_{\hat{\Omega}}\frac{p^n-p^{n+1}}{\triangle t}z(R+rcos(\theta))d\phi d\theta + \\ \mu\int_{\hat{\Omega}}(\frac{r}{R+rcos(\theta)}\frac{\partial p^n}{\partial\phi}\frac{\partial z}{\partial\phi} + \frac{R+rcos(\theta)}{r}\frac{\partial p^n}{\partial\theta}\frac{\partial z}{\partial\theta})d\phi d\theta = 0, \forall z \in V_h. \end{cases} \tag{3.41}$$

In (3.40), y_{T_h} is an approximation of y_T belonging to V_h. It follows from the perturbation analysis done in the previous chapter that the differential $DJ_h^{\triangle t}$ at control $\mathbf{v}$ is given by

$$DJ_h^{\triangle t}(\mathbf{v}) = \{v^n + w^n|_{\hat{\omega}}\}_{n=1}^N. \tag{3.42}$$

The solution $\mathbf{u}_{kh}^{\triangle t}$ of the problem (3.32) is characterized by

$$DJ_h^{\triangle t}(\mathbf{u}_{kh}^{\triangle t}) = \mathbf{0}. \tag{3.43}$$

3.5 Implementation details

Taking advantage of equations (3.37)-(3.43), it is quite natural to solve the fully discrete control problem (3.32) via the solution of (3.43). This solution can be achieved through a *conjugate-gradient algorithm.* This conjugate-gradient algorithm reads as follows: Throughout the discussion we have :

$$\boxed{\int_{\hat{\Omega}} = \sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\mathcal{K}}}$$

and

$$\boxed{\int_{\hat{\omega}} = \sum_{\mathcal{K}\in\mathcal{T}_{\hat{\omega}h}} \int_{\mathcal{K}}}$$

Step 1:Initialization

$$\begin{cases} \mathbf{u}_0 = \{u_0^n\}_{n=1}^N \ is \ given \ in \ \mathcal{U}_h^{\triangle t} and \\ Y_{0h} \ is \ the \ interpolation \ of \ y_0^0 \ in \ the \ function \ space \ V_h; \end{cases} \tag{3.44}$$

$$\begin{cases} y_0^0 = Y_{0h} \\ solve \\ for \;\; n = 1, \dots, N, \\ r \int_{\hat{\Omega}} \frac{y_0^n - y_0^{n-1}}{\triangle t} z(R + rcos\theta) d\phi d\theta + \\ \mu \int_{\hat{\Omega}} (\frac{r}{R + rcos\theta} \frac{\partial y_0^n}{\partial \theta} \frac{\partial z}{\partial \theta} + \\ \frac{R + rcos\theta}{r} \frac{\partial y_0^n}{\partial \theta} \frac{\partial z}{\partial \theta}) d\phi d\theta = \\ r \int_{\hat{\omega}} u_0^n z (R + rcos\theta) d\phi d\theta, \forall z \in V_h, \end{cases} \tag{3.45}$$

and

$$\begin{cases} p_0^{N+1} = k(y_0^N - y_{Th}), \\ for \;\; n = N, \dots, 1, \;\; solve \\ r \int_{\hat{\Omega}} \frac{p_0^n - p_0^{n+1}}{\triangle t} z(R + rcos\theta) d\phi d\theta + \\ \mu \int_{\hat{\Omega}} (\frac{r}{R + rcos\theta} \frac{\partial p_0^n}{\partial \phi} \frac{\partial z}{\partial \phi} + \frac{R + rcos\theta}{r} \frac{\partial p_0^n}{\partial \theta} \frac{\partial z}{\partial \theta}) d\phi d\theta = 0, \;\; \forall z \in V_h. \end{cases} \tag{3.46}$$

Set

$$g_0^n = \{ u_0^n + p_0^n|_{\hat{\omega}} \}, \;\; \forall n = 1, \dots, N. \tag{3.47}$$

If

$$\frac{r \triangle t \sum_{n=1}^{N} \int_{\hat{\omega}} |g_0^n|^2 (R + rcos\theta) d\phi d\theta}{max[1, r \triangle t \sum_{n=1}^{N} \int_{\hat{\omega}} |u_0^n|^2 d\phi d\theta]} \leq tol$$

take $\mathbf{u}_{kh}^{\triangle t} = u_0$; otherwise set $\mathbf{w}_0 = \mathbf{g}_0$.

Step 2: Descent direction

Then for $q \geq 0$,assuming that $\mathbf{u}_q$,$\mathbf{g}_q$, and $\mathbf{w}_q$ are known, the last two different from $\mathbf{0}$, we compute $\mathbf{u}_{q+1}, \mathbf{g}_{q+1}$, and if necessary, $\mathbf{w}_{q+1}$ as follows:

Solve

$$\begin{cases} \bar{y_q}^0 = 0, \\ for\ \ n = 1, \ldots, N, \\ solve \\ r\int_{\hat{\Omega}} \frac{\bar{y_q}^n - \bar{y_q}^{n-1}}{\triangle t} z(R + rcos(\theta)) d\phi d\theta + \\ \mu\int_{\hat{\Omega}} (\frac{r}{R+rcos\theta} \frac{\partial \bar{y_q}^n}{\partial \phi} \frac{\partial z}{\partial \phi} + \frac{R+rcos\theta}{r} \frac{\partial \bar{y_q}^n}{\partial \theta} \frac{\partial z}{\partial \theta}) d\phi d\theta = \\ r\int_{\hat{\omega}} w_q^n z(R + rcos\theta) d\phi d\theta, \forall z \in V_h, \end{cases} \tag{3.48}$$

and

$$\begin{cases} \bar{p}_q^{N+1} = k\bar{y_q}^N; \\ for\ \ n = N, \ldots, 1,\ \ solve \\ \int_{\hat{\Omega}} \frac{\bar{p}_q^n - \bar{p}_q^{n+1}}{\triangle t} z(R + rcos\theta) d\phi d\theta + \\ \mu\int_{\hat{\Omega}} (\frac{r}{R+rcos\theta} \frac{\partial \bar{p_q}^n}{\partial \phi} \frac{\partial z}{\partial \phi} + \frac{R+rcos\theta}{r} \frac{\partial \bar{p_q}^n}{\partial \theta} \frac{\partial z}{\partial \theta}) d\phi d\theta = 0,\ \ \forall z \in V_h, \end{cases} \tag{3.49}$$

Set

$$\bar{\mathbf{g}}_q = \{\bar{g_q}\}_{n=1}^N = \{w_q^n + \bar{p_q}^n|_{\hat{\omega}}\}_{n=1}^N \tag{3.50}$$

and compute

$$\alpha_q = \frac{\sum_{n=1}^{n=N} \int_{\hat{\omega}} |g_q^n|^2 (R + rcos\theta) d\phi d\theta}{\sum_{n=1}^{n=N} \int_{\omega} \bar{g_q^n} w_q^n (R + rcos\theta) d\phi d\theta} \tag{3.51}$$

and then

$$\mathbf{u}_{q+1} = \mathbf{u}_q - \alpha_q \mathbf{w}_q \tag{3.52}$$

$$\mathbf{g}_{q+1} = \mathbf{g}_q - \alpha_q \bar{\mathbf{g}_q} \tag{3.53}$$

Step 3: **Testing the convergence: Construction of the new descent direction**.

If

$$\boxed{\frac{\sum_{n=1}^{N} \int_{\hat{\omega}} |g_{q+1}^n|^2 (R + rcos\theta) d\phi d\theta}{max[\sum_{n=1}^{N} \int_{\hat{\omega}} |g_0^n|^2 (R + rcos\theta) d\phi d\theta, \sum_{n=1}^{N} \int_{\hat{\omega}} |u_{q+1}^n|^2 (R + rcos\theta) d\phi d\theta]} \leq tol}$$

take $\mathbf{u}_{kh}^{\triangle t} = \mathbf{u}_{q+1}$; otherwise, compute

$$\beta_q = \frac{\sum_{n=1}^{n=N} \int_{\hat{\omega}} |g_{q+1}^n|^2 (R + rcos\theta) d\phi d\theta}{\sum_{n=1}^{n=N} \int_{\hat{\omega}} |g_q^n|^2 (R + rcos\theta) d\phi d\theta}, \tag{3.54}$$

and

$$\mathbf{w}_{q+1} = \mathbf{g}_{q+1} + \beta_q \mathbf{w}_q. \tag{3.55}$$

Do $q + 1 \rightarrow q$ and return to Step 2.

Concerning the choice of *tol*, following Chapter 1 of [13], we advocate to take $tol = 10^d$ where d is the the number of digits used for the floating point representation of the real numbers. By a slight variant of the analysis done in Chapter 1 of [13] for the "ordinary heat equation", one can prove that for given value of *tol*, the number of iterations necessary to achieve the convergence of the above algorithm, varies like $\sqrt{k}$.

Remark 3.5.1: The various discrete linear elliptic problems in the above conjugate gradient algorithm have been implemented in FEniCS, a python based package for the automation of solution of partial differential equations. Visualization tools like VTK and Para-view have been used to visualize the control and optimal state. FEniCS users only have to feed the variational formulation of the program and it generates corresponding forms related to the variational formulation which is further translated to a low-level language and it automatically executes the numerical quadrature (Simpson) scheme.

3.6 Numerical examples

3.6.1 Approximate controllability result: Test case-1

Table 3.1 provides information about the parameters taken in our first experiment. Our target function for the first test case is:

$$y_T(\phi, \theta) = cos(\phi)cos(\theta) \tag{3.56}$$

We choose our initial condition for the state equation to be $y_0 = 0$. First, we present numerical results for which the control is supported on different horizontal strip domains. The stopping tolerance for the conjugate-gradient algorithm is chosen to be $tol = 10^{-5}$. Our initial control is chosen as $u_0 = 0$. To investigate the effect of the penalty parameter, k, we chose k as $k = 10^2, 10^4$, and 10^6. The numerical results associated with the first experiment have been presented in Table 3.2. where y^c and u^c represent the computed state and corresponding optimal control, respectively. CGiters denotes the number of conjugate gradient iterations necessary to achieve convergence of the algorithm. We also compute the L^2 norm of the control represented by $\|u\|_{L^2(\hat{\omega}\times(0,T))}$ and controllability error given by

Table 3.1: Parameters used in our experiment

Physical Parameters	μ	1
	T	1
	R	2
	r	1
Penalty parameter	k	$10^2, 10^4, 10^6$
Time discretization parameter	$\triangle t$	10^{-2}

$\frac{\|y_T - y^c(T)\|_{L^2(\hat{\Omega})}}{\|y_T\|_{L^2(\hat{\Omega})}}$. We use Norm u^c for the L^2 norm of the control and Rel. Error for the controllability error. Here we took $\triangle\phi=\triangle\theta=\frac{\pi}{32}$ for the space discretization.

Table 3.2: Numerical results for different values of penalty parameter (k)

$\hat{\omega}$	k	CGIters	Norm u^c	Rel. Error
$(0, 2\pi) \times (0, 2\pi)$	10^2	10	18.6980	0.0080
	10^4	11	18.7226	0.0073
	10^6	11	18.7226	0.0064
	10^8	7	18.4987	0.0050
$(0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$	10^2	176	107.0434	0.0590
	10^4	236	112.4040	0.0465
	10^6	236	116.4316	0.0445
$(0, 2\pi) \times (0, \frac{\pi}{2})$	10^2	462	168.4714	0.1734
	10^4	952	263.6573	0.1320

We observed that we get better controllability results when we implement the control on the whole domain. Upon decreasing the area of the strip, the relative error increases which is consistent with the physical properties. We also want to state that the convergence is very poor for $\hat{\omega} = (0, 2\pi) \times (0, \frac{\pi}{2})$. We enforced smaller coefficient of viscosity to achieve convergence for values of k greater than 10^4. Next, we analyze the convergence of the algorithm in Table 3.3 with respect to the discretization of our problem domain. Here $h = \triangle\phi=\triangle\theta$ is the length of partition on ϕ and θ axis. If we refine the mesh, the convergence improves which is in accordance with the theory of finite element as proposed in [13]. Table 3.4 shows the influence of the viscosity coefficient, μ, on the convergence of our algorithm. For this, we choose $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$, k= 10^4, and $h = \triangle\phi=\triangle\theta = \frac{\pi}{64}$.

Table 3.3: Summary of convergence results for $k = 10^2$, $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$, $T = 1$, and $\triangle t = 0.01$

h	CGIters	Norm u^c	Rel. Error
$\frac{\pi}{8}$	660	109.6480	0.1259
$\frac{\pi}{16}$	327	124.9445	0.0590
$\frac{\pi}{32}$	176	107.0434	0.0342
$\frac{\pi}{64}$	66	82.3933	0.0242

We observe that there exists an optimal value of the coefficient of viscosity(μ) affecting the convergence of the algorithm. In Section 3.6.1.1, we present the visualization of our numerical experiment. The visualization shows that when $\hat{\Omega} = \hat{\omega}$, y_T and $y^c(T)$ are quite close to each other, but from a practical point of view it is not realistic.

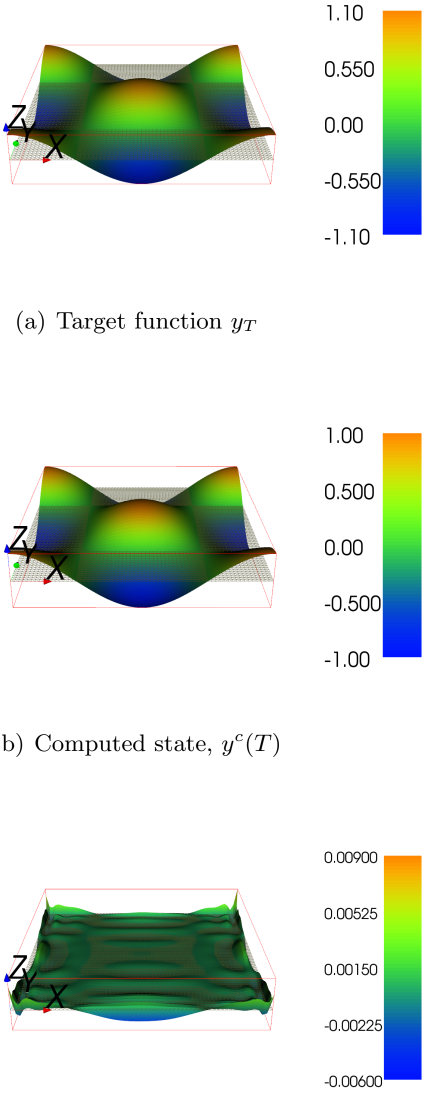
Table 3.4: Effect of the viscosity coefficient(μ), $\triangle\theta = \triangle\phi = \frac{\pi}{32}$.

μ	CGIters	Norm u^c	Rel. Error
1	14	18.6980	0.0080
100	483	705.6426	0.6351
0.1	14	16.1778	0.0075
0.001	10	2.2196	0.0018
0.0001	251	222.1901	0.6073

3.6.1.1 Visualization of the numerical results

We plot the desired target y_T, optimal control u^c and computed state $y^c(T)$ when the control domain is $\hat{\omega} = (0, 2\pi) \times (0, 2\pi)$ Figure 3.4(a), (b), and (c), respectively. We choose $k = 10^4$, $\triangle t = \frac{1}{100}$, $\triangle\phi = \triangle\theta = \frac{\pi}{64}$, and $y_T(\phi, \theta) = cos(\phi)cos(\theta)$ as our target function. The initial condition chosen for the state equation is y_0= 0 and coefficient of viscosity $\mu = 1$. We also present the visualization of numerical results for control domain $(0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$ in Figure 3.5. We observed that our algorithm achieved a good agreement between the desired and computed state for the physical parameters mentioned in Table 3.1. We also plot the difference between the desired and computed state which is taken after projecting onto function space, V_h. We observed that the difference is maximum for the part of the domain where the control is inactive. We observed in the visualization that implementing control on the whole domain gives a better performance than implementing it on the horizontal or vertical strip of the domain. For $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$, we choose same target function but we choose coefficient of viscosity μ to be 0.001. We choose a small timestep for the value of $k = 10^4$ since it is computationally expensive to get convergence for coefficient of viscosity $\mu = 1$. Finally, we present the snapshots of control at different instants of time in Figure 3.6.

Remark 3.6.1.1. The color bar represents the value of the function corresponding to points on the rectangular domain, while in case of the difference, it represents the value of the difference taken pointwise. Here we use X, Y, and Z just to represent the coordinate axes as automatically embedded in the VIPER visualization package[31]. X and Y symbolises the ϕ and θ axes respectively.

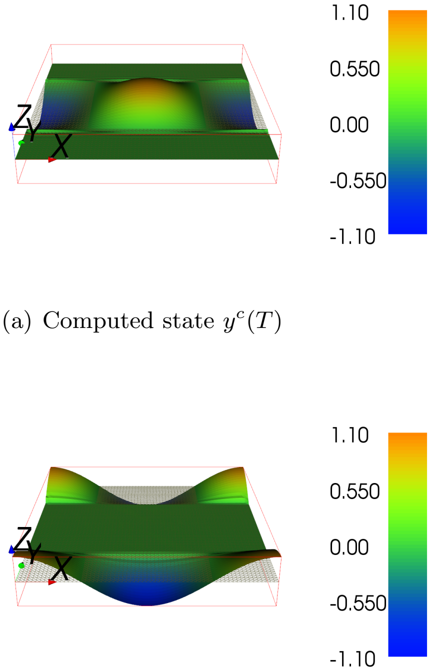


(a) Target function y_T

(b) Computed state, $y^c(T)$

(c) Difference between the target function, y_T, and computed state, $y^c(T)$

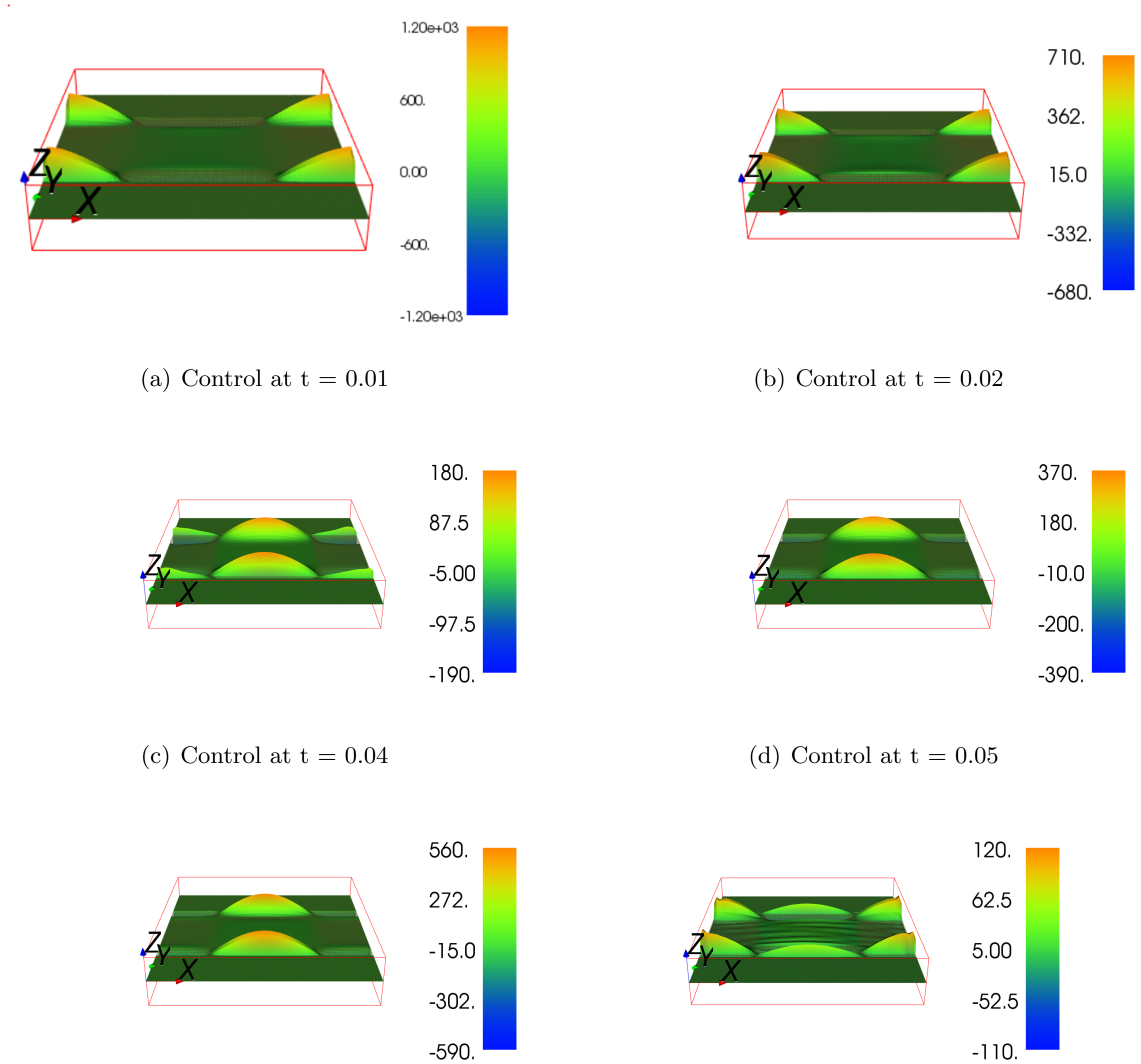
Figure 3.4: Visualization of the numerical results for $\hat{\omega} = (0, 2\pi) \times (0, 2\pi), \mu = 1, k = 10^4$, and $T = 1$. The color bar represents the value of the function at mesh coordinates while in the case of the difference it represents the value of the difference taken pointwise.



(a) Computed state $y^c(T)$

(b) Difference between the target function and computed state

Figure 3.5: Visualization of the numerical results for $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2}), \mu = 0.001$, $k = 10^4$, and $T = 1$. The color bar represents the value of the function at mesh coordinates while in the case of the difference it represents the value of the difference taken pointwise.



(a) Control at t = 0.01

(b) Control at t = 0.02

(c) Control at t = 0.04

(d) Control at t = 0.05

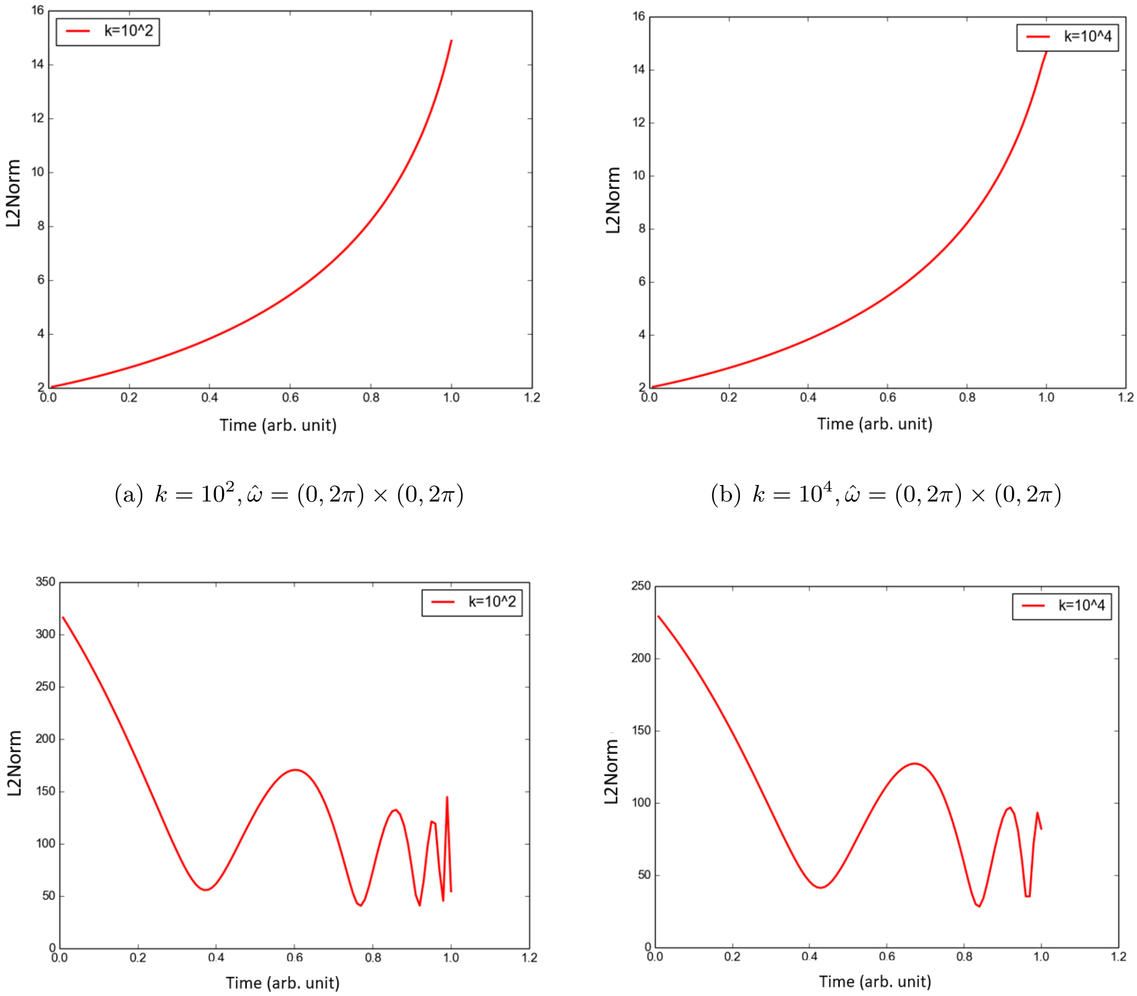
(e) Control at t = 0.07

(f) Control at t = 0.09

Figure 3.6: Snapshots of the optimal control at different time instants for $\omega = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2}), \mu = 1$, $k = 10^4$, and $T = 0.1$. The color bar represents the value of the function at mesh coordinates.

3.6.1.2 Visualization of the time evolution of the L^2 norm of optimal control

We present the visualization of the L^2 norm of control for the control domains $\hat{\omega} = (0, 2\pi) \times (0, 2\pi)$ and $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$.



(a) $k = 10^2, \hat{\omega} = (0, 2\pi) \times (0, 2\pi)$

(b) $k = 10^4, \hat{\omega} = (0, 2\pi) \times (0, 2\pi)$

(c) $k = 10^2, \hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$

(d) $k = 10^4, \hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$

Figure 3.7: Time evolution of the L^2 norm of the optimal control for $k = 10^2, 10^4$, $T = 1$, $\triangle t = 0.01$ and $\mu = 1$.

3.6.2 Approximate controllability result: Test case-2

In this section, we present the results of our second experiment with target function y_T mentioned in (3.57). Table 3.5 provides the parameters used in this second experiment. We choose small time steps since larger time steps are computationally expensive.

$$y_T(\phi, \theta) = Acos(\phi) + Bsin(\theta) \tag{3.57}$$

Table 3.5: Parameters used in the second experiment

Physical Parameters	μ	1
	T	0.1
	R	2
	r	1
Penalty parameter	k	$10^2, 10^4, 10^6$
Time discretization parameter	$\triangle t$	10^{-2}

We choose our initial condition for the state equation as $y_0 = 0$. First, we present numerical results when the control is supported on the horizontal strip domain $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$. In (3.58), we chose $A = 1$ and $B = 1$ for our numerical experiments. The stopping tolerance for the conjugate gradient algorithm was chosen to be $tol = 10^{-5}$. We chose $\mathbf{u}_0 = 0$ as our initial guess for the control. To investigate the effect of the penalty parameter k, we chose different values of k namely $k = 10^4$, $k = 10^6$ and $k = 10^8$. The numerical results associated with this second experiment is presented in Table 3.6 where y^c and u^c represent the computed state and corresponding optimal control, respectively. CGiters denotes the number of conjugate gradient iterations necessary to achieve the convergence of the algorithm. We compute the L^2 norm of the control represented by $\|u\|_{L^2(\hat{\omega}\times(0,T))}$ and controllability error given by $\frac{\|y_T - y^c(T)\|_{L^2(\hat{\Omega})}}{\|y_T\|_{L^2(\hat{\Omega})}}$. Here we took $\triangle\phi$=$\triangle\theta$=$\frac{\pi}{32}$ for the space discretization.

Better control behaviour was seen when we implement the control on the whole domain. As we decrease the area of the strip, the relative error increases which is consistent with the physical properties. Next, we analyse the convergence of the algorithm in Table 3.7 with respect to the discretization of our problem domain. Here $h = \triangle\phi$=$\triangle\theta$ is the length of partition on ϕ and θ axis. We observed that as we go upon refining the mesh, the Rel. Error shows stabilizing behaviour suggesting that the conjugate gradient algorithm gives a mesh independent result for this test problem. In particular, the computed state differs from the desired target by 1.5 percent in most cases for this choice of target function.

Table 3.8 shows the influence of the viscosity coefficient, μ, on the convergence of our algorithm. For this we chose $\hat{\omega} = (0, 2\pi) \times (0, \frac{\pi}{2})$ and $h = \triangle\phi$=$\triangle\theta = \frac{\pi}{64}$. In Section 3.6.2.1, we present the visualization of our numerical experiment. The visualization shows that for $\hat{\Sigma} = \hat{\omega}$, y_T and $y^c(T)$ are quite close to each other, but from practical point of view it is not realistic.

Table 3.6: Numerical results for different values of penalty parameter (k)

$\hat{\omega}$	k	CGIters	Norm u^c	Rel. Error
$(0, 2\pi) \times (0, 2\pi)$	10^2	6	18.6980	0.0149
	10^4	7	18.4987	0.0113
	10^6	7	18.7226	0.0083
	10^8	7	18.7226	0.0083
$(0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$	10^2	19	91.7723	0.2959
	10^4	73	145.3960	0.1278
	10^6	105	245.3960	0.0728
$(0, 2\pi) \times (0, \frac{\pi}{2})$	10^2	45	127.2236	0.3708
	10^4	60	163.6573	0.1840
	10^6	75	445.0803	0.1731

Table 3.7: Summary of convergence results for $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$, $\triangle t = 0.01$, and $k = 10^8$.

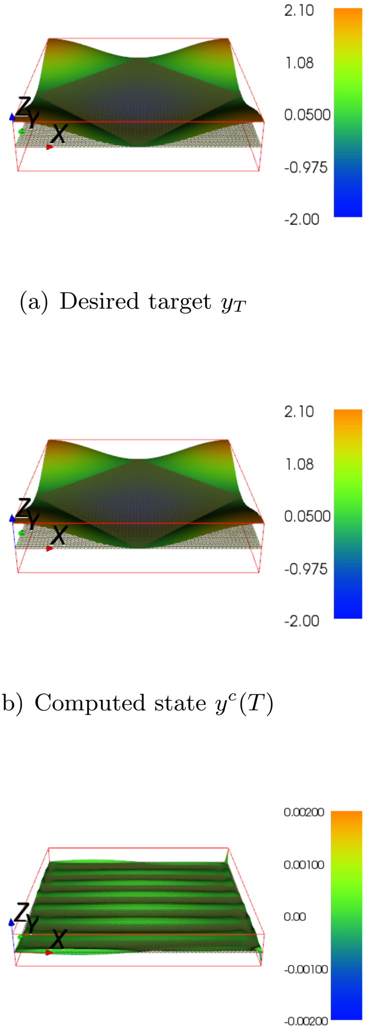
h	CGIters	Norm u^c	Rel. Error
$\frac{\pi}{28}$	29	35.3329	0.0168
$\frac{\pi}{32}$	28	37.0525	0.0154
$\frac{\pi}{64}$	21	37.1098	0.0152
$\frac{\pi}{96}$	20	37.8519	0.0152

Table 3.8: Effect of the viscosity coefficient (μ)

μ	CGIters	Norm u^c	Rel. Error
1	6	18.6980	0.0149
10	15	269.2369	0.0176
0.1	14	16.1778	0.0075
0.01	4	7.2201	0.0075

3.6.2.1 Visualization of the numerical results

We present the plot of the desired target, optimal control and state solution of the final iteration for the second experiment. We chose $k = 10^4$, $\triangle t = \frac{1}{100}$, $\triangle\phi = \triangle\theta = \frac{\pi}{64}$ and $y_T(\phi, \theta) = cos(\phi) + sin(\theta)$ as our target function. We chose initial condition, y_0, to be zero for the state equation and $u_0 = 0$ as the initial guess for control. Here we chose our viscosity coefficient to be $\mu = 1$. We present the visualization for the control domains $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$ and $(0, 2\pi) \times (0, \frac{\pi}{2})$. We observe that our algorithm achieves a good agreement between the desired and computed state for the physical parameters mentioned in Table 3.5. We also plotted the difference between the desired and computed state which is taken after projecting on the function space V_h. We observed that the difference is maximum for the part of the domain where the control is inactive. We observed that implementing the control on the whole domain gives a better performance than implementing it on a horizontal or vertical strip of the domain.



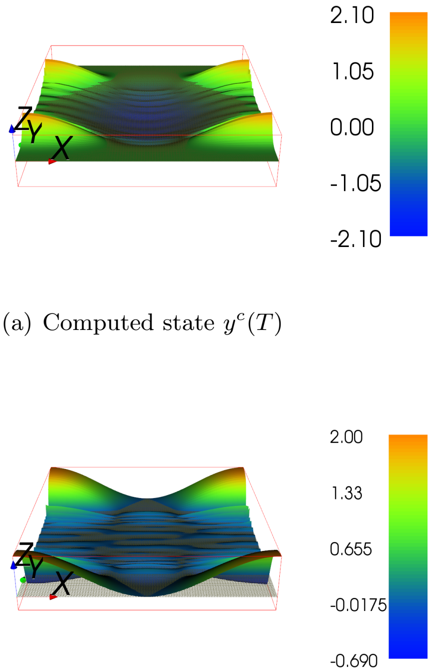
(a) Desired target y_T

(b) Computed state $y^c(T)$

(c) Difference between the desired target y_T and computed state $y^c(T)$

Figure 3.8: Visualization of the numerical results for $\hat{\omega} = (0, 2\pi) \times (0, 2\pi)$, $\mu = 1$, $k = 10^4$, and $T = 0.1$. The color bar represents the value of the function at mesh coordinates while in the case of the difference it represents the value of the difference taken pointwise.

We also present the visualization results for the vertical strip and horizontal strip domain of the second test case in Figure 3.9 and Figure 3.10. Our target function for the vertical and horizontal strip domain is same as the one we chose for the whole domain. We observe a similar pattern for both horizontal and vertical strip domains i.e., the desired and optimized solution profiles are of similar shape but their maximum values differ significantly on the part of the domain where the control is not active. Finally, we visualize the time evolution of L^2 norm of the optimal control for control domain $\hat{\omega} = (0, 2\pi) \times (0, 2\pi), (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$, and $(\frac{\pi}{2}, \frac{3\pi}{2}) \times (0, 2\pi)$ in Figure 3.11. We chose $k = 10^2$ and $T = 1$ for these experiments. The number of iterations necessary to achieve the convergence in case of vertical strip domains was greater than the number of iterations needed in case of horizontal strip domain. We also observe that L^2 norm of control showed the same behaviour as the one observed for the first experiment.

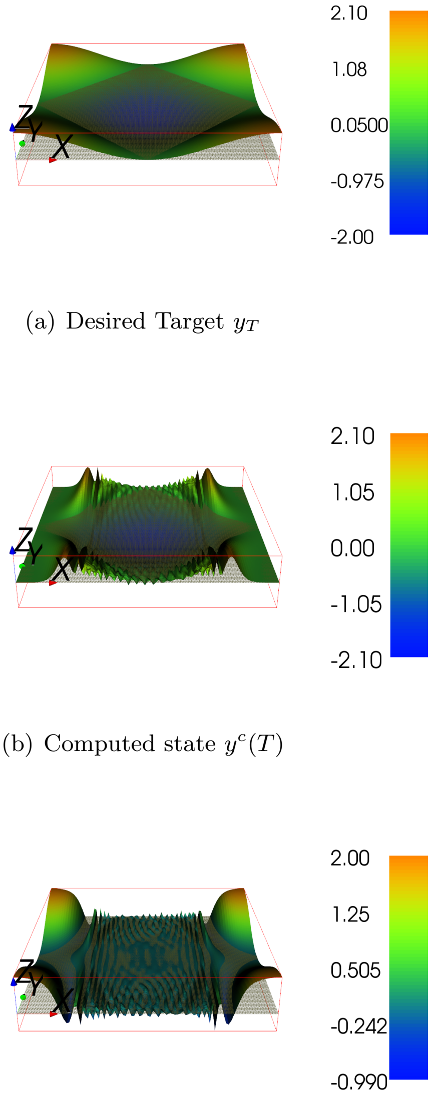


(a) Computed state $y^c(T)$

(b) Difference between the target function, y_T, and computed state y^c

Figure 3.9: Visualization for the horizontal strip $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2}), \mu = 1, k = 10^4$, and $T = 0.1$. The color bar represents the value of the function at mesh coordinates while in the case of the difference it represents the value of the difference taken pointwise.

Remark 3.6.2.2 The difference is nearly zero where the control is active and maximum where the control is inactive.

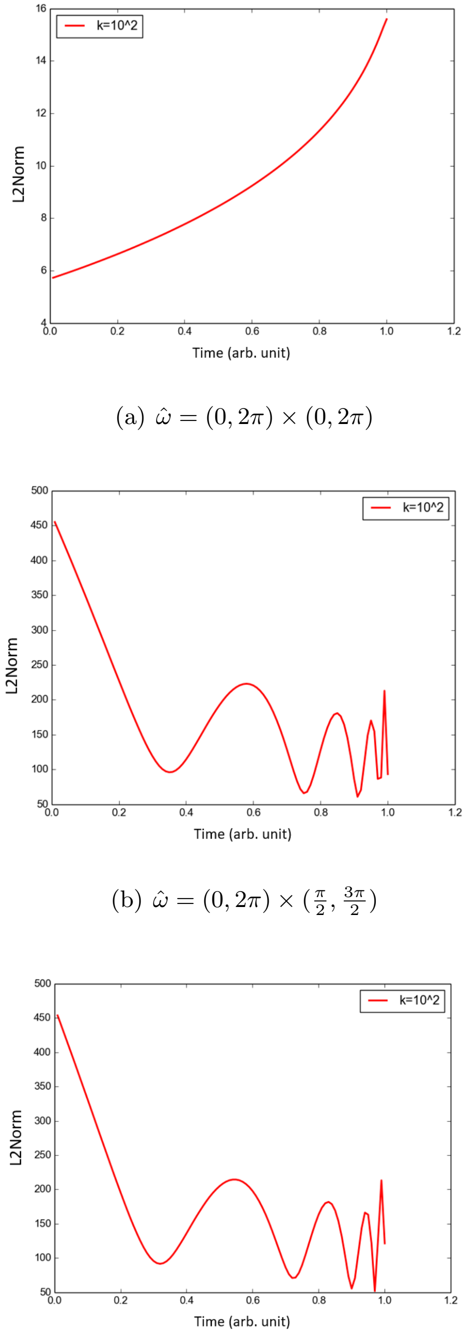


(a) Desired Target y_T

(b) Computed state $y^c(T)$

(c) Difference between the target function, y_T, and computed state $y^c(T)$

Figure 3.10: Visualization for vertical strip domain $\hat{\omega} = (\frac{\pi}{2}, \frac{3\pi}{2}) \times (0, 2\pi), \mu = 1, k = 10^4$, and $T = 0.1$. The color bar represents the value of the function at mesh coordinates while in the case of the difference it represents the value of the difference taken pointwise.



(a) $\hat{\omega} = (0, 2\pi) \times (0, 2\pi)$

(b) $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$

(c) $\hat{\omega} = (\frac{\pi}{2}, \frac{3\pi}{2}) \times (0, 2\pi)$

Figure 3.11: Time evolution of the L^2 norm of the optimal control for $k = 10^2, T = 1$, and $\triangle t = 0.01$.

3.6.3 Null-controllability

Our last set of numerical experiment associated with control on the torus is related to the test case where the target function is $y_T = 0$. As stated in [26], the distributed control problem under consideration will go from y_0 (initial state) to $y_T = 0$ in finite time. Under these circumstances, a question which arises naturally is the following one: since the "natural tendency" of the system, if perturbed from $y = 0$, is to return to the above state at "exponential speed", is it possible to find a control $\mathbf{u}$ such that the target function $y_T = 0$ can be reached in finite time? If this is possible, the system has the so-called null-controllability property. Indeed, it is proven in [26] that $\forall$ T, $0 < T < \infty$, and $y_0 \in L^2(\Sigma)$, there exists $u \in L^2(\Sigma \times (0, T))$ such that the corresponding solution of the distributed system under consideration verifies $y(T) = 0$.

We want to check the null-controllability property with respect to time T. We, therefore begin our numerical experiment with $\hat{\omega} = (\frac{\pi}{2}, \frac{3\pi}{2}) \times (0, 2\pi)$. We take as in [40], r=0.5, R=2, the other parameters being the same as before, namely $\mu = 1$, $\triangle t$=$\frac{1}{100}$, $\triangle\theta$=$\triangle\phi = \frac{\pi}{32}$, and $tol = 10^{-5}$, the conjugate gradient algorithm being initialized by $u_0 = 0$ and a initial condition chosen for the state equation was

$$y_0(\phi, \theta) = cos(\phi) + sin(\theta) \tag{3.58}$$

Our algorithm has better convergence properties for null controllability at the penalty parameter $k = 10^8$ which is desirable since our approach was to compute an approximation of the control of minimal norm (L^2 metric) realizing the null-controllability, using values of k as large as possible. In Table 3.10, we present the numerical results related to null controllability for times T=1, 2, and 3.

We present the visualization of our numerical experiments for different values of time in Figure 3.13, 3.14 and 3.15. We observe that approximate null-controllability is better

Table 3.9: Approximate null-controllability, $\hat{\omega}$=$(\frac{\pi}{2}, \frac{3\pi}{2}) \times (0, 2\pi)$, and $T = 0.1$

$\hat{\omega}$	k	CGIters	$\|y^c(T)\|_{L^2(\Sigma)}$
$(\frac{\pi}{2}, \frac{3\pi}{2}) \times (0, 2\pi)$	10^2	37	0.1401
	10^4	49	0.1225
	10^6	46	0.1243
	10^8	7	0.0389

Table 3.10: Approximate null-controllability, $\hat{\omega}$=$(0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$, and $\triangle\theta = \triangle\phi = \frac{\pi}{32}$

T	k	CGIters	$\|y^c(T)\|_{L^2(\Sigma)}$
1	10^6	59	0.2848
	10^8	13	0.1586
2	10^6	8	0.1370
	10^8	8	0.1366
3	10^6	6	0.0842
	10^8	6	0.0841

observed when we increase the simulation time, which is not surprising as per the theory stated in [26]. We observe that as T increases and when we reach T=9, null controllability is almost achieved.

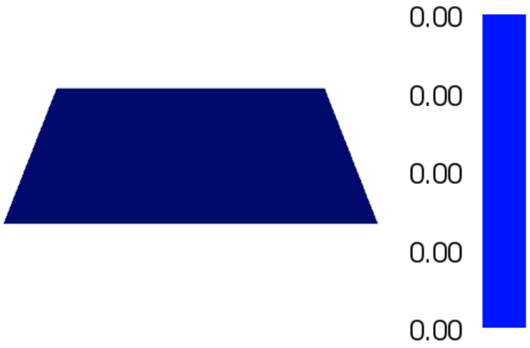
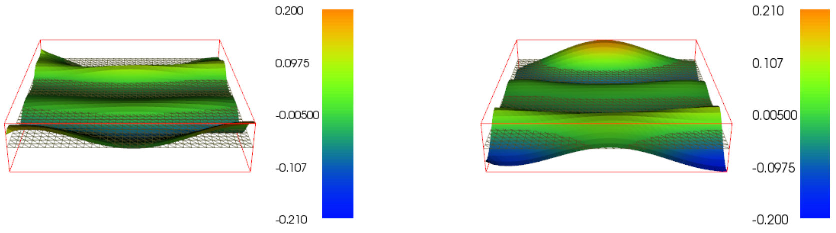


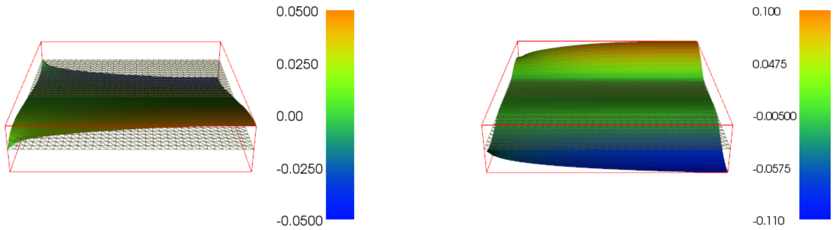
Figure 3.12: Desired target $y_T = 0$



(a) Computed state $y^c(T)$

(b) Difference between the desired target, y_T, and computed state $y^c(T)$

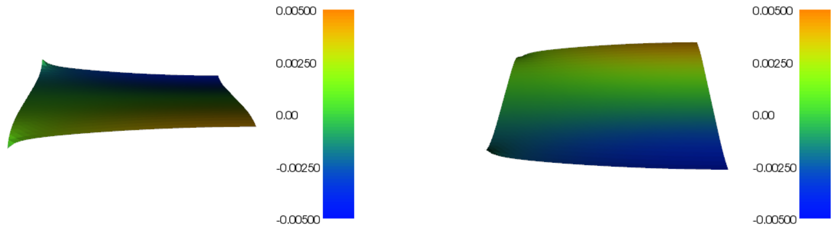
Figure 3.13: Approximate null controllability for $T = 1$, $\triangle\theta = \triangle\phi = \frac{\pi}{32}$, and $k = 10^8$. The color bar represents the value of the function at mesh coordinates while in the case of the difference it represents the value of the difference taken pointwise.



(a) Computed State

(b) Difference between the target function and state solution

Figure 3.14: Approximate null controllability for T=3, $\triangle\theta = \triangle\phi = \frac{\pi}{32}$, and $k = 10^8$. The color bar represents the value of the function at mesh coordinates while in the case of the difference it represents the value of the difference taken pointwise.

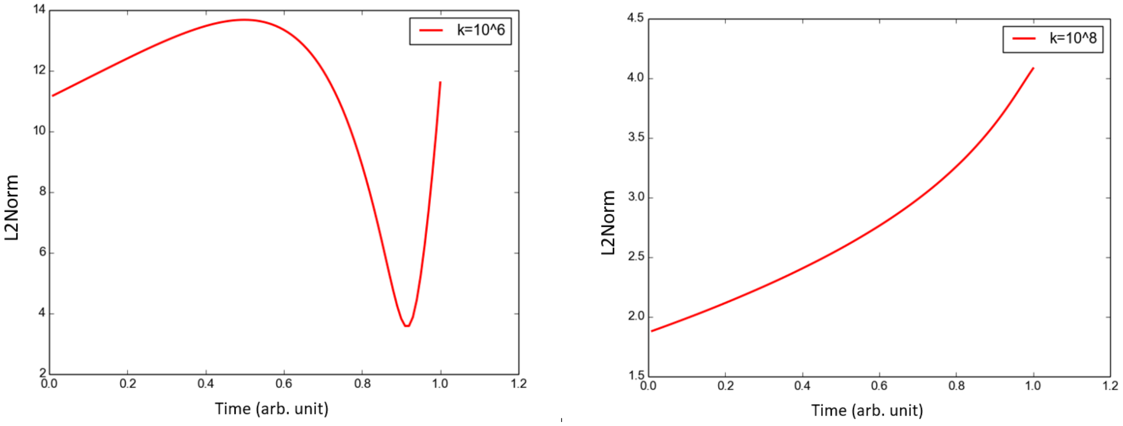


(a) Computed State $y^c(T)$

(b) Difference between the target function and state solution

Figure 3.15: Approximate null controllability for T=9,$\triangle\theta = \frac{\pi}{32} = \triangle\phi$, and $k = 10^8$. The color bar represents the value of the function at mesh coordinates while in the case of the difference it represents the value of the difference taken pointwise.

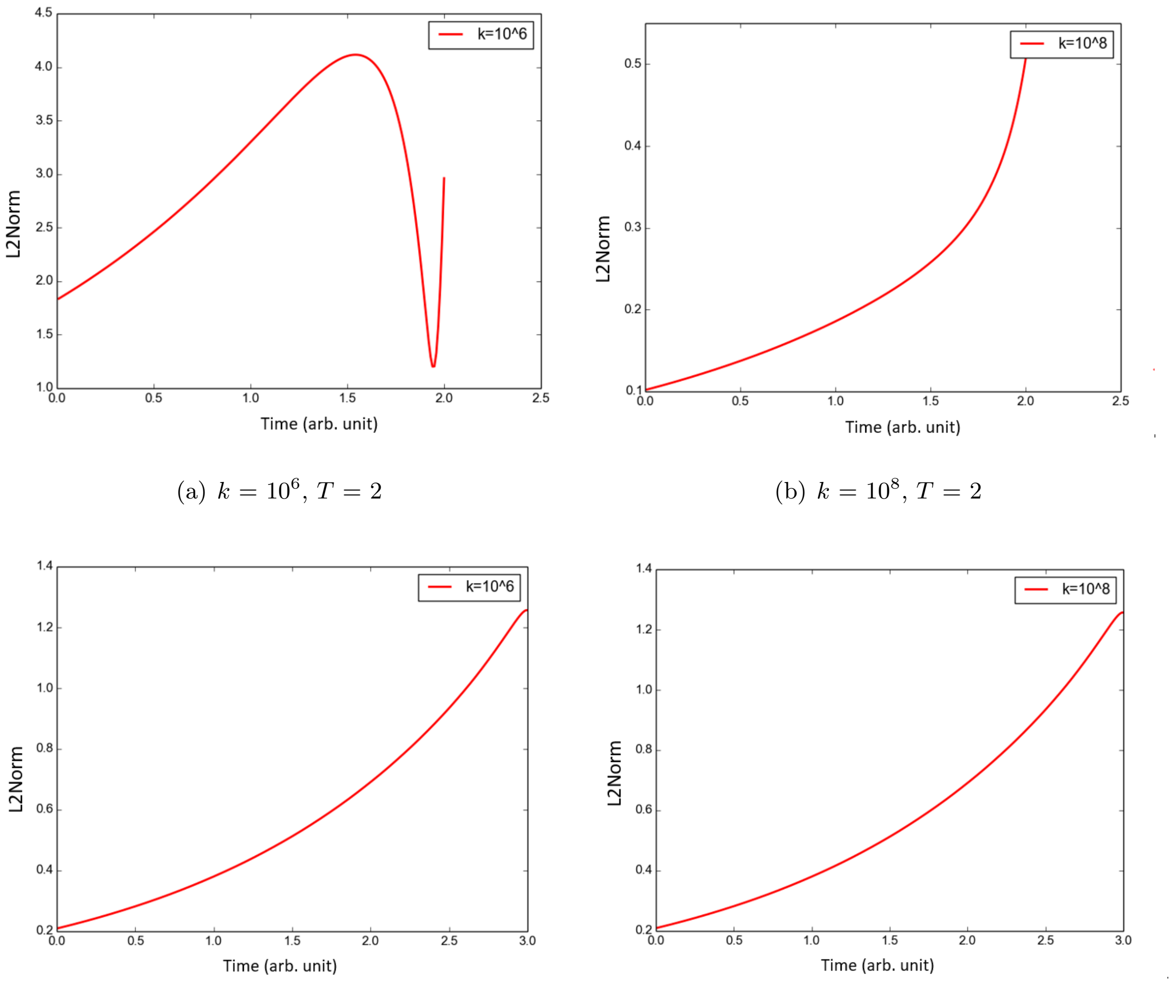
The numerical results we obtained were consistent with those stated in [26] regarding null-controllability. We will conclude this section with the visualization of the evolution of the L^2 norm of control for $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$ and different values of T. We chose as the penalty parameter $k = 10^6$ and $k = 10^8$. We observed that for $T \geq 3$, the L^2 norm of control for $k = 10^6$ and $k = 10^8$ have similar behaviour.



(a) $k = 10^6$, $T = 1$

(b) $k = 10^8$, $T = 1$

Figure 3.16: Time evolution of the L^2 norm of the optimal control for $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$, $k = 10^6, 10^8$, and T=1.



(a) $k = 10^6$, $T = 2$

(b) $k = 10^8$, $T = 2$

(c) $k = 10^6$, $T = 3$

(d) $k = 10^8$, $T = 3$

Figure 3.17: Time evolution of the L^2 norm of the optimal control for $\hat{\omega} = (0, 2\pi) \times (\frac{\pi}{2}, \frac{3\pi}{2})$, $k = 10^6, 10^8$, $T = 2$, and $T = 3$.

Chapter 4

A second approach to the optimal control of diffusion phenomena on manifolds

In this chapter, we propose an alternative approach for the control of diffusion phenomena on manifolds. This idea popped up after having discussion with Dr. Robert Azencot of University of Houston. He suggested me to work with the mesh itself and then implement control on the given mesh geometry, so my effort was directed to search for or build a robust solver for PDE's on a three dimensional mesh. The idea of using FEniCS for automating the solution of PDE over manifolds already had been implemented by that time. There are other finite element libraries as stated in Glowinski et al. [7] where the solution is approximated using a isoparametric finite element method. The only difference is that the facets of a polyhedral surface are quadrilaterals rather than triangles. This chapter covers an alternative approach to control diffusion phenomena on manifolds. This chapter also summarizes the distinctive mathematical features of finite element formulations defined over computational domains that are immersed manifolds.

4.1 Mathematical formulation in FeniCS

Through out this section, we let Σ be a smooth m-dimensional manifold immersed in $\mathbb{R}^n$, with $m \leq n$. For simplicity, we also let $1 \leq m \leq n$. We will refer to m as the manifold dimension or topological dimension, and n as the physical or geometric dimension. We approximate this manifold by piecewise linear tesellation of simplices (intervals in one topological dimension and triangles or tetrahedra in the three topological dimensions) $\mathcal{T} = \{K\}$. In particular, each simplex cell K in the mesh $\mathcal{T}$ will then have a topological dimension, m, and geometric dimension, n.

4.1.1 Galerkin projection on the manifold

The finite element method is based on the concept of finite element spaces. A finite element space V is defined to contain all functions that have some specified polynomial expansion in each cell of the mesh, together with some specified continuity constraint between neighbouring cells. Broadly speaking, the finite element discretization of a partial differential equation can be described as the projection of that equation onto some finite element space V. The Galerkin projection of a function f onto a finite element space, V, is a basic finite element operation and defined as the function, v, in V such that

$$\int_{\mathcal{T}} vwdx = \int_{\mathcal{T}} fwdx, \forall \ w \in V. \tag{4.1}$$

If V is N-dimensional with basis $\{\phi_j\}$ then we may write

$$v = v_j \phi_j, \tag{4.2}$$

where $\{v_j\}_{j=1}^N$ are the expansion coefficients of v relative to the basis $\{\phi_j\}$. Here, we, follow Einstein summation convention in which summation occurs over a repeated index within a product. Taking $w = \phi$ in (4.1) for $i = 1, \dots, N$, we obtain a finite dimensional linear system for the expansion coefficients v_j :

$$M_{ij} v_j = b_i, \tag{4.3}$$

with

$$M_{ij} = \int_{\mathcal{T}} \phi_i \phi_j dx = \sum_{K \in \mathcal{T}} \int_K \phi_i \phi_j dx \tag{4.4}$$

and

$$b_i = \int_{\mathcal{T}} f \phi_i dx = \sum_{K \in \mathcal{T}} \int_K f \phi_i dx \tag{4.5}$$

Moreover, for each $K \in \mathcal{T}$, we label the local integral contributions

$$M_{K,ij} = \int_K \phi_i \phi_j dx, \tag{4.6}$$

and

$$b_{K,i} = \int_K f \phi_i dx. \tag{4.7}$$

In view of equations (4.6) and (4.7), the assembly of the operators M and b reduces to the evaluation of sums of certain integrals over the individual cells $K \in \mathcal{T}$. This procedure is the standard assembly strategy for the finite element method.

4.1.2 Change of coordinates

A change of coordinates to reference cell K_0 offers a standard and efficient evaluation procedure for each of the local contributions in equations (4.6) and (4.7). Recalling that each cell $K \subset \mathbb{R}^n$ is of topological dimension m, we define a fixed reference cell $K_0 \subset \mathbb{R}^m$ and assume that there exists a mapping G_K such that $T = G_K(K_0)$. We write here and throughout $X = (X_1, \dots, X_m)$ for the coordinates of a points in reference space and $x = (x_1, \dots, x_n)$ for the coordinates in the physical space. Figure 4.1 illustrates this mapping and the notation employed.

Similarly, we will employ lower case Greek letters for basis functions in physical space, and the corresponding upper case letters for the pullback of those functions to the reference cell. For scalar valued functions, the pullback is through function composition:

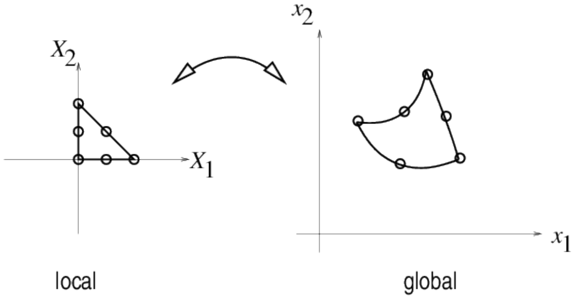


Figure 4.1: Isoparametric mapping of a $P2$ element

$$\Phi_i(X) = \phi_i(x) = \phi_i(G_K(X)) \tag{4.8}$$

Using the definitions above and the usual change of coordinates rule, eq.(4.4) becomes

$$\begin{cases} \int_K \phi_i(x)\phi_j(x)dx = \int \phi_i(G_K(X))\phi_j(G_K(X))dx \\ \int_{K_0} \Phi_i(X)\Phi_j(X)|\mathbf{J}_K|dX, \end{cases} \tag{4.9}$$

where J_K is the Jacobian of the transformation G_K and $|\mathbf{J}_K|$ is the Jacobian determinant.

4.1.3 The Jacobian and its pseudo-determinants

The derivation in equation (4.9) applies for both the standard case $m{=}n$ and the immersed manifold case where $m < n$. The only difference for the latter case is the generalised definitions of the jacobian and its determinant. In general, the Jacobian, $\mathbf{J}$, of the transformation $G : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is given by the matrix

$$J_{\gamma\tau} = \frac{\partial G(X)_\gamma}{\partial X_\tau} = \frac{\partial x_\gamma}{\partial X_\tau}; \;\; \gamma = 1, \ldots, n, \;\; \tau = 1, \ldots, m \tag{4.10}$$

Note that τ varies over the manifold dimension, m, which is also the geometric and topological dimension of the reference cell, while γ varies over the physical dimension, n. To make this concrete, the Jacobian for a two dimensional manifold immersed in $\mathbb{R}^3$ is given by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} \end{bmatrix}$$

For affine transformation, G_K, the jacobian, $\mathbf{J}_K$, will be constant over each cell K. For non-affine transformations, for instance in the case of curved cells, the Jacobian will vary as function of X.

The Jacobian pseudo-determinant is the transformation of the volume of the differential integral measure. For a one-dimensional manifold, this is the length of the single column vector of $\mathbf{J}$, while for a two-dimensional manifold this is the volume of the parallelogram spanned by two columns of $\mathbf{J}$. More precisely, writing the Jacobian in terms of its column vectors $\mathbf{J} = [\mathbf{J}_1|\mathbf{J}_2|\ldots|\mathbf{J}_m]$, we have

$$|\mathbf{J}| = \begin{cases} |\mathbf{J}_1|_2 \ \ , m = 1, \\ \\ |\mathbf{J}_1 \times \mathbf{J}_2|_2 \ \ , \ \ m = 2, \end{cases} \tag{4.11}$$

where $|.|_2$ denotes the Euclidean norm. The pseudo-determinant employed here is the square root of the Gram determinant. Note that, in the n=m case, this reduces to the absolute value of the usual definition of the determinant.

4.1.4 Derivatives on a manifold

In order to evaluate more complicated variational forms, it is necessary to be able to evaluate derivatives of functions defined on the manifold. As before, it is sufficient only to consider the case of a basis function defined on a single cell, since all integrals will be decomposed into sums of integrals over basis functions on a single cell.

Suppose we have some functions $\phi(x)$ defined on a cell $K \subset \mathbb{R}^n$ with pullback Φ defined on the reference cell $K_0 \subset \mathbb{R}^m$. The gradient of Φ in reference space is immediate:

$$(\nabla_X \Phi(X))_\tau = \frac{\partial \Phi(X)}{\partial X_\tau}, \quad \tau = 1, \ldots, m. \tag{4.12}$$

Define the tangent space of cell K as the image of the corresponding Jacobian $\mathbf{J}$ over reference space; thus, any $\mathbf{v}$ in the tangent space can be written as $\mathbf{v} = \mathbf{J}\mathbf{V}$ for some $\mathbf{V}$ in the reference space. We define the gradient of ϕ in physical space $\nabla_x \phi$ via the usual Gateaux directional derivative:

$$\nabla_x \phi(x).\mathbf{v} = \lim_{\epsilon \to 0} \frac{\phi(x + \epsilon\mathbf{v}) - \phi(x)}{\epsilon} \tag{4.13}$$

for any $\mathbf{v}$ in the tangent space.

Assume that the mapping G (pullback map from reference to physical) defined in 4.1.2 is affine and non-degenerate, such that the columns of $\mathbf{J}$ are linearly independent. It follows from the above definition that

$$\begin{cases} \nabla_X \Phi(x).\mathbf{v} = \lim_{\epsilon\to 0} \frac{\Phi(x+\epsilon\mathbf{v})-\Phi(x)}{\epsilon} = \\ \lim_{\epsilon\to 0} \frac{\phi(x+\epsilon\mathbf{v})-\phi(x)}{\epsilon} = \nabla_x \phi(x).\mathbf{v} \end{cases} \tag{4.14}$$

Next, let $J^\dagger$ denote the Moore-Penrose pseudo inverse of J, given in this case by

$$J^\dagger = (J^T J)^{-1} J^T \tag{4.15}$$

where the superscript T denotes the transpose. Then clearly, for $\mathbf{v} = \mathbf{J}\mathrm{V}$,

$$J^\dagger \mathbf{v} = (J^T J)^{-1}(J^T J)V = V \tag{4.16}$$

Inserting Eq.(4.16) into Eq.(4.15), and rearranging, we find that

$$\nabla_x \phi(x).\mathbf{v} = (J^\dagger)^T \nabla_X \Phi(X).\mathbf{v} \tag{4.17}$$

In our implementation, vector quantities are always represented as elements of the n-dimensional space in which the manifold is immersed. In this representation, we additionally require that

$$\nabla_x \phi(x).\mathbf{k} = 0 \tag{4.18}$$

where $\mathbf{k}$ is the unit normal vector to the cell K, and hence we obtain the n-dimensional vector

$$\nabla_x \phi(x) = (J^\dagger)^T \nabla_X \Phi(X) \tag{4.19}$$

From (4.15), it follows that the column space of $(J^{\dagger})^T$ coincides with that of $\mathbf{J}$ which implies that$\nabla_x \phi(x)$ is the tangent space of cell K as expected. For most of the technical details concerning the solution of PDEs on the sphere and other manifolds we refer to [34], [33], [39], [25], and references therein.

Remark 4.1.1. Finite elements in 3D and for manifolds builds on the same ideas and concepts as in 1D and 2D, but there is simply much more to compute because the specific mathematical formulas in 3D are more complicated and the evaluation of dof (degree of freedom) maps also gets more complicated. The manual work is tedious, lengthy, and error-prone so computer automation is a must.

4.2 Framework to solve the state equation on a manifold

The time discrete version of the state equation, after some rearrangement (assuming y^{k-1} is known from computations at the previous time level) is

$$\begin{cases} y^0 = y_0 \\ y^k - \triangle t \mu \triangle y^k = y^{k-1} + \triangle t v^k, \quad k = 1, 2, \dots \end{cases} \tag{4.20}$$

Given y_0, we can solve for y^0, y^1, y^2 and so on.

Here, y^k represents solution at k-th time level, y^0 is the initial condition, μ the coefficient of viscosity and v^k is the control at time level k. We only stick to the whole domain in our experiments, therefore, we formulate all the expressions over Σ. Replacing y^k by y, we have now the following variational form.

$$a_0(y,z) = \int_{\Sigma} y^0 z dx \tag{4.21}$$

$$L_0(z) = \int_{\Sigma} y_0 z dx \tag{4.22}$$

$$a(y,z) = \int_{\Sigma} (yz + dt\nabla_{\Sigma} y.\nabla_{\Sigma} z) dx \tag{4.23}$$

$$L(z) = \int_{\Sigma} (y^{k-1} + dtv^k) z dx \tag{4.24}$$

Instead of solving (4.21) by a finite element method, i.e projecting y_0 onto the finite dimensional space via the problem $a_0(y,z) = L_0(z)$, we could simply interpolate y^0 from y_0. That is, if $y^0 = \sum_{j=1}^{N} Y_j^0 \phi_j$, we simply set $Y_j = y^0(x_j, y_j)$, where (x_j, y_j) are coordinates of node number j.

Remark 4.2.1 There are two costly operations in the time loop assembly of the right hand side and solution of the linear system via solve call in FEniCS. The assembly process involves work proportional to the number of degress of freedom N, while the solve operation has a work estimate of $O(N^{\alpha})$ for some $\alpha \geq 1$. As $N \to \infty$, the solve operation will dominate for $\alpha > 1$, but for the values of N typically used on smaller computer, the assembly step may still represent a considerable part of the total work at each time level. Avoiding time assembly can therefore contribute to a significant speed-up of a finite element code of the state equation.

To avoid repeated assembly, Let's look at the right hand side

$$\boxed{L(z) = \int_{\Sigma} (y^{k-1} + dtv^k) z dx} \tag{4.25}$$

which varies in time through y^{k-1}, v^k, and possibly also with dt, if the time-step is adjusted during simulation. The technique for avoiding repeated assembly consists of expanding the finite element functions in sum over the basis functions ϕ_i to identify matrix vector product, that build up the complete system.

$$L(z) = \sum_{j=1}^{N} Y_j^{k-1} \int_{\Sigma} \phi_j \hat{\phi}_i dx + dt \sum_{j=1}^{N} V_j^k \int_{\Sigma} \hat{\phi}_i \phi_j dx \tag{4.26}$$

$\hat{\phi}_i$ is the basis for test function. Introducing $M_{ij} = \int_{\Sigma} \hat{\phi}_i \phi_j dx$ which can be further transformed in terms of expression over the reference triangle using expression (4.8) and (4.9).

$$\left\{ \begin{array}{l} \int_{\Sigma} \hat{\phi}_i \phi_j dx = \sum_{K \in \mathcal{T}} \int_K \phi_i(G_K(X)) \phi_j(G_K(X)) dx \\ = \sum_{K \in \mathcal{T}} \int_{K_0} \Phi_i(X)) \Phi_j(X) |J_K| dx \end{array} \right. \tag{4.27}$$

In the above expression, J is the jacobian of the transformation G_K and $|J_K|$ is the Jacobian determinant. Thus (4.26) can be written as

$$L(z) = \sum_{j=1}^{N} M_{ij} Y_j^{k-1} + dt \sum_{j=1}^{N} M_{ij} V_j^k \tag{4.28}$$

which is nothing but a matrix vector product.

$$MY^{k-1} + dtMV^k \tag{4.29}$$

where $Y^{k-1} = \{Y_1^{k-1}, Y_2^{k-1}, \dots, Y_N^{k-1}\}$ and $V^k = \{V_1^k, V_2^k, \dots, V_N^k\}$. Similarly, We can find the coefficient matrix. Inserting $z = \hat{\phi}_i$ and $y^k = \sum_{j=1}^{N} Y_j^k \phi_j$ in (4.23)

$$\left\{ \begin{array}{l} a(y,z) = \sum_{j=1}^{N} Y_j^k \int_{\Sigma} \hat{\phi}_i \phi_j + dt \sum_{j=1}^{N} Y_j^k \int_{\Sigma} \nabla \hat{\phi}_i \cdot \nabla \phi_j = \\ \sum_{j=1}^{N} \sum_{K \in \mathcal{T}} Y_j^k \int_K \hat{\phi}_i \phi_j dx + dt \sum_{j=1}^{N} \sum_{K \in \mathcal{T}} \int_K \nabla \hat{\phi}_i \cdot \nabla \phi_j = \\ \sum_{j=1}^{N} M_{ij} Y_j^k + dt \sum_{j=1}^{N} \mathcal{K}_{ij} Y_j^k = MY^k + dt\mathcal{K}Y^k \end{array} \right. \tag{4.30}$$

where

$$\boxed{M_{ij} = \sum_{K \in \mathcal{T}} \int_{K_0} \Phi_i(X) \Phi_j(X) |J_K| dX} \tag{4.31}$$

and using equation (4.17)

$$\boxed{\mathcal{K}_{ij} = \sum_{K\in\mathcal{T}} \int_{K_0}((J^{\dagger})^T \nabla_X \phi_i(X)) \cdot ((J^{\dagger})^T \nabla_X \phi_j(X))|J_K| dX} \tag{4.32}$$

In the above expression, (4.32) $J^{\dagger}$ is the Moore-Penrose pseudo inverse of J.

4.3 Dolfin-adjoint framework for optimal control

We will present in this section the recent development of dolfin-adjoint Farrel et al. [11] for the optimal control. Its main features are an intuitive mathematical interface, a high degree of automation, and an efficient implementation of the generated adjoint model. The framework is based upon the extension of domain-specific language for variational problems to cleanly express complex optimization problems in a compact, high-level syntax. Based on the high-level representation, the FEniCS code generation technology generates parallel optimized, low-level C++ code for the forward and adjoint systems. The computed functional and gradient information is passed to the optimization algorithm to update the parameter values.

4.3.1 The optimization framework

The core of the framework relies on two software components: first, the FEniCS system is used to solve the forward and adjoint PDEs. Second, libadjoint and dolfin-adjoint automatically derives the associated adjoint system for gradient information which solves our control problem. The libadjoint and dolfin-adjoint libraries enable the automatic derivation of tangent linear and adjoint models from forward models written in DOLFIN. The purpose of libadjoint to facilitate the development of tangent linear and adjoint models based on the fundamental abstraction of considering the forward model as a sequence of equation solves. Based on this abstraction, the library builds a symbolic description of the forward model, the tape, from which it can automatically derive the symbolic representation of the associated tangent linear and adjoint systems. The optimization process consists of iteratively evaluating the functional of interest at different points in the parameter(control) space. The framework transform the selected linear-quadratic optimal control problems into quadratic optimization problems in Hilbert spaces. The numerical methods employed in the framework is the one proposed in Tröltzsch [41].

4.3.2 Pseudo-code

The program outline to solve the control problem on the manifold is as follows:

- read input file/make input file: get the number of elements, the number of nodes, and the physical parameters ((final time T), discretization parameter, and viscosity coefficient).
- read mesh/make mesh
- while (time $(t) < T$) : time loop (to solve forward equation)
  - determine time step.
  - do intermediate time step.
  - calculate solution for the forward problem (y^k) at time k.
- create the tape for the adjoint equation.
- calculate the gradient.

- implement the optimization algorithm.

Remark 4.3.1 We cannot always just discretize our equations, posed it over as a problem in $\mathbb{R}^n$, solve it with standard optimization software, and expect it to solve for large problems. The algorithm requires knowledge about the function space involved because function space have their own geometric properties. If we ignore the structure of the function space, then mesh-dependent results are possible. Therefore, the algorithm must have information about the function space involved. Riesz map is used to remove the mesh-dependency and handles functional analytic details of the problem. We implemented the SLSQP [23] algorithm with reisz map from python PyOpt package [36] to remove mesh-dependence in all the numerical experiments in the following sections.

4.4 Numerical experiments

In this section, we will present numerical results obtained by the technique just described. We will start by investigating the control of diffusion phenomena on a torus. We have discussed controllability on the surface of a torus in great detail in Chapter 3. We therefore present small details of results for controllability on a torus. For all experiments, we used second order approximations for the unknowns. We will however discuss the control of diffusion phenomenon on sphere in great detail for small time T. Simulations are parallelized for faster computation.

4.4.1 Numerical result for a torus

We start with a simple example to analyse the controllability properties on a torus. For this alternative approach, we import the mesh of the torus to the finite element solver.

The control is supported on the whole domain.

Table 4.1: Parameters used in the controllability experiment with the second approach

Physical Parameters	μ	1
	T	0.1
	R	2
	r	1
Penalty parameters	k	10^2
Time discretization parameter	$\triangle t$	10^{-2}

We choose our target function y_T as.

$$\boxed{y_T = cos(x) + sin(y)} \tag{4.33}$$

We first present the visualizaton of the functional values at different iterations as seen in Figure 4.2. The algorithm starts with a zero value for the control e.g. $\mathbf{v}^0 = 0$. The corresponding functional value $J_h^{\triangle t}(0)$ is 307.4970 which is close to the analytical value and the algorithm converges to a local optimum after 72 iterations. We plotted only one visualization result for torus in Figure 4.4.

Remark 4.4.1.1 The mesh on torus is generated using a major radius of $R = 2.0$ and minor radius of r=1. We follow the strategies of dolfin mesh-generation and python-enumeration on θ and ϕ axis and the map as discussed in Chapter 3. We emphasize that constructing the mesh with this approach is similar to that we discussed in the previous section. However, the results are more appealing because section the control is visible on the surface of the torus. We construct a routine which given an expression in polar coordinates, return one which expects positions in polar coordinates.

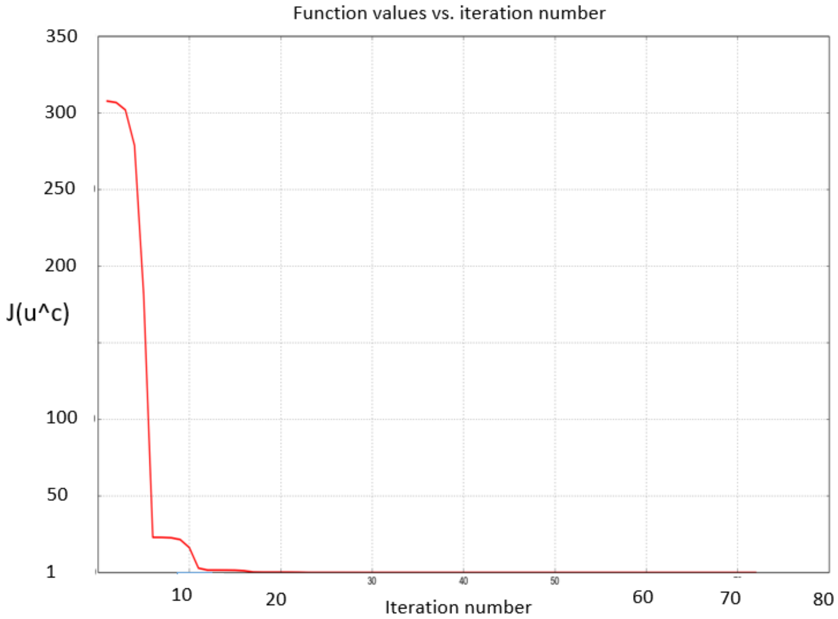


Figure 4.2: Functional values vs. iterations for $T = 0.1$, $k = 10^2$, and $\mu = 1$.

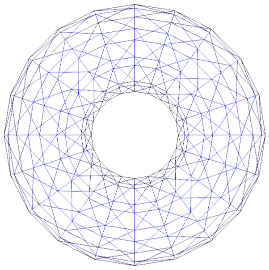
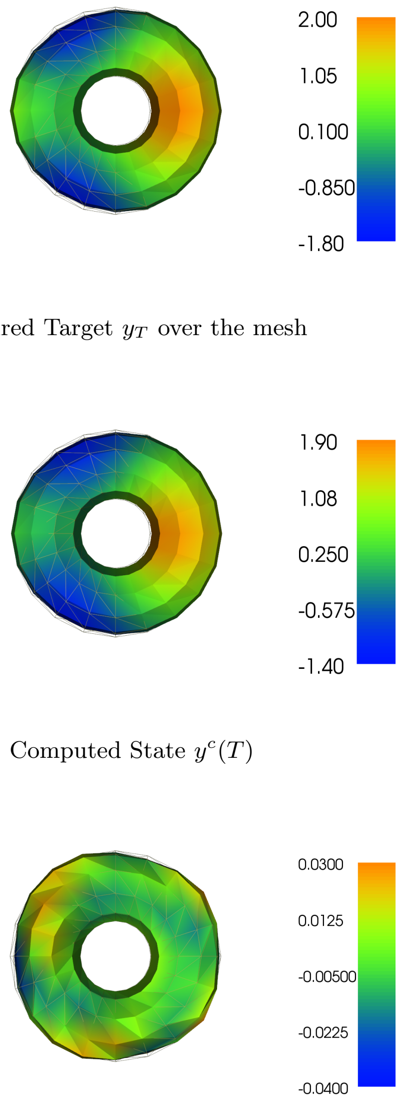


Figure 4.3: Mesh on the torus



(a) Desired Target y_T over the mesh

(b) Computed State $y^c(T)$

(c) Difference between the desired target, y_T, and computed state, $y^c(T)$.

Figure 4.4: Visualization of the numerical results for $T = 0.1$, $k = 10^2$, and μ =1. The color bar indicates the value of the function at mesh coordinates.

4.4.2 Results for a sphere

We chose $T = 0.1$ and $T = 1$ as final time to investigate controllability on a sphere. The approach is robust because we only had to import the mesh. We chose the following set of parameters for the first numerical experiment.

Table 4.2: Parameters used in experiment for control on a sphere

Physical parameters	μ	1,0.1,100
	T	0.1
	R	2
Penalty parameters	k	$10^2, 10^4$
Time discretization parameter	$\triangle t$	10^{-2}

Our control domain for the first experiment was a tetrahedron in the first octant. We observed in Figure 4.6 a very promising result for $tol = 10^{-13}$ and our computed state y^c and target function y_T looks nearly identical. Table 4.3 shows the summary of convergence results for the different values of μ, k and the described physical parameters. We chose our target function y_T and initial condition y_0 as described in (4.34). We plotted the time evolution of the L^2 norm of the optimal control for very small values of tolerance $tol = 10^{-13}$ and T=0.1 as seen in Figure 4.5.

$$\begin{cases} y_T = sin(z)cos(y), \\ y_0 = 0. \end{cases} \tag{4.34}$$

The algorithm takes (almost 2 days) to converge for a small stopping tolerance on four core processor, even though we chose small time T. We, therefore chose a large tolerance $tol = 10^{-5}$ for the second experiment. We chose the same target function y_T as above and

Table 4.3: Summary of convergence result for control on sphere, tol $=10^{-13}$

μ	k	iterations	Norm u^c	Rel.error
0.1	10^2	287	45.1712	0.0008
1	10^4	1065	65.5171	0.0008

from our previous experience we plan to choose the coefficient of viscosity $\mu = 0.001$ and $\mu = 1$. In Table 4.4 we present the convergence results related to the second experiment for simulation time $T = 1$ and different values of the penalty parameter k. The control is supported on the whole domain(Σ). We observed that for a small viscosity coefficient, the controllability was achieved in a modest number of iterations (10 on average) and cost of the control (L^2 norm) is less because of faster diffusion.

Table 4.4: Summary of convergence results for control on sphere, tol $=10^{-5}$, and $T = 1$

μ	k	iterations	Norm. u^c	Rel.error
0.001	10^2	7	0.8163	0.0295
	10^4	93	0.8206	0.0021
1	10^2	14	11.5660	0.0465
	10^4	223	30.5172	0.0044

4.4.2.1 Visualization of numerical results for the sphere:

The method for generating mesh on the sphere uses a uniformly subdivided cube where each vertex position is normalized and multiplied by the sphere radius. This creates a non uniformly subdivided sphere where the triangles closer to the center of a cube face are bigger than the ones closer to the edges of the cube.

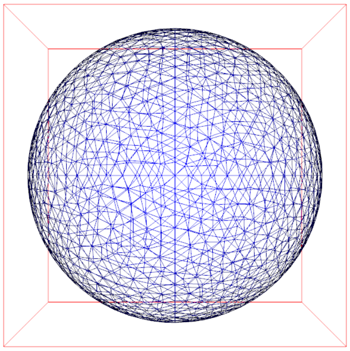


Figure 4.5: Mesh on the surface of sphere

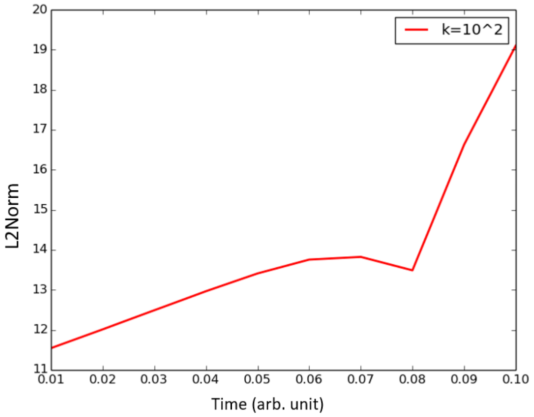
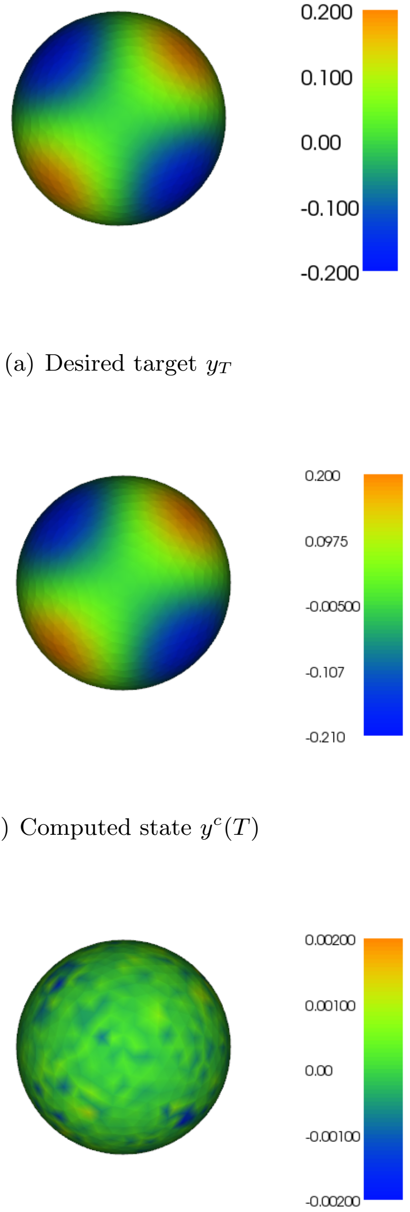


Figure 4.6: Time evolution of the L^2 norm of the optimal control for $T = 0.1$, $tol = 10^{-13}$, and $\mu = 1$



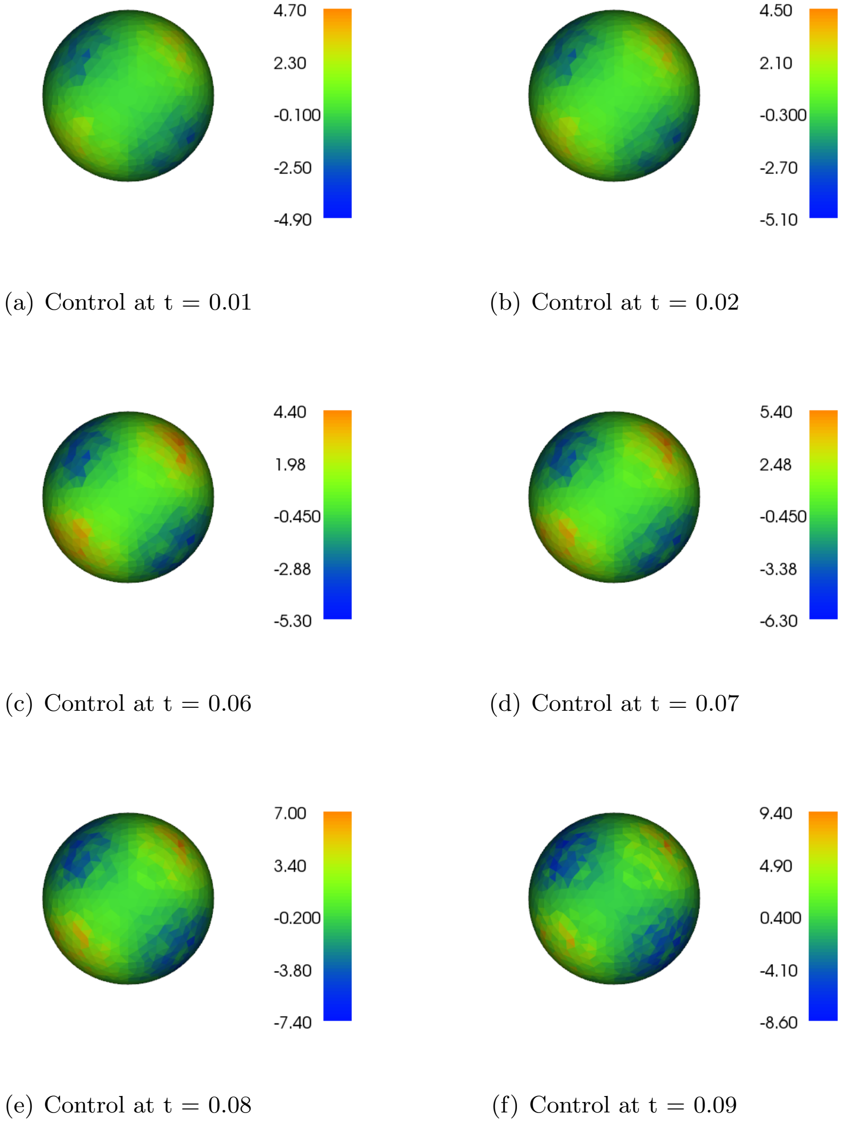
(a) Desired target y_T

(b) Computed state $y^c(T)$

(c) Difference between the desired target, y_T, and computed state, $y^c(T)$.

Figure 4.7: Visualization of numerical results for $T = 0.1$, $k = 10^2$, μ=1, and $\triangle t = 0.01$. The color bar represents the value of the function at mesh coordinates.

We present snapshots of control at different time points for $T = 0.1$, with a penalty parameter $k = 10^2$, and viscosity coefficient $\mu = 0.1$ as seen in Figure 4.7. We also present the snapshots of control for different points of time when the control is implemented on the whole domain with $T = 0.1$, a penalty parameter $k = 10^2$, and viscosity coefficient $\mu = 1$ as seen in Figure 4.9.



(a) Control at t = 0.01

(b) Control at t = 0.02

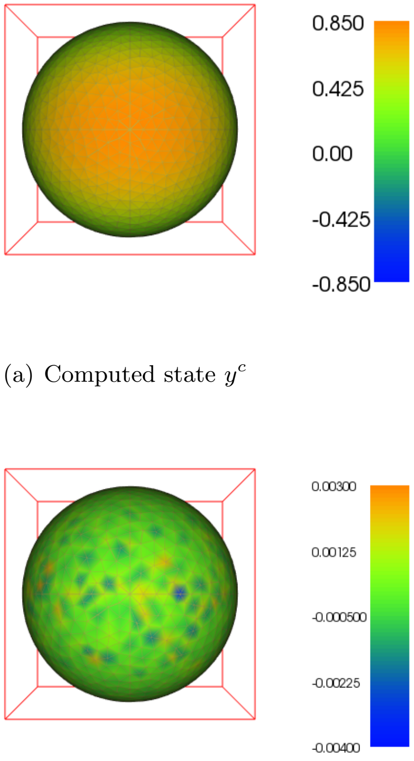
(c) Control at t = 0.06

(d) Control at t = 0.07

(e) Control at t = 0.08

(f) Control at t = 0.09

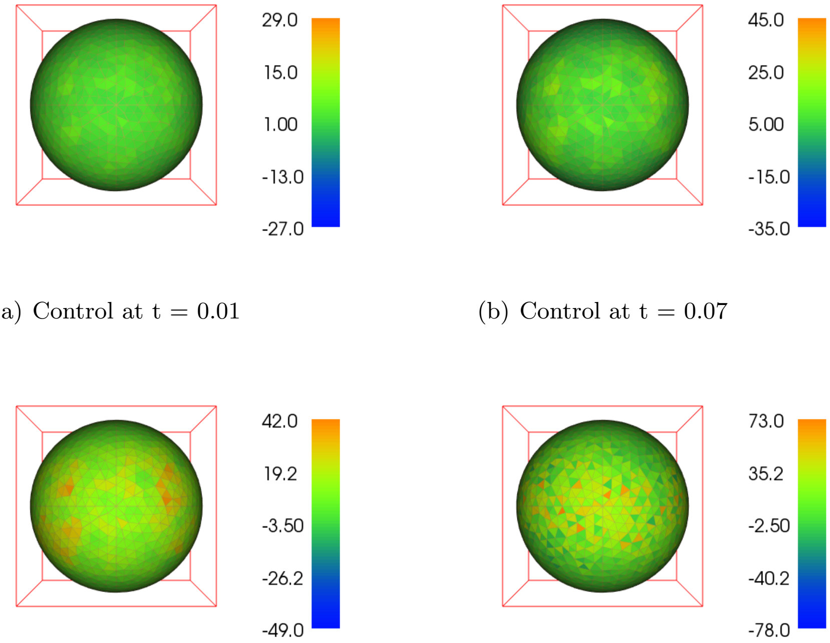
Figure 4.8: Snapshots of the control for the sphere $T = 0.1$, $k = 10^2$, and $\mu = 0.1$. The color bar represents the value of the function at mesh coordinates and tells how strong the control is, in the different regions of the sphere.



(a) Computed state y^c

(b) Difference between the Desired Target and computed state

Figure 4.9: Visualization of numerical results for $T = 0.1, k = 10^2$, and $\mu = 1$. The color bar represents the value of the function at mesh coordinates.



(a) Control at t = 0.01

(b) Control at t = 0.07

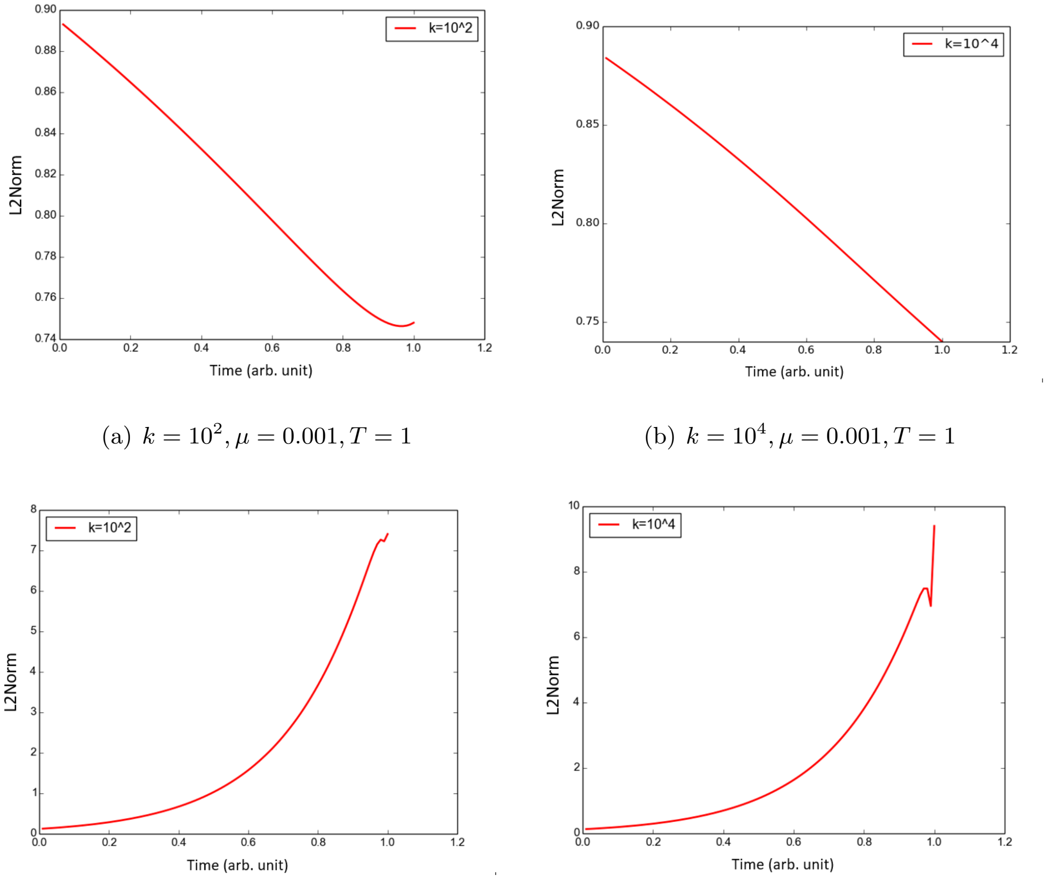
(c) Control at t = 0.08

(d) Control at t = 0.09

Figure 4.10: Snapshots of the optimal control for $T = 0.1$, $k = 10^2$, and $\mu = 0.1$. The color bar represents the value of the function at mesh coordinates.

4.4.2.2 Visualization of the L^2 norm of the optimal control

Finally, we present the evolution of L^2 norm of control for different penalty parameter values (k) and viscosity coefficients μ. We chose $T = 1$ in all the below mentioned visualization. We choose relatively large tolerance $tol = 10^{-5}$ to get faster convergence for these experiments.



(a) $k = 10^2, \mu = 0.001, T = 1$

(b) $k = 10^4, \mu = 0.001, T = 1$

(c) $k = 10^2, \mu = 1, T = 1$

(d) $k = 10^2, \mu = 1, T = 1$

Figure 4.11: Time evolution of the L^2 norm of the optimal control for $k = 10^2, 10^4$, and T $= 1$.

Chapter 5

Control of distributed parameter systems modelled by parabolic variational inequalities of the obstacle type

In this chapter, we will investigate the numerical solution of the control problem modelled by parabolic variational inequalities. The general point of view adopted in this chapter has its roots in the work by R. Glowinski[14]. First, we will introduce the model and describe the solution method. In Section 5.4 and 5.5, we will discuss the discretization of the model problem and then a conjugate gradient algorithm for solving the problem numerically. Finally we will present numerical results of optimal control problem related to variational inequality.

5.1 Introduction

Optimal control problems for variational inequalities have been a subject of interest in the optimal control community starting from the 1980s. The motivation for this study comes from two broad interesting applications.

- Reynolds lubrication(thin film)
- Principles of electro-wetting on dielectric(EWOD).

Principle of electro-wetting on dielectric has applications in solar concentrators, mass spectronomy and electrofluidic displays. These problems are fairly complicated from both the analytical and computational point of view. Our goal here is to discuss the solution of control problems for parabolic inequalities of the obstacle type by taking advantage of the penalty based technique. Using penalty, we will be able to approximate the parabolic variational inequalities by nonlinear parabolic equations and then we will apply the fairly classical method discussed in [13].

5.2 Problem formulation

Let Ω and ω be two bounded domains of $\mathbb{R}^d$, with $d \geq 1$, verifying $\omega \subset \Omega$. The *control problem* that we consider is defined as follows:

$$\begin{cases} Find \ u \in \mathcal{U} \ such \ that \\ J(u) \leq J(v), \forall v \in \mathcal{U}, \end{cases} \tag{5.1}$$

with

$$\mathcal{U} = L^2(\omega \times (0,T)),$$

$0 < T < \infty$, and

$$J(\mathbf{v}) = \frac{1}{2} \int\limits_{\omega \times (0,T)} |\mathbf{v}|^2 dxdt + \frac{k_1}{2} \int\limits_{\Omega \times (0,T)} |y - y_d|^2 dxdt + \frac{k_2}{2} \int\limits_{\Omega} |y(T) - y_T|^2 dx \tag{5.2}$$

In (5.2), we assume that: (i) $k_1, k_2 \geq 0$, with $k_1 + k_2 > 0$; (ii) $y_d \in L^2(\Omega \times (0,T))$ and $y_T \in L^2(\Omega)$; (iii) y is defined from v via the solution of the following *parabolic variational inequality*

$$\begin{cases} y(0) = y_0(\in K); \\ a.e \;\; on \;\; (0,T) \;\; , y(t) \in K \;\; and \\ \langle \frac{\partial y(t)}{\partial t}, z - y(t) \rangle + \int\limits_{\Omega} \overline{A} \nabla y(t).\nabla(z - y(t)) \\ \geq \langle f(t), z - y(t) \rangle + \int\limits_{\omega} v(t)(z - y(t))dx, \forall z \in K, \end{cases} \tag{5.3}$$

where (using $\phi(t)$ denote the function $x \to \phi(x,t)$) and:

- $\overline{A} \in (L^\infty(\Omega))^{d \times d}, \exists \; \alpha > 0$ such that $\overline{A}(x)\xi.\xi \geq \alpha |\xi|^2, \forall \xi \in mathbbR^d$, a.e in Ω,
- ·The convex set K is defined by

$$K = \{z | z \in H_0^1(\Omega), z \geq \phi \;\; a.e \;\; on \;\; \Omega\}, \tag{5.4}$$

with $\phi \in C^0(\Omega) \cap H^1(\Omega)$ and $\phi|_{\partial\Omega} \leq 0$; K is closed and non empty in $H_0^1(\Omega)$ since it contains $\phi^+(= max(0, \phi))$,

- $\langle \cdot, \cdot \rangle$ denotes the duality pairing between $H^{-1}(\Omega)$(the dual space of $H_0^1(\Omega)$) and $H_0^1(\Omega)$,

- $f \in L^2(0,T; H^{-1}(\Omega))\cdot$

Proving the existence of solution to the elliptic analogues of problem (5.1) is easy; on the other hand, proving the existence of solution to (5.1) is more complicated task since it requires using spaces like $L^p(0,T;X)$, where X is a banach space. Assuming that the solution do exist, we will discuss a method to approximate them.

5.3 Penalty approximation of the control problem (5.1)

Let ϵ be a positive parameter. We approximate the control problem (5.1) by

$$\begin{cases} Find\ \ u_\epsilon \in \mathcal{U}\ \ such\ \ that \\ J_\epsilon(u_\epsilon) \leq J_\epsilon(\mathbf{v}), \forall \mathbf{v} \in \mathcal{U}, \end{cases} \tag{5.5}$$

with

$$J_\epsilon(v) = \frac{1}{2} \int\limits_{\omega\times(0,T)} |v|^2 dxdt + \frac{k_1}{2} \int\limits_{\Omega\times(0,T)} |y - y_d|^2 dxdt + \frac{k_2}{2} \int\limits_{\Omega} |y(T) - y_T|^2 dx, \tag{5.6}$$

where, in (5.6), k_1, k_2, y_d and y_T are like in (5.2), and where y is obtained from v via the solution of the following *nonlinear parabolic equation.*

$$\begin{cases} y(0) = y_0(\in K); \\ a.e \;\; on \;\; (0,T), \;\; y(t) \in H_0^1(\Omega) \;\; and \\ \langle \frac{\partial y(t)}{\partial t}, z\rangle + \int\limits_\Omega \overline{A}\nabla y(t).\nabla z dx - \epsilon^{-1} \int\limits_\Omega ((y(t)-\phi)^-)^2 z dx = \\ \langle f(t), z\rangle + \int\limits_\omega v(t) z dx, \forall z \in H_0^1(\Omega)\cdot \end{cases} \tag{5.7}$$

consider the nonlinear operator $\boldsymbol{A}$ defined by

$$\boxed{\boldsymbol{A(z)} = -\nabla.\overline{A}\nabla z - \epsilon^{-1}((z-\phi)^-)^2} \tag{5.8}$$

$\boldsymbol{A}$ is continuous from $H_0^1(\Omega)$ into $H^{-1}(\Omega)$; it is also strongly monotone since it verifies

$$\begin{cases} \langle \boldsymbol{A}(y_2) - \boldsymbol{A}(y_1), y_2 - y_1\rangle = \int\limits_\Omega \overline{A}\nabla(y_2 - y_1)\cdot(y_2 - y_1)dx- \\ \epsilon^{-1} \int\limits_\Omega [((y_2-\phi)^-)^2 - ((y_1-\phi)^-)^2](y_2 - y_1)dx \\ \geq \int\limits_\Omega \overline{A}\nabla(y_2 - y_1)\cdot\nabla(y_2 - y_1)dx \\ \geq \alpha \int\limits_\Omega |\nabla(y_2 - y_1)|^2 dx, \;\; \forall \;\; y_1, y_2 \in H_0^1(\Omega). \end{cases} \tag{5.9}$$

Due to the monotonicity of the above operator $\boldsymbol{A}$, (5.7) has a unique solution which follows from [24], [28] and [8]. In order to solve the control problem iteratively (using a *conjugate gradient* algorithm for example) it may be most useful to be able to compute the differential $DJ_\epsilon(\mathbf{v})$ of J_ϵ at $\mathbf{v}$. We will find it using perturbation method as discussed in [13].

5.3.1 Computation of $DJ_k(v)$:optimality condition

Suppose that δv is a perturbation of v in $\mathcal{U} = L^2(0,T;H_0^1(\Omega))$ then,

$$\begin{cases} \delta J_\epsilon(v) = \int\limits_{\omega\times(0,T)} DJ_\epsilon(v)\delta v dxdt = \\ \int\limits_{\omega\times(0,T)} v\delta v dxdt + k_1 \int\limits_{\Omega\times(0,T)} (y-y_d)\delta y dxdt + k_2 \int\limits_{\Omega} (y(T)-y_T)\delta y(T)dx, \end{cases} \tag{5.10}$$

where, in (5.10) δy is the solution of the following *parabolic equation*(obtained by perturbation of (5.7))

$$\begin{cases} \delta y(0) = 0; \\ a.e\ \ on\ \ (0,T),\ \ \delta y(t) \in H_0^1(\Omega)\ \ and \\ \langle \frac{\partial}{\partial t}\delta y(t), z\rangle + \int\limits_\Omega \overline{A}\nabla \delta y(t).\nabla z dx + 2\epsilon^{-1}\int\limits_\Omega (y(t)-\psi)^- \delta y(t) z dx = \\ \int\limits_\omega \delta v(t) z dx, \forall z \in H_0^1(\Omega). \end{cases} \tag{5.11}$$

Let us consider a function p defined over $\Omega \times (0,T)$, such that

$$\{p, \frac{\partial p}{\partial t}\} \in L^2(0,T;H_0^1(\Omega)) \times L^2(0,T;H^{-1}(\Omega)), \tag{5.12}$$

a property which implies $p \in C^0([0,T];L^2(\Omega))$ as in [29]. Take $z = p(t)$ in (5.11) and integrate the resulting relation from $t=0$ to $t=T$; we obtain then

$$\begin{cases} \int\limits_\Omega p(T)\delta y(T)dx - \int\limits_0^T \langle \frac{\partial p(t)}{\partial t}, \delta y(t)\rangle dt\ \ + \int\limits_{\Omega\times(0,T)} \overline{A}^t \nabla p.\nabla \delta y dxdt + \\ 2\epsilon^{-1} \int\limits_{\Omega\times(0,T)} (y-\phi)^- p\delta y dxdt = \int\limits_{\omega\times(0,T)} p\delta v dxdt. \end{cases} \tag{5.13}$$

Suppose that p is the (unique) solution of the following linear parabolic equation(the *adjoint equation*)

$$
\begin{cases}
p(t) \in H_0^1(\Omega), \;\; a.e \;\; on \;\; (0,T), \\
p(T) = k_2(y(T) - y_T), \\
-\langle \frac{\partial p(t)}{\partial t}, z \rangle + \int\limits_\Omega \overline{A}^t \nabla p(t).\nabla z dx + 2\epsilon^{-1} \int\limits_\Omega (y(t) - \phi)^- p(t) z dx \\
= k_1 \int\limits_\Omega (y(t) - y_d(t)) z dx, \forall z \in H_0^1(\Omega).
\end{cases}
\tag{5.14}
$$

Taking $z = \delta y(t)$ in (5.14) and combining with (5.10) and (5.13), we obtain

$$
\boxed{\int\limits_{\omega \times (0,T)} DJ_\epsilon(v) \delta v dx dt = \int\limits_{\omega \times (0,T)} (v + p) \delta v dx dt}
\tag{5.15}
$$

that is,

$$
\boxed{DJ_\epsilon(v) = v + p|_{\omega \times (0,T)}}
\tag{5.16}
$$

Remark 5.3.1. The computation of $DJ_\epsilon(v)$ will provide a guideline, when computing (in section 5.4) the time discrete analogue of $DJ_\epsilon(v)$.

Remark 5.3.2. The operator form of (5.14) (a weak formulation is given by)

$$
\begin{cases}
-\frac{\partial p}{\partial t} - \nabla.\overline{\boldsymbol{A}}^t \nabla p + 2\epsilon^{-1}(y - \psi)^- p = k_1(y - y_d) \;\; in \;\; \Omega \times (0,T), \\
p(T) = k_2(y(T) - y_T) \\
p = 0 \;\; on \;\; \partial\Omega \times (0,T),
\end{cases}
\tag{5.17}
$$

which is (a relatively) simple linear parabolic equation.

From a practical point of view, the numerical solution of problem requires its *space-time discretization.* The variational approach we took above makes easy the *space approximation* of (5.5), if one uses, for example, the finite element techniques discussed in [13] . The *time discretization* requires a more careful attention and will be discussed in the following section.

5.4 Time discretization of the control problem

Let $N \geq 1$. We define the time discretization step by $\triangle t = \frac{T}{N}$ and approximate the penalized control problem (5.5) by

$$\begin{cases} Find\ \mathbf{u}_\epsilon^{\triangle t}(=\{u^n\}_{n=1}^N) \in \mathcal{U}^{\triangle t} such\ that \\ J_\epsilon^{\triangle t}(\mathbf{u}_\epsilon^{\triangle t}) \leq J_\epsilon^{\triangle t}(\mathbf{v})\ \ \forall \mathbf{v}(=\{v^n\}_{n=1}^N \in \mathcal{U}^{\triangle t}, \end{cases} \tag{5.18}$$

where in (5.18)

$$\mathcal{U}^{\triangle t} = (L^2(\omega))^N, \tag{5.19}$$

and

$$J_\epsilon^{\triangle t}(\mathbf{v}) = \frac{\triangle t}{2}\sum_{n=1}^N \int_\omega |v^n|^2 dx + k_1\frac{\triangle t}{2}\sum_{n=1}^N \int_\Omega |y^n - y_d^n|^2 dx + \frac{k_2}{2}\int_\Omega |y^N - y_T|^2 dx, \tag{5.20}$$

where in (5.20), $\{y^n\}_{n=1}^N$ is obtained from $\mathbf{v}$ via the solution of following *time-discrete nonlinear parabolic equation*

$$y^0 = y_0; \tag{5.21}$$

for $n = 1, \ldots, N$, $\{y^{n-1}, v^n\} \to y^n$ via the solution of the following *nonlinear elliptic problem*

$$\begin{cases} y^n \in H_0^1(\Omega), \\ \int\limits_\Omega \frac{y^n - y^{n-1}}{\triangle t} z dx + \int\limits_\Omega \overline{A}\nabla y^n \cdot \nabla z dx - \epsilon^{-1}\int\limits_\Omega ((y^n - \phi)^-)^2 z dx = \\ \langle f^n, z\rangle dx + \int\limits_\omega v^n z dx,\ \ \forall z \in H_0^1(\Omega). \end{cases} \tag{5.22}$$

Using the strict monotonicity and the continuity of the operator

$$z \to \frac{z}{\triangle t} - \nabla \cdot \bar{A} \nabla z - \epsilon^{-1}((z-\phi)^-)^2 : H_0^1(\Omega) \to H^{-1}(\Omega), \tag{5.23}$$

one can easily show that each of the N nonlinear elliptic problems (5.22) has a unique solution.

In order to solve the control problem (5.18) by a conjugate gradient algorithm operating in $\mathcal{U}^{\triangle t}$, We equip $\mathcal{U}^{\triangle t}$ with the inner product $(.,.)_{\triangle t}$ defined (with obvious notation) by

$$(\mathbf{v}, \mathbf{w})_{\triangle t} = \triangle t \sum_{n=1}^{n=N} \int\limits_{\omega} v^n w^n dx, \tag{5.24}$$

and the corresponding norm. We are going to discuss, just below, the computation of the differential $DJ_\epsilon^{\triangle t}$ of $J_\epsilon^{\triangle t}$. To compute $DJ_\epsilon^{\triangle t}(\mathbf{v})$, we follow the approach taken in Section (5.3) to compute $DJ_\epsilon(\mathbf{v})$. Let us consider thus $\mathbf{v} \in \mathcal{U}^{\triangle t}$ and let us denote by $\delta \mathbf{v}$ a perturbation of $\mathbf{v}$. We have then

$$\delta J_\epsilon^{\triangle t}(\mathbf{v}) = (DJ_\epsilon^{\triangle t}(\mathbf{v}), \delta \mathbf{v})_{\triangle t}, \tag{5.25}$$

and also

$$\delta J_\epsilon^{\triangle t}(\mathbf{v}) = \triangle t \sum_{n=1}^{N} \int\limits_{\omega} v^n \delta v^n dx + k_1 \triangle t \sum_{n=1}^{N} \int\limits_{\Omega} (y^n - y_d^n) \delta y^n dx + k_2 \int\limits_{\Omega} (y^N - y_T) \delta y^N dx, \tag{5.26}$$

with $\{\delta y^n\}_{n=1}^N$ verifying the following perturbation of (5.21) and (5.22):

$$\delta y^0 = 0; \tag{5.27}$$

for $n = 1, \ldots, N, \{y^{n-1}, v^n\} \rightarrow y^n$ via the solution of

$$\begin{cases} \delta y^n \in H_0^1(\Omega), \\ \int\limits_\Omega \frac{\delta y^n - \delta y^{n-1}}{\triangle t} z dx + \int\limits_\Omega \overline{A} \nabla \delta y^n . \nabla z dx + \\ 2\epsilon^{-1} \int\limits_\Omega (y^n - \phi)^- \delta y^n z dx = \int\limits_\omega \delta v^n z dx, \ \ \forall z \in H_0^1(\Omega) \end{cases} \tag{5.28}$$

Let us consider $\{p^n\}_{n=1}^N \in (H_0^1(\Omega))^N$; taking $z = p^n$ in (5.28) we obtain, by summation and after multiplying by $\triangle t$,

$$\begin{cases} \triangle t \sum_{n=1}^N \int\limits_\Omega \frac{\delta y^n - \delta y^{n-1}}{\triangle t} p^n dx + \triangle t \sum_{n=1}^N [\int\limits_\Omega \overline{A} \nabla \delta y^n . \nabla p^n dx + \\ 2\epsilon^{-1} \int\limits_\Omega (y^n - \phi)^- \delta y^n p^n dx] = \triangle t \sum_{n=1}^N \int\limits_\omega \delta v^n p^n dx, \end{cases} \tag{5.29}$$

which implies (by *discrete integration by parts*)that

$$\begin{cases} \int\limits_\Omega p^{N+1} \delta y^N dx + \triangle t \sum_{n=1}^N [\int\limits_\Omega \frac{p^n - p^{n+1}}{\triangle t} \delta y^n dx + \\ \int\limits_\Omega \overline{A}^t \nabla p^n \cdot \nabla \delta y^n dx + 2\epsilon^{-1} \int\limits_\Omega (y^n - \phi)^- p^n \delta y^n dx] = \triangle t \sum_{n=1}^N \int\limits_\omega p^n \delta v^n dx, \end{cases} \tag{5.30}$$

with p^{N+1} still undetermined. Suppose that $\{p^n\}_{n=1}^{N+1}$ verifies the following discrete adjoint equation.

$$p^{N+1} = k_2(y^N - y_T); \tag{5.31}$$

for $n = N, \ldots, 1$, $\{y^n, p^{n+1}\} \rightarrow p^n$ via the solution of the following well-posed *linear elliptic problem*

$$\begin{cases} p^n \in H_0^1(\Omega), \\ \int\limits_\Omega \frac{p^n - p^{n+1}}{\triangle t} z dx + \int\limits_\Omega \overline{A}^t \nabla p^n \cdot \nabla z dx + \\ 2\epsilon^{-1} \int\limits_\Omega (y^n - \phi)^- p^n z dx = k_1 \int\limits_\Omega (y^n - y_d^n) z dx, \forall z \in H_0^1(\Omega). \end{cases} \tag{5.32}$$

Taking z = δy^n in (5.32) and combining with (5.31), (5.26) and (5.25), we obtain

$$(DJ_\epsilon^{\triangle t}(\mathbf{v}), \delta\mathbf{v})_{\triangle t} = \triangle t \sum_{n=1}^{N} \int_\omega (v^n + w^n)\delta v^n dx, \tag{5.33}$$

that is

$$DJ_\epsilon^{\triangle t}(\mathbf{v}) = \{v^n + p^n|_\omega\}_{n=1}^N. \tag{5.34}$$

Remark 5.4.1. The way $DJ_\epsilon^{\triangle t}(\mathbf{v})$ was computed, via the solution of (5.21),(5.22) (5.26), (5.31) and (5.32), suggest that one needs to store $\{y^n\}_{n=1}^N$(in fact its fully discrete analogue obtained by space discretization). Actually a closer look shows that if one operates properly, one needs to store a very small number of snapshots to compute the differential of the cost function.

5.5 Conjugate gradient solution of the problem 5.1

To solve the above control problem numerically, Polak-Ribiere's conjugate gradient algorithm was implemented since $\mathcal{U}^{\triangle t}$ is a hilbert space for the inner product defined by (5.24) and the associated norm. The Polak-Ribiere's conjugate gradient algorithm reads as follows:

Initialization

$$\mathbf{u}_0 (= \{{u_0}^n\}_{n=1}^N) \ \textit{is given in} \ \mathcal{U}^{\triangle t}. \tag{5.35}$$

Solve

$$y_0^0 = y_0; \tag{5.36a}$$

$$\begin{cases} for \;\; n = 1, \ldots, N, \;\; \{y_0^{n-1}, u_0^n\} \rightarrow y_0^n \;\; via \;\; the \;\; solution \;\; of \\ y_0^n \in H_0^1(\Omega), \\ \int\limits_\Omega \frac{y_0^n - y_0^{n-1}}{\triangle t} zdx + \int\limits_\Omega \bar{A} \nabla y_0^n . \nabla zdx - \\ \epsilon^{-1} \int\limits_\Omega ((y_0^n - \phi)^-)^2 zdx = \langle f^n, z \rangle + \int\limits_\omega u_0^n zdx, \;\; \forall z \in H_0^1(\Omega), \end{cases} \tag{5.36b}$$

and

$$p_0^{N+1} = k_2(y_0^N - y_T); \tag{5.37a}$$

$$\begin{cases} for \;\; n = N, \ldots, 1, \;\; \{y_0^n, p_0^{n+1}\} \rightarrow p_0^n \;\; via \;\; the \;\; solution \;\; of \\ p_0^n \in H_0^1(\Omega), \\ \int\limits_\Omega \frac{p_0^n - p_0^{n+1}}{\triangle t} zdx + \int\limits_\Omega \overline{A}^t \nabla p_0^n . \nabla zdx + \\ 2\epsilon^{-1} \int\limits_\Omega (y^n - \phi)^- p_0^n zdx = k_1 \int\limits_\Omega (y^n - y_d^n) zdx; \forall z \in H_0^1(\Omega) \end{cases} \tag{5.37b}$$

Define $\mathbf{g}_0 \in \mathcal{U}^{\triangle t}$ by

$$\boxed{\mathbf{g}_0 = \{u_0^n + p_0^n|_\omega\}_{n=1}^N} \tag{5.38}$$

if $\frac{\| \; \mathbf{g}_0\|_{\triangle t}}{max[1, \| \; \mathbf{u}_0\|_{\triangle t}]} \leq$ tol take $\mathbf{u}_\epsilon^{\triangle t} = \mathbf{u}_0$; otherwise, set

$$\mathbf{w}_0 = \mathbf{g}_0 \tag{5.39}$$

For $k \geq 0$, $\{\mathbf{u}_k, \mathbf{g}_k, \mathbf{w}_k\}$ being known with $\mathbf{g}_k$ and $\mathbf{w}_k$ different from 0, compute $\{\mathbf{u}_{k+1}, \mathbf{g}_{k+1}\}$ and if necessary $\mathbf{w}_{k+1}$ as follows:

Descent direction

$$\begin{cases} \rho_k \in \mathbb{R}_+, \\ J_\epsilon^{\triangle t}(\mathbf{u}_k - \rho_k \mathbf{w}_k) \leq J_\epsilon^{\triangle t}(\mathbf{u}_k - \rho \mathbf{w}_k), \forall \rho \in \mathbb{R}_+ \end{cases} \tag{5.40}$$

and set

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \rho_k \mathbf{w}_k \tag{5.41}$$

Testing the convergence and construction of the new descent direction:

Solve

$$y_{k+1}^0 = y_0; \tag{5.42a}$$

$$\begin{cases} for \;\; n = 1, \dots, N, \;\; \{y_{k+1}^{n-1}, u_{k+1}^n\} \to y_{k+1}^n \;\; via \;\; the \;\; solution \;\; of \\ y_{k+1}^n \in H_0^1(\Omega), \\ \int\limits_\Omega \frac{y_{k+1}^n - y_{k+1}^{n-1}}{\triangle t} z dx + \int\limits_\Omega \bar{A} \nabla y_{k+1}^n . \nabla z dx - \\ \epsilon^{-1} \int\limits_\Omega ((y_{k+1}^n - \phi)^-)^2 z dx = \langle f^n, z \rangle + \int\limits_\omega u_{k+1}^n z dx \;\; , \forall z \in H_0^1(\Omega), \end{cases} \tag{5.42b}$$

and

$$p_{k+1}^{N+1} = k_2 (y_{k+1}^N - y_T); \tag{5.43a}$$

$$\begin{cases} for \;\; n = N, \dots, 1, \;\; \{y_{k+1}^n, p_{k+1}^{n+1}\} \to p_{k+1}^n \;\; via \;\; the \;\; solution \;\; of \\ p_{k+1}^n \in H_0^1(\Omega), \\ \int\limits_\Omega \frac{p_{k+1}^n - p_{k+1}^{n+1}}{\triangle t} z dx + \int\limits_\Omega \overline{A}^t \nabla p_{k+1}^n . \nabla z dx + \\ 2\epsilon^{-1} \int\limits_\Omega (y_{k+1}^n - \phi)^- p_{k+1}^n z dx = k_1 \int\limits_\Omega (y_{k+1}^n - y_d^n) z dx, \;\; \forall z \in H_0^1(\Omega). \end{cases} \tag{5.43b}$$

Define $\mathbf{g}_{k+1} \in \mathcal{U}^{\triangle t}$ by

$$\mathbf{g}_{k+1} = \{u_{k+1}^n + p_{k+1}^n|_\omega\}_{n=1}^N. \tag{5.44}$$

If $\frac{\| \mathbf{g}_{k+1}\|_{\triangle t}}{max[\| \mathbf{u}_{k+1}\|_{\triangle t}, \| \mathbf{g}_0\|_{\triangle t}]} \leq tol$, take $\mathbf{u}_{\epsilon}^{\triangle t} = \mathbf{u}_{k+1}$; otherwise, compute

$$\gamma_k = \frac{(\mathbf{g}_{k+1} - \mathbf{g}_k, \mathbf{g}_{k+1})_{\triangle t}}{\| \mathbf{g}_k\|^2_{\triangle t}} \ \ [Polak - Ribiere's \ \ update] \tag{5.45}$$

and

$$\mathbf{w}_{k+1} = \mathbf{g}_{k+1} + \gamma_k \mathbf{w}_k. \tag{5.46}$$

Do $k+1 \rightarrow k$ and return to (5.40).

The practical implementation of the above algorithm requires:

- The space approximation of the control problem (5.18).
- The solution of the finite dimensional problems approximating (after space approximation) the elliptic problems (5.36b),(5.37b),(5.42b)and (5.43b).
- The solution of the fully discrete analogue of the line search problem (5.40).

Let's assume that $\Omega \subset \mathbb{R}^2$. Concerning the *space approximation* the simplest way to proceed is to approximate ω by a polygonal domain ω_h and then triangulate ω_h using a *finite element triangulation* $\mathcal{T}_h^{\omega}$. Similarly, we triangulate Ω using finite element triangulation $\mathcal{T}_h^{\Omega}$ verifying $\mathcal{T}_h^{\Omega}|_{\bar{\omega_h}} = \mathcal{T}_h^{\omega}$. Figure 5.1 verifies the above assumption for triangulation:

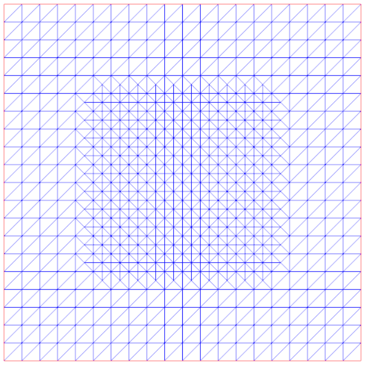


Figure 5.1: Triangulation of $\Omega = (0,1) \times (0,1)$ and $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$

Following, for example, [13], two simple ways to approximate $L^2(\omega)$ are given by

$$\mathcal{V}_h^0 = \{\mathbf{v} | \mathbf{v} \in L^2(\omega_h), \mathbf{v}|_{\mathcal{K}} \in P_0, \forall \mathcal{K} \in \mathcal{T}_h^\omega\}, \tag{5.47}$$

$$\mathcal{V}_h^1 = \{\mathbf{v} | \mathbf{v} \in C^0(\bar{\omega}_h), \mathbf{v}|_{\mathcal{K}} \in P_1, \forall \mathcal{K} \in \mathcal{T}_h^\omega\}, \tag{5.48}$$

where, in (5.47) and (5.48) $P_0(\mathcal{K})$ and $P_1(\mathcal{K})$ are the space consisting of piecewise constant and linear polynomials respectively on mesh cell $\mathcal{K}$. From the above approximation of $L^2(\omega)$ we approximate $\mathcal{U}^{\triangle t}$ by

$$\mathcal{U}_h^{l,\triangle t} = (\mathcal{V}_h^l)^N, \;\; for \;\; l = 0, 1. \tag{5.49}$$

and we approximate the space $H_0^1(\Omega)$ by

$$\boxed{Z_{0h} = \{z | z \in C^0(\bar{\Omega}_h), z|_{\mathcal{K} \in P_1}, \forall \mathcal{K} \in \mathcal{T}_h^\Omega, z = 0 \;\; on \;\; \partial\Omega_h\}} \tag{5.50}$$

where $\Omega_h = \Omega$ if Ω is a polygonal domain of $\mathbb{R}^2$, and a polygonal approximation of Ω otherwise (we assume all the vertices of $\mathcal{T}_h^\Omega$ belong to $\bar{\Omega}$). It is then quite natural to approximate the space K by

$$K_h = \{z | z \in Z_{0h}, z(P) \geq \phi(P), \forall P \in \sum\nolimits_h\} \tag{5.51}$$

Above, $\sum_h$ is the set of the vertices of $\mathcal{T}_h^\Omega$. We approximate then the penalized control problems (5.5) and (5.18) by

$$\begin{cases} Find\ \mathbf{u}_h^{\triangle t} = \{u^n\}_{n=1}^N \in \mathcal{U}_h^{l,\triangle t} such\ that \\ J_{h\epsilon}^{\triangle t}(\mathbf{u}_\epsilon^{\triangle t}) \leq J_{h\epsilon}^{\triangle t}(\mathbf{v}), \forall\ \mathbf{v}(= \{v^n\}_{n=1}^N) \in \mathcal{U}_h^{l,\triangle t}, \end{cases} \tag{5.52}$$

with

$$J_{h\epsilon}^{\triangle t}(\mathbf{v}) = \frac{\triangle t}{2} \sum_{n=1}^{N} \int\limits_{\omega_h} |v^n|^2 dx + \frac{k_1 \triangle t}{2} \sum_{n=1}^{N} \int\limits_{\Omega_h} |y^n - y_{dh}^n|^2 dx + \frac{k_2}{2} \int\limits_{\Omega_h} |y^N - y_{Th}|^2 dx, \tag{5.53}$$

where in (5.53):

- y_{dh} and y_{Th} are approximations of y_d and y_T(obtained by interpolation in general, if y_d and y_T are continuous functions)

- $\{y^n\}_{n=1}^N$ is obtained from $\mathbf{v}$ via the solution of the following fully discrete non-linear parabolic problem

$$y^0 = y_{0h}; \tag{5.54a}$$

$$\begin{cases} For\ \ n = 1, \dots, N,\ \ \{y^{n-1}, v^n\} \to y^n\ \ via\ \ the\ \ solution\ \ of \\ y^n \in Z_{0h}, \\ \int\limits_{\Omega_h} \frac{y^n - y^{n-1}}{\triangle t} zdx + \int\limits_{\Omega_h} \overline{A} \nabla y^n . \nabla zdx - \\ \frac{\epsilon^{-1}}{3} \sum_{P \in \Sigma_{0h}} \int\limits_{\Omega} |\omega_P| ((y^n(P) - \phi(P))^-)^2 z(P) dx = \\ \langle f^n, z \rangle_h + \int\limits_{\omega_h} v^n zdx\ \ , \forall z \in Z_{0h}, \end{cases} \tag{5.54b}$$

where in (5.54) (i)y_{0h} is the approximation of y^0 (obtained by interpolation in general if y^0 is a continuous function).(ii) $\sum_{0h}$ is the set of the vertices of $\mathcal{T}_h^\Omega$ which are not located on $\partial\Omega_h$. (iii) ω_p is the polygonal union of those triangles of $\mathcal{T}_h^\Omega$ which have P as a common vertex and $|\omega_P|$ is the measure of ω_P.

The associated adjoint equation reads as:

$$\begin{cases} p^{N+1} \in Z_{0h}, \\ \int\limits_{\Omega_h} P^{N+1} zdx = k_2 \int\limits_{\Omega_h} (y^N - y_{Th}) zdx\ \ , \forall z \in Z_{0h} \end{cases} \tag{5.55}$$

for $n = N, \dots, 1, \{p^{n+1}, y^n\} \to p^n$ via the solution of following *linear discrete elliptic problem*

$$\begin{cases} p^n \in Z_{0h}, \\ \int\limits_{\Omega_h} \frac{p^n - p^{n+1}}{\triangle t} zdx + \int\limits_{\Omega_h} \overline{A} \nabla p^n . \nabla zdx + \\ \epsilon^{-1} \sum_{P \in \Sigma_{0h}} |\omega_p| ((y^n(P) - \phi(P))^-) p^n(P) z(P) dx = \\ k_1 \int\limits_{\Omega_h} (y^n - y^n_{dh}) zdx; \forall z \in Z_{0h}. \end{cases} \tag{5.56}$$

Concerning the solution of the *nonlinear discrete elliptic problems* (5.54), we advocate Newton method which can be automated by tools in FEniCS. In a nutshell, one can just pass the nonlinear form, the unknown state variable as function object, the essential boundary conditions and the variational form for the Jacobian of the nonlinear form. We will discuss the implementation with minor modification from (5.54) in Chapter 6.

Finally, concerning the solution of the fully discrete analogue of the line search problem (5.40), we advocate the *Back tracking inexact line search using armijo rule* to readily identify a relatively small interval containing the solution, that is, the fully discrete analogue of the solution ρ^k of the one dimensional optimization problem (5.40). The code used for the line search was a direct implementation of the Pseuso-code in the book by Nocedal and Wright[35].

Remark 5.5.1. The function f^n occurring in the right hand side of (5.54) is a convenient approximation of f at $t = n \triangle t$ and we define $f^n \in H^{-1}(\Omega)$ as $f^n = f(n \triangle t)$. Then, since $\langle .,. \rangle_h$ is an inner product on Z_{0h}, one may define f_h^n by

$$\langle f_h^n, z \rangle_h = \langle f^n, z \rangle, \forall z \in Z_{0h}, \;\; f_h^n \in Z_{0h}$$

Discussion: checkpointing

If the forward problem is non-linear, then the solution computed by the forward problem must be available during the execution of the adjoint problem. The adjoint problem depends on the forward solution. If the adjoint problem is solved backwards in time, the forward solution (or the ability to recompute it) must be available for the entire length of forward and adjoint solves.

In large simulations, it quickly becomes impractical to store the entire forward solution through time at once. The alternative to storing the forward solution is to recompute them when necessary from checkpoints stored during the forward run; however a naive recomputation would greatly increase the computational burden of the problem to be solved. Therefore, some balance of storage and re-computation is necessary. This problem has been extensively studied in [6] and a similar memory saving devices has been introduced by Griewank [15] in the context of *Reverse-Mode Automatic Differentiation.* The checkpoint can be thought of as a pointers representing the intermediate states of the evolution. To implement the checkpointing, we invoke revolve library. The routine revolve sets the checkpoints in binomial fashion and the intermediate values are being recalculated instead of being recorded. All coding has been done in python.

5.6 Numerical examples

Test problem

In the following experiments, we will investigate the controllability issues related to variational inequality. For these investigations, we used the data mentioned in Table 5.1. In the numerical experiments, the desired target y_T, initial condition as y_0 and the source term f, are, for simplicity replaced by approximations, y_{Th}, y_{0h}, f_h. For simplicity we choose $\phi = 0$ and diffusion tensor to be identity matrix for all numerical experiments. The primary reason being the difficulty we faced while generating the form for the nonlinear solver in FEniCS [39].

Table 5.1: Parameters used to investigate control of parabolic variational inequality

Physical Parameters	Ω	$(0,1) \times (0,1)$
	ω	$(0,1) \times (0,1)$
Penalty parameters	k_1, k_2	$10^2, 10^4, 10^6$
Time discretization parameter	$\triangle t$	10^{-2}

$$\begin{cases} y_T(x,y) = \exp(-\frac{1}{1-x^2} - \frac{1}{1-y^2}) \\ f = |xy - 0.5| + 0.25 \\ y_d = 0 \end{cases} \tag{5.57}$$

5.6.0.1 Numerical results for $\omega = (0,1) \times (0,1)$

We choose $\triangle t = 0.01$ as the time-step and total time $T = N \times timestep$, where N is the parameter supplied by the developer. For the first experiment we choose $N = 100$ which implies $T = 1$. In these experiments, we have chosen $u_0 = 0$ as initial guess for control, and k_1, k_2 as the value of the penalty parameter. The corresponding numerical results have been summarized in Table (5.2), (5.3), and (5.4) where u^c and y^c denote the computed control and corresponding computed state, respectively, and CGIters is the number of iterations required to achieve the convergence of the conjugate gradient algorithm with tolerance, $tol = 10^{-6}$. If the stopping tolerance is too fine then the optimization algorithm performs badly, albeit the relative error decreases. Norm u^c is $\|u^c\|_{L^2(\omega\times(0,T))}$ and Rel. error denotes the relative error between the desired target y_T and computed state $y^c(T)$ which we denote by $\frac{\|y^c(T)-y_T\|_{L^2(\Omega)}}{\|y_T\|_{L^2(\Omega)}}$. First of all we present the controllability result when control is distributed on the whole domain. However, It is not practically realistic to place the control actuators on the whole domain, we therefore investigated the controllability issues for the case when control is distributed on the subset of the domain. We investigated the effect of ϵ on the convergence when the control is distributed on the whole domain. We present the summary of convergence for $k_1 = 10^4$, $k_2 = 10^2$ and different values of ϵ in Table 5.2. It is not surprising that we get better performance at $\epsilon = 10^{-8}$. It is well known from the proof stated in Mignot et al. [30], Glowinski [14] that the solution y_ϵ of the penalized problem converges to that of variational inequality as $\epsilon \to 0$. However we do not see significant difference in the Rel.error and convergence of the algorithm is not only decided by ϵ. The penalty parameters k_1 and k_2, the desired target y_T and source term f are also key players affecting the Rel. error.

Table 5.2: Summary of convergence for $k_1 = 10^4$, $k_2 = 10^2$, $\omega = (0,1) \times (0,1)$, and T=1

ϵ	k_1	k_2	CGIters	Norm u^c	Rel. error
10^{-2}	10^4	10^2	41	1.0125	0.2551
10^{-3}			40	1.0275	0.2536
10^{-4}			36	1.0377	0.2481
10^{-6}			18	1.2579	0.2421
10^{-8}			17	1.8119	0.1357

We, therefore fix the value of $\epsilon = 10^{-8}$ and study the effect of different values of k_1 and k_2 on the convergence of the conjugate gradient algorithm. We observe that k_1 is the dominating factor concerning the aspect of the optimal control and the convergence behaviour of the algorithm. The performance is better when the value of k_1 is relatively large as compared to the value of k_2, but when we choose the value of k_2 relatively large as compared to the value of k_1 the performance of our conjugate gradient algorithm deteriorates and fails to converge to minimum of J, as soon as the discretization parameters are small enough. The numerical experiments reported indicate that a substancial performance may be obtained after a modest number of iterations. We also investigate the effect of discretization parameter on the convergence of the conjugate-gradient algorithm. We choose $\epsilon = 10^{-8}$, $k_1 = 10^4$, $k_2 = 10^2$ and $T = 1$ to investigate the h-convergence. We observe that as we go on refining the mesh the number of iterations required to achieve

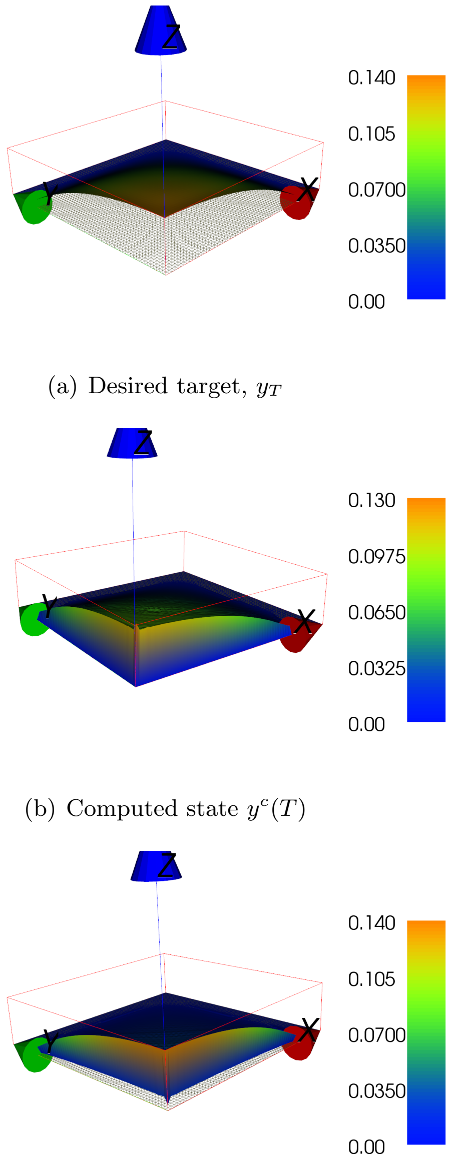
Table 5.3: Numerical results with $\epsilon = 10^{-8}$, and $\omega = \Omega = (0,1) \times (0,1)$

ϵ	k_1	k_2	CGIters	Norm u^c	Rel.error
10^{-8}	10^2	10^2	62	1.5243	0.2481
	10^4	10^2	17	1.8119	0.1357
	10^6	10^2	14	0.1182	0.0180
	10^2	10^4	558	15.2588	0.2965
	10^2	10^6	> 1268	-	-

the convergence decreases. However the values of L^2 norm of the optimal control and Rel. error show stabilizing behaviour. No dramatic differences in Rel.error when the discretization is refined. We present the visualization of computed state y^c, the desired target y_T for $\omega = (0,1) \times (0,1)$ in Figure 5.2 and the snapshots of control at different instants of time for this first experiment in Figure 5.3. We observe that the control creates an even temporal distribution. Finally we present the visualization of time evolution of the L^2 norm of the optimal control for different values of k_1 and k_2 in Figure 5.4.

Table 5.4: Numerical results for h-convergence, $T = 1$, $k_1 = 10^4$, $k_2 = 10^2$, $\triangle t = 0.01$, and $\epsilon = 10^{-8}$

h	Norm u^c	CGIters	Rel.error
$\frac{1}{32}$	3.9345	23	0.3161
$\frac{1}{64}$	1.8119	17	0.1357
$\frac{1}{128}$	1.6192	8	0.1124
$\frac{1}{256}$	1.0290	6	0.1048

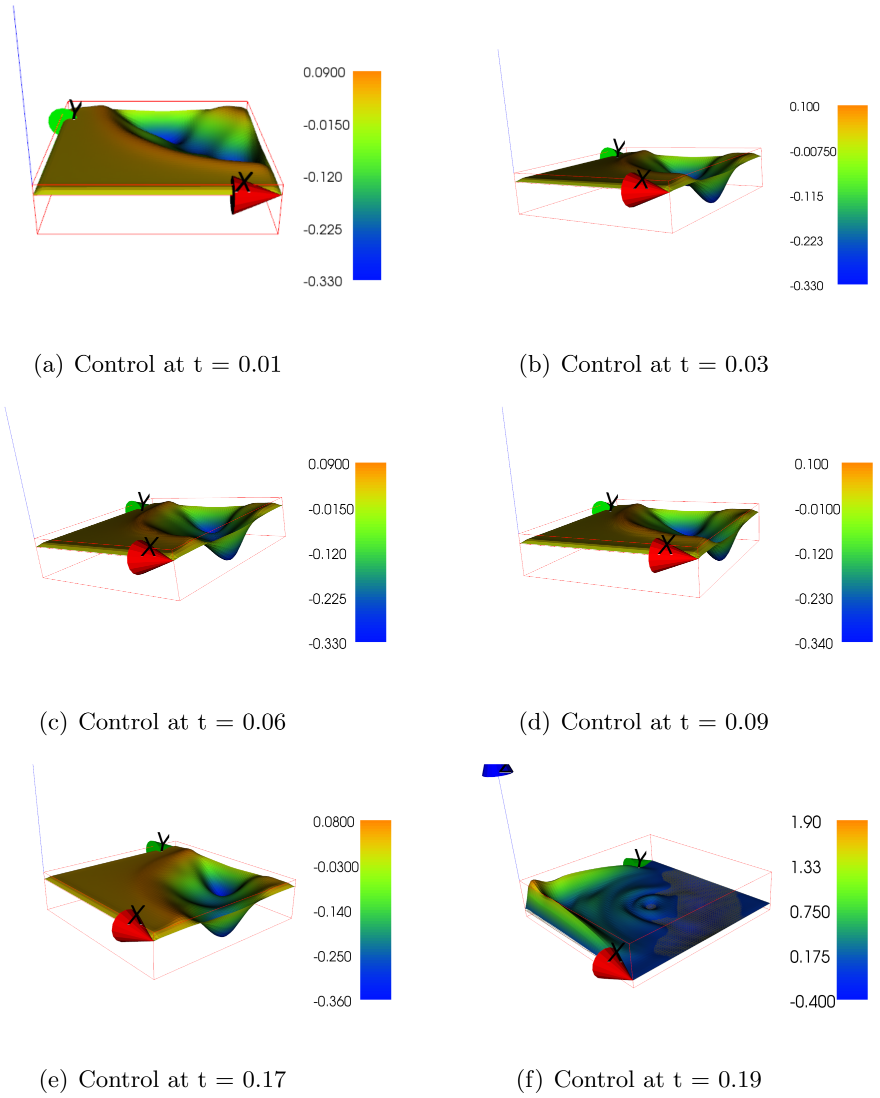


(a) Desired target, y_T

(b) Computed state $y^c(T)$

(c) Difference between the desired target, y_T, and computed state $y^c(T)$

Figure 5.2: Visualization of the desired target, y_T, computed state, $y^c(T)$, for $k_1 = 10^4$, $k_2 = 10^2$, $h = \frac{1}{64}$, $T = 1$, and $\omega = \Omega$. The color bar represents the value of function at mesh coordinates.



(a) Control at t = 0.01

(b) Control at t = 0.03

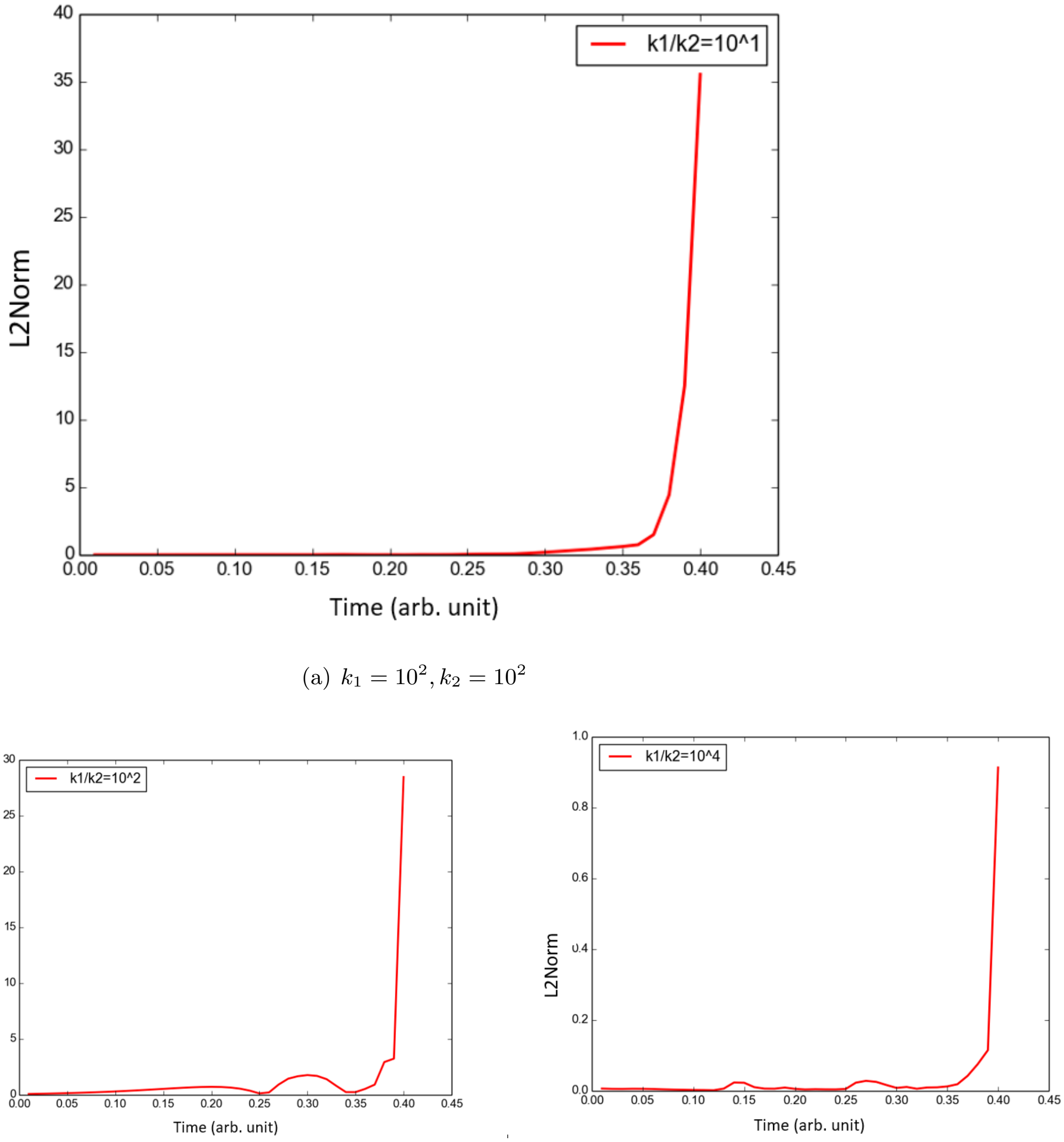
(c) Control at t = 0.06

(d) Control at t = 0.09

(e) Control at t = 0.17

(f) Control at t = 0.19

Figure 5.3: Snapshots of the optimal control at different time instants, $k_1 = 10^6$, $k_2 = 10^2$, $\omega = \Omega$, $T = 0.2$, and $\triangle t = 0.01$. The color bar represents the value of the function at mesh coordinates.



(a) $k_1 = 10^2, k_2 = 10^2$

(b) $k_1 = 10^4, k_2 = 10^2$

(c) $k_1 = 10^6, k_2 = 10^2$

Figure 5.4: Time evolution of the L^2 norm of the optimal control function for $k_1 = 10^4$ and $k_2 = 10^2$, $T = 0.4$, $\triangle t = 0.01$, and $\omega = \Omega$.

5.6.0.2 Numerical results for $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$

In this section we present the controllability result when the control is supported on the sub domain $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$. First, we investigate the effect of epsilon on the controllability for the value of $k_1 = 10^4$ and $k_2 = 10^2$. We choose $tol = 10^{-6}$ for this experiment. We present the summary of convergence results for k_1 and k_2 fixing $\epsilon = 10^{-6}$ in Table 5.5. In Table 5.6 we present the effect of discretization parameter h(mesh width of discretization) on the controllability for $k_1 = 10^4$, $k_2 = 10^2$ and time $T = 1$ when control is implemented on the sub-domain. We plot the computed state $y^c(T)$, the desired target y_T and difference between the computed state and desired target in Figure 5.5. In Figure 5.6 we present the snapshots of the control at different instants of time. Figure 5.6(a) and 5.6(b) is presented differently with an intent to demonstrate the activation of control near zero in the negative y axis. Activation of control near zero in the negative y axis contributes to the low value of the cost of the optimal control.

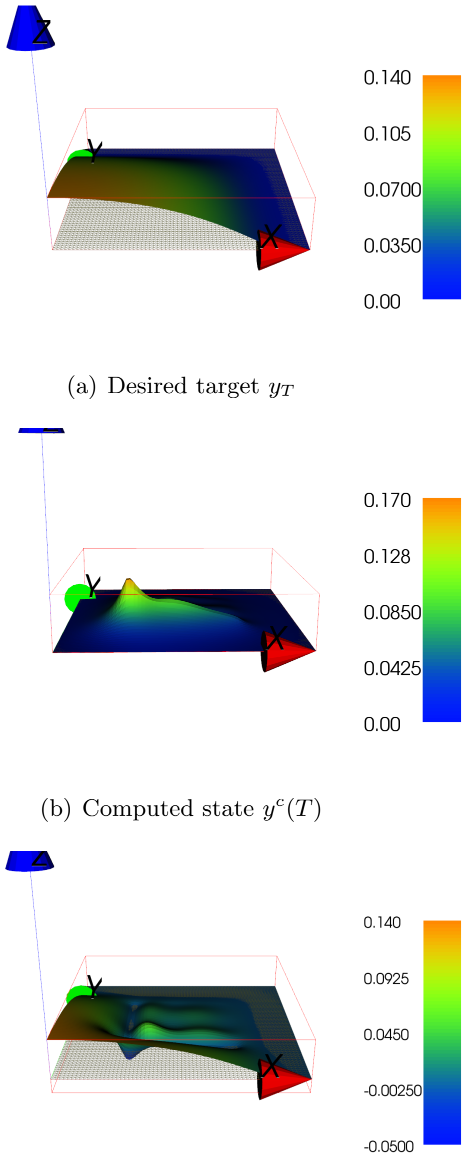
Table 5.5: Summary of convergence results for $k_1 = 10^4$, $k_2 = 10^2$, $T = 0.5$, $\triangle t = 0.01$, and $h = \frac{1}{64}$

ϵ	k_1	k_2	CGIters	Rel.error	Norm u^c
10^{-2}	10^4	10^2	55	0.4240	1.2504
10^{-3}			54	0.4227	1.2536
10^{-4}			48	0.4215	1.2761
10^{-6}			16	0.4110	1.3785
10^{-8}			14	0.4041	1.6192

Table 5.6: Summary of h-convergence for $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$, $\triangle t = 0.01$, T=0.5, $k_1 = 10^4$, and $k_2 = 10^2$

h	Norm u^c	CGIters	Rel.error
$\frac{1}{16}$	3.7868	81	0.4174
$\frac{1}{32}$	2.7387	56	0.4156
$\frac{1}{64}$	1.6192	16	0.4110
$\frac{1}{128}$	1.0290	9	0.4039

We observe that the cost of the control($\|u^c\|_{L^2(\omega \times (0,T))}$) in this case does not show significant difference from the case when the control was distributed on the whole domain. The primary reason for this similarity is the selection of the symmetrical domain which activates the control in similar fashion. We also observe the similar controllability behaviour in case of the sub domain i.e k_1 is the dominating factor. The Rel.error shows stabilizing behaviour and the factor that effects Rel.error are k_1, k_2 and ϵ. The performance is better when we choose high value of $k_1 = 10^6$ and low value of $k_2 = 10^2$. Finally, we present the visualization of the time evolution of the L^2 norm of the optimal control for control supported on sub domain in Figure 5.7.

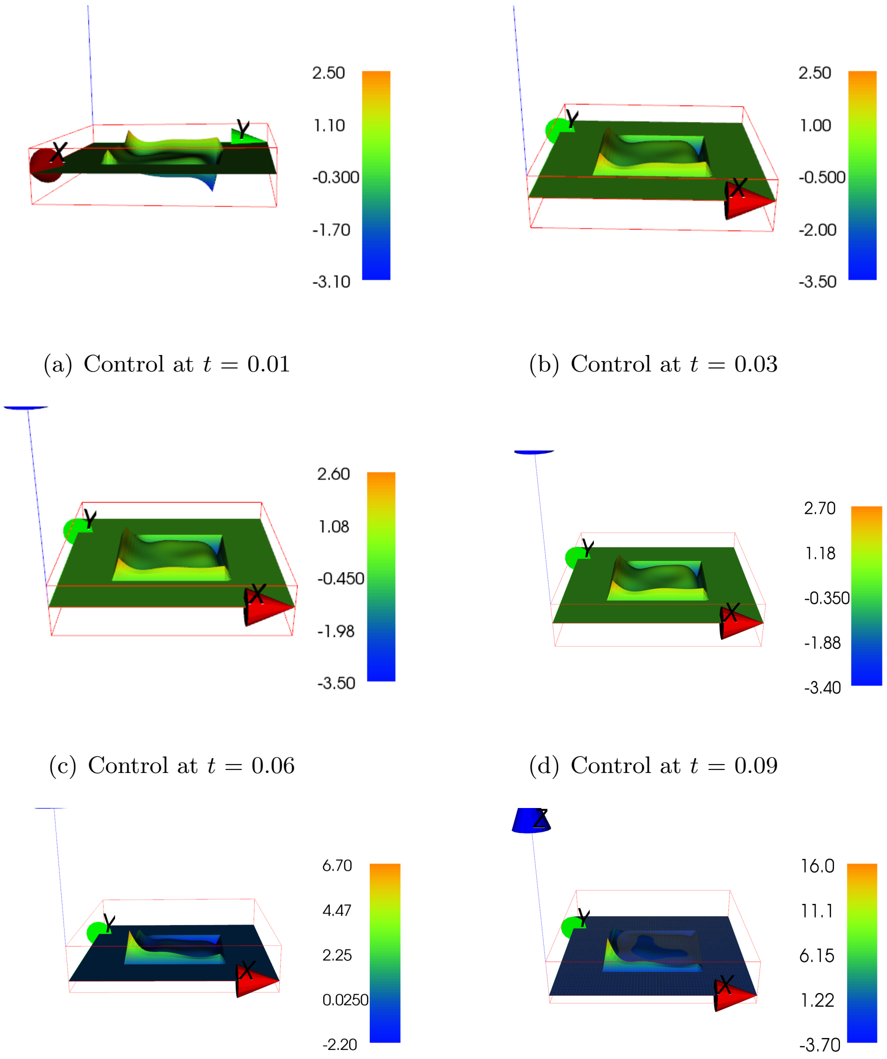


(a) Desired target y_T

(b) Computed state $y^c(T)$

(c) Difference between the desired target, y_T, and computed state $y^c(T)$.

Figure 5.5: Visualization of y^c, y_T and the difference of y^c and y_T for $k_1 = 10^6$, $k_2 = 10^2$, $T = 1$, $h = \frac{1}{64}$, and $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$. The color bar represents the value of the function at mesh coordinates.



(a) Control at $t = 0.01$

(b) Control at $t = 0.03$

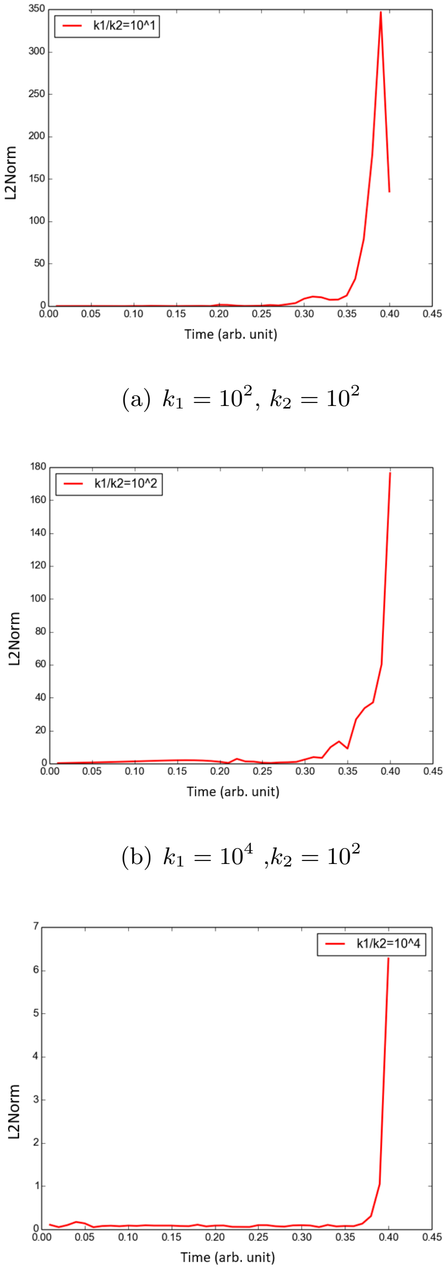
(c) Control at $t = 0.06$

(d) Control at $t = 0.09$

(e) Control at t = 0.17

(f) Control at $t = 0.19$

Figure 5.6: Snapshots of the optimal control at different time instants, $k_1 = 10^6$, $k_2 = 10^2$, $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$, $T = 0.2$, and $\triangle t = 0.01$. The color bar represents the value of the function at mesh coordinates.



(a) $k_1 = 10^2$, $k_2 = 10^2$

(b) $k_1 = 10^4$,$k_2 = 10^2$

(c) $k_1 = 10^6$, $k_2 = 10^2$

Figure 5.7: Time evolution of the L^2 norm of the optimal control function for $k_1 = 10^4$ and $k_2 = 10^2$, $T = 0.4$, $\triangle t = 0.01$, and $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$.

We end this section with controllability results for space discretization used in (5.51) to discretize the control space i.e:

$$\mathcal{V}_h^0 = \{\mathbf{v}|\mathbf{v} \in L^2(\omega_h), \mathbf{v}|_{\mathcal{K}} \in P_0, \forall \mathcal{K} \in \mathcal{T}_h^\omega\}, \tag{5.58}$$

The control is assumed to be piecewise constant. More precisely, u is assumed to be constant on each triangle of the triangulation. In this case, each basis function equals unity on exactly one triangle and zero otherwise. We use the data in Table 5.1 for this experiment. Also the desired target y_T, the source term f and y_d is same as the first experiment. We choose $T = 1$, $\triangle t = 0.01$, $h = \frac{1}{64}$ for this experiment but for visualization purposes we choose small T to demonstrate the activation of control. In order to investigate the effect of the penalty parameter ϵ on the controllability of variational inequality, we choose the value of k_1 and k_2 as 10^6 and 10^2 respectively because of the experience we got after experimenting with previous test cases. The control is supported on $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$ which is symmetrical around the center. We present our investigation for the effect of ϵ on controllability in Table 5.7. We observe in Table 5.8 that the conjugate gradient algorithm takes less iterations to converge if we discretize the control space as mentioned in (5.58). The reason for such behaviour is the appearance of diagonal matrix in the control term after full discretization of the objective functional.

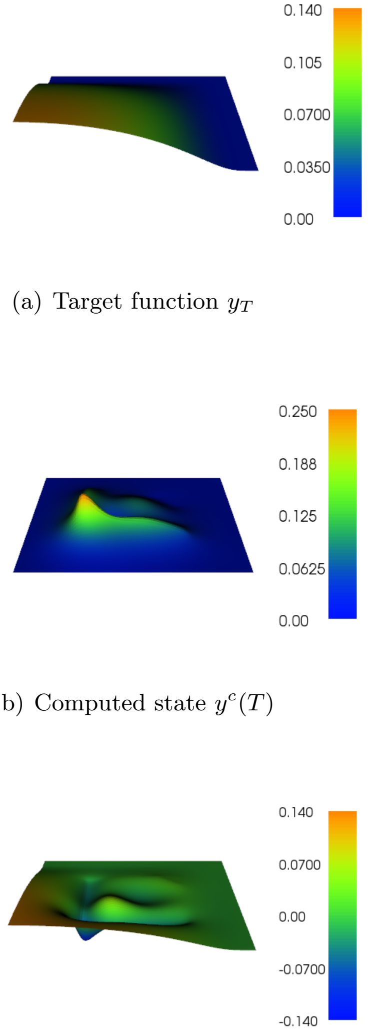
We present the visualization of desired target y_T, computed state $y^c(T)$ and the difference of $y^c(T)$ and y_T in Figure 5.7. The time evolution of the L^2 norm of the optimal control for different values of k_1 and k_2 is shown in Figure 5.10. Figure 5.9 clearly shows the piecewise activation of control when control is supported on the whole domain $\omega = (0, 1) \times (0, 1)$. However, we will not delve into the greater details and only stick to the case where control is distributed on the sub domain $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$.

Table 5.7: Summary of convergence results with $k_1 = 10^6$, $k_2 = 10^2$, $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$, h=$\frac{1}{64}$, and T=1

ϵ	k_1	k_2	CGIters	Norm u^c	Rel.error
10^{-2}	10^6	10^2	70	3.3948	0.0174
10^{-3}	10^6	10^2	64	2.9109	0.0174
10^{-5}	10^6	10^2	23	1.6193	0.0174
10^{-8}	10^6	10^2	17	0.6572	0.0064

Table 5.8: Numerical results with $\epsilon = 10^{-8}$, $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$, T=1, and $h = \frac{1}{64}$

ϵ	k_1	k_2	CGIters	Norm u^c	Rel.error
10^{-8}	10^2	10^2	11	2.1656	0.4758
	10^4	10^2	12	1.3854	0.2962
	10^6	10^2	17	0.6572	0.0064

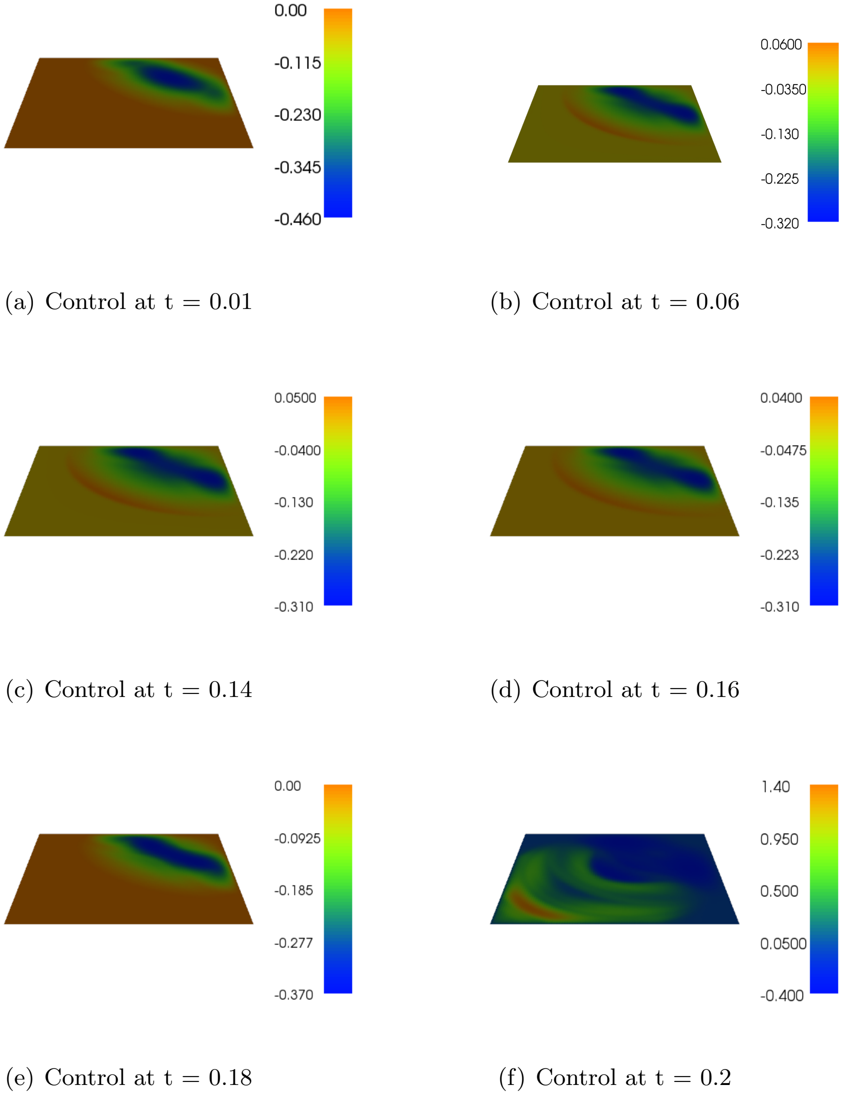


(a) Target function y_T

(b) Computed state $y^c(T)$

(c) Difference between the target function, y_T, and computed state $y^c(T)$

Figure 5.8: Visualization for $\omega = (\frac{1}{4}, \frac{3}{4}) \times (\frac{1}{4}, \frac{3}{4})$,$T$=0.4, $k_1 = 10^4$, and $k_2 = 10^2$. The color bar indicates the value of the function at mesh coordinates while in the case of the difference it represents the value of the difference taken pointwise.



(a) Control at t = 0.01

(b) Control at t = 0.06

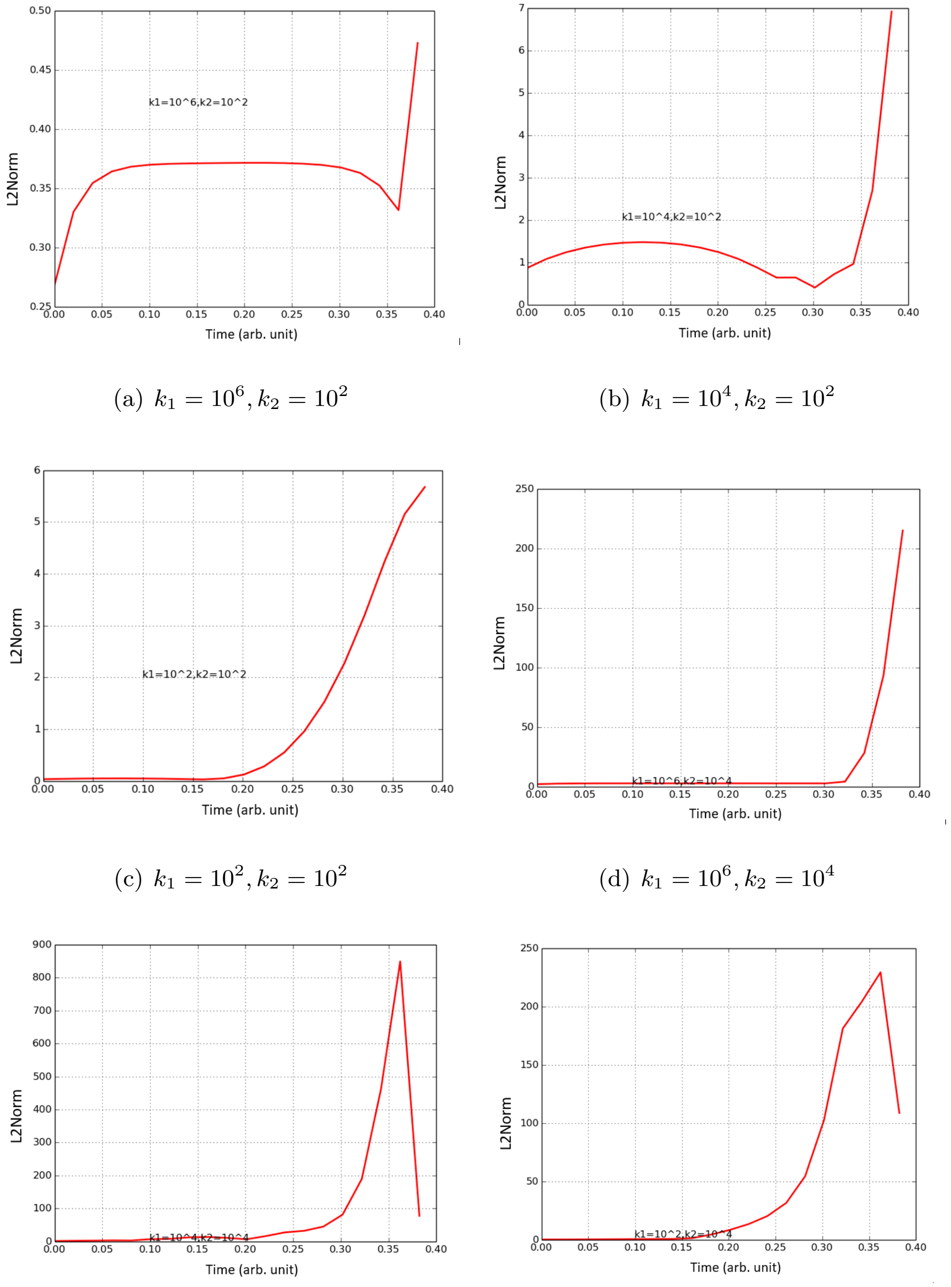
(c) Control at t = 0.14

(d) Control at t = 0.16

(e) Control at t = 0.18

(f) Control at t = 0.2

Figure 5.9: Snapshots of the optimal control at different instants of time, $k_1 = 10^6, k_2 = 10^2$, $\omega = (0,1)^2$, and T=0.2. The color bar represents the value of the function at mesh coordinates.



(a) $k_1 = 10^6, k_2 = 10^2$

(b) $k_1 = 10^4, k_2 = 10^2$

(c) $k_1 = 10^2, k_2 = 10^2$

(d) $k_1 = 10^6, k_2 = 10^4$

(e) $k_1 = 10^4, k_2 = 10^4$

(f) $k_1 = 10^2, k_2 = 10^4$

Figure 5.10: Time evolution of the L^2 norm of optimal control for different values of k_1 and k_2, $\triangle t = 0.01$, $T = 0.4$, and $\omega = (\frac{1}{4}, \frac{3}{4})^2$.

Chapter 6

Implementation in FEniCS

This chapter gives an overview on the implementation of important part of presented algorithm into the finite element library FEniCS. The FEniCS project is a collection of open-source software for the automation of mathematical problems based on differential equations. The most important aspect of FEniCS is the Unified Form Language(UFL, see Alnaes et al. [1], [2]) which allows to generate assembly routines by providing only the variational equation. This is an enormous saving of human resources which would be spent if one implements and tests the assembly routines for bilinear forms by hand. Another big advantage of UFL is the possible use of automatic differentiation(AD) in order to avoid the implementation of long formulas which are associated with errors in calculation and implementation. Besides that, FEniCS offers a wide range of high-end routines that allow the user to solve quite easily in few lines of code. We start with the implementation of periodic boundary condition for the model problem we discussed in Chapter 2 and Chapter 3. We also give a brief description of the linear and nonlinear solver of FEniCS to automate the solution of state equation and adjoint equation in Chapter 3, 4 and 5.

6.1 Implementing periodic boundary condition

To implement periodic boundary condition we create subclass of superclass "**Subdomain**" with "**map**" as one of its method. The function "**map**" maps a coordinate x in domain H to a coordinate y in the domain G, it is used for periodic boundary conditions, so that the right boundary of the domain is mapped to the left boundary and top boundary is mapped to the bottom. When the class is defined, we create the boundary by making an instance of the class. Also notice that in order for periodic boundary conditions to work correctly it is necessary that the mesh nodes on the periodic boundaries match up. A 2D mesh is created using the built-in class "**RectangleMesh**", and we define a finite element function space relative to this space using "**FunctionSpace**". We then pass the instance of the periodic boundary class as an argument to the "**FunctionSpace**". It specifies that all functions in the function space have periodic boundaries defined by the instance of the periodic boundary class. Below mentioned is the code chunk that implements the periodic boundary condition. Note that because of round-off errors, it is often wise to instead specify $(y < \epsilon)$ or $(y > 1 - \epsilon)$ and $(x < \epsilon)$ or $(x > 1 - \epsilon)$ where ϵ is a small number (such as machine precision). We used "DOLFIN_EPS" (machine precision) to represent the small number ϵ which we show in the code chunk.

Listing 6.1: Periodic Boundary Code

```
class PeriodicBoundary(SubDomain):
    # Left boundary is ”target domain” G
      def inside(self, x, on_boundary):
         return bool(x[0] < DOLFIN_EPS and x[0] > -DOLFIN_EPS
         and x[1] < DOLFIN_EPS and x[1] > -DOLFIN_EPS
         and on_boundary)

    # Map right boundary (H) to left boundary (G)
      def map(self, x, y):
         y[0] = x[0] - 2*pi
         y[1] = x[1]
    # Map bottom boundary to the top boundary
      def map(self,x,z):
         z[0]=x[0]
         z[1]=x[1] - 2*pi
pbc =PeriodicBoundary()
# Create mesh and define function space
V = FunctionSpace(mesh, ”CG”, 1,constrained_domain=pbc)
```

6.2 Linear solver in FEniCS

We have presented in great detail the technique for speeding up FEniCS simulator for time-dependent problems in section 4.2 of Chapter 4. However we want to mention that the linear system generated during the time loop for state equations (2.39), (3.49) and the adjoint equations (2.31), (3.50) from Chapter 2 and 3 respectively are solved using DOLFIN solve function. A linear system $Ax = b$ may be solved by calling $solve(A, x, b)$, where A is a matrix and x and b are vectors. Optional arguments may be passed to specify the solver method and preconditioner. Possible values for the solver method and preconditioner depend on which linear algebra backend is used and how that has been configured. In our implementation, we used LU factorization method for solving the linear system.

6.3 Nonlinear solver in FEniCS

First let's recall the nonlinear equation in (5.45)

$$\begin{cases} y^n \in Z_{0h} \\ \int\limits_{\Omega} \frac{y^n - y^{n-1}}{\triangle t} z dx + \int\limits_{\Omega} \overline{A} \nabla y^n . \nabla z dx - \\ \epsilon^{-1} \int\limits_{\Omega} ((y^n - \phi)^-)^2 z dx = \\ \langle f^n, z \rangle + \int\limits_{\omega} v^n z dx, \forall z \in H_0^1(\Omega) \end{cases} \tag{6.1}$$

This implies that for each time step we have to solve a non-linear equation. Another challenge in solving the state equation is to implement the penalty operator since penalty operator is not smooth and the algorithm needs $C^1(\Omega)$ functions for the penalty operator. We introduced C^1 regularization of the penalty operator. Using the definition of the negative part of the function we can now define our penalty term as:

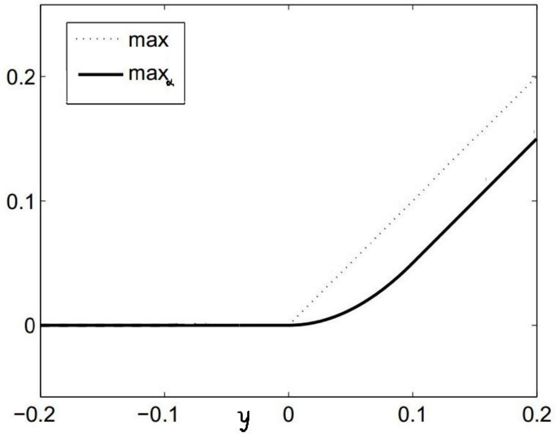
$$\begin{cases} (y - \phi)^- = -min((y - \phi), 0) \\ \qquad\qquad = max(-(y - \phi), 0) \end{cases} \tag{6.2}$$

For a fixed smoothing parameter $\alpha > 0$, we define the following regularized/smoothed operator, if $\phi \equiv 0$, with motivation from [30] and [17]. The defined regularized operator has

Lipschitz property. We choose α= 10^{-4} for numerical experiments.

$$\mathbf{max}_\alpha(0, y) := \begin{cases} y - \frac{\alpha}{2} \ ; y \geq \alpha \\ \frac{y^2}{2\alpha} \ ; y \in (0, \alpha) \\ 0 \ ; y \leq 0 \end{cases} \tag{6.3}$$

To solve equation (6.1), we first created a UFL form corresponding to the nonlinear variational problem. We then pass the created UFL form to ”**NonlinearVariationalProblem**” class of dolfin as:



Listing 6.2: Nonlinear problem

```
problem = NonlinearVariationalProblem(F, y, bcs, J)
solver  = NonlinearVariationalSolver(problem)
solver.solve()
```

The above class creates nonlinear variational problem with a list of boundary conditions. Above, F corresponds to the nonlinear form F(y,z) generated from (6.1). y is the unknown function object, bcs represents the essential boundary conditions(in general a list of DirichletBC objects), and J is the variational form for the Jacobian of F. The Jacobian form is specified which allows the use of a nonlinear solver that relies on the Jacobian (using Newton's method). Finally the nonlinear variational problem is solved by calling the solve method of the NonlinearVariationalProblem class. The maximum number of newton's iteration we chose for all numerical experiments is 50.

Chapter 7

Summary, Conclusions and Future Work

This chapter briefly reviews the main issues connected with the contents of this thesis, discusses some implications and suggestions for future work .

We have in this work studied numerical methods for control of systems modelled by partial differential equations. All control problems considered in this thesis may abstractly be viewed as involving a forward map composed of the following operations:

1. *Application* of the control to a partial differential equation.
2. *Solution* of the state equation.
3. *Observation* of the solution.

The control problem consists of matching the observation, in a least-squares sense, with some given target objective by manipulating the control. More precisely we have studied the control of diffusion phenomenon on manifolds(particularly torus and sphere) and the control of distributed parameter system modelled by parabolic variational inequality of obstacle type.

Section 7.1 summarizes what is covered in Chapters 2-4 regarding computational issues such as discretization, algorithms, and controllability. In Section 7.2, we discuss the controllability issues of variational inequality of obstacle type.

7.1 Discretization and computational issues

7.1.1 Discretization

The computational problem considered in Chapters 2-4 can be in general terms categorized as Linear-quadratic parabolic control problems. The process under investigation is non-stationary, where time comes to play as an additional physical parameter. These processes evolve within the space-time cylinder $Q := \Omega \times (0, T)$. The control function $u = u(x, t)$ acts on the subset of the problem domain Ω. In order to get, from the very beginning, an idea of what sort of optimality condition can be expected, we apply the perturbation technique. Optimality conditions involve another linear PDE, the adjoint PDE which is, however, well posed, since a final condition is prescribed and not an initial condition; if an initial conditions were prescribed instead, we would have the typical case of an ill-posed backward parabolic equation known from the theory of inverse problems. The method, we use often refered to as *discretize then optimize* consists of a full discretization of both the parabolic problem and the cost functional, leading to a (large) optimization problem.

7.1.2 Optimization

In general, iterative methods have to be employed for the numerical solution of optimal-control problem. Most methods of this kind require gradient information which can be obtained by the solution of an adjoint equation. The approach that is strongly advocated in this thesis is to perform the derivation of the adjoint equation and gradient in the fully discrete case. This guarantees that accurate descent directions are provided to the minimization algorithm.

For the control of diffusion phenomenon on manifolds controllability is better achieved when we implement the control actuator on the whole domain than implementing the control on the subset of the problem domain.

While implementing the second approach for the control of diffusion phenomenon on manifolds, we observed that our algorithm does not perform well for larger time and is memory demanding which lead to segmentation fault unless we implement some check-pointing scheme. Technically speaking, this approach is not much different from the first approach since to create the mesh we use the mapping technique and then wrap it to get the desired mesh for torus but because of the available solver in FEniCS for manifolds we wished to give it a try. There is a great difficulty in tuning the parameters with the second approach since every time we wish to experiment we will have to generate the mesh. Also, we want to mention that the optimization algorithm in the optimization framework proposed in chapter-4 is not efficient.

7.1.3 Optimal support of the control.

It appears also interesting to analyze deeper the ill-posedness of the problem with respect to the distribution of the support of the control. We also want to investigate the situation where the support depends on the time variable(e.g chattering control).

7.2 Distributed control of parabolic variational inequality of obstacle type

Chapter 5 attempts to demonstrate the controllability issues with variational inequality which has wide range of applications, including to name a few are Tripology and EWOD. A finite-element-based discretization technique is used for the partial differential equations, and the optimal control is solved by Newton/Conjugate gradient algorithm applied to the associated nonlinear optimization problem.

The Newton/Conjugate algorithm needs gradient information which is computed using the adjoint-equation technique. In line with the discretization strategy advocated throughout the thesis, the gradient and the adjoint equation are derived in the fully discrete case.

Due to the choice of objective function and the nonlinear nature of the state equation, a naive implementation of the adjoint-equation solver would require the full storage, at every point in time and space, of the solution to the state equation. For large time-dependent problems, storing the whole estimate of the forward and adjoint solutions can quickly exceed the available memory: The storage of the entire forward solution trajectory is avoided by using a checkpointing strategy to balance storage and computation cost.

The control problem related to parabolic variational inequality may have potential for solving problems related to Reynolds lubrication on thin films and Electro-wetting on dielectrics(EWOD). These problems are complicated because of the moving boundary.

All problems in this thesis have physical relevance and gives useful information. Control for three dimensional domain will be interesting problem for the control of parabolic variational inequality of the obstacle type. This remains for the future work to be found out.

A conclusion from this study is that the control problem for the parabolic variational inequalities using exterior penalty based technique are feasible, but they will be, as a rule of thumb, at least an order of magnitude more expensive to compute than pure simulations. To model the macroscopic phenomenon governed by variational inequality, we need time scaling and length scaling and experimental data with appropriate boundary condition. Using memory saving device described in Section 5.5.1, even control problem for variational inequalities in three space dimensions will be amenable to the techniques outlined above. However, because of the extreme computational demands for such problems, it will be crucial to develop methods exploiting parallelism.

Bibliography

[1] M.S. Alnaes and A. Logg. “Unified Framework for Finite Element Assembly.” In: *J. Computat. Sci. Eng.* 4 (2009), pp. 231–244.

[2] M.S. Alnaes et al. *UFL: A finite element form language.* Berlin: Springer-Verlag, 2011. Chap. 17, pp. 299–334.

[3] M. S. Alnæs et al. *FEniCS:Automated Solution of Differential Equation by Finite Element Method.* 2013. URL: `http://www.fenicsproject.org`.

[4] M. Andrle and A. El Badia. “On an inverse source problem for the heat equation. Application to a pollution detection problem, II”. In: *Inverse Problems in Science and Engineering* 23.3 (2015), pp. 389–412.

[5] A. El Badia and T. Ha-Duong. “An inverse source problem in potential analysis”. In: *Inverse Problems* 16.3 (2000), pp. 651–655.

[6] M. Berggren. “Optimal Control of Time Evolution Systems:Controllability Investigations and Numerical Algorithms.” PhD thesis. Rice University, 1995. Pp. 5–40.

[7] A. Bonito and R. Glowinski. “On the nodal set of the eigenfunctions of the Laplace-Beltrami Operator for bounded surfaces in $\mathbb{R}^3$.” In: *Communications on Pure and Applied Analysis.* 13.10 (2014), pp. 2–22.

[8] H. Brezis and W. A. Strauss. "Semi-linear second-order elliptic equations in L^1". In: *J. Math. Soc. Japan* 25.4 (1973), pp. 565–590.

[9] C. Carthel, J.L. Lions, and R. Glowinski. "On exact and approximate boundary controllability for the heat equation." In: *Optimization theory Appl* 82.10 (1994), pp. 42–484.

[10] J. Casti. "On the general inverse problem of optimal control theory". In: *Journal of Optimization Theory and Applications* 32.4 (1980), pp. 491–497.

[11] P.E. Farrel et al. "Automated derivation of the adjoint of high- level transient finite element programs." In: *SIAM Journal on Scientific Computing.* 322 (2013), pp. 891–921.

[12] M. B. Giles and N. A. Pierce. "An Introduction to the Adjoint Approach to Design". In: *Flow, Turbulence and Combustion* 65.3 (2000), pp. 393–415.

[13] R. Glowinski. *Exact and Approximate Controllability for Distributed Parameter Systems:A Numerical Approach.* Cambridge University Press, 2008, pp. 1–53.

[14] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems.* Springer-Verlag,New York, 1984, pp. 98–140.

[15] A. Griewank and A. Walther. "Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation." In: *ACM Transactions on Mathematical Software.* (March 2000), pp. 19–45.

[16] M. Hintermüller, R. HW Hoppe, and C. Löbhard. "Dual-weighted goal-oriented adaptive finite elements for optimal control of elliptic variational inequalities". In: *ESAIM: Control, Optimisation and Calculus of variations* 20.2 (2014), pp. 524–546.

[17] M. Hintermüller and I. Kopacka. “A smooth penalty approach and a nonlinear multigrid algorithm for elliptic MPECs”. In: *Computational Optimization and Applications* 50.1 (2011), pp. 111–145.

[18] M. Hintermüller and T. Surowiec. “A bundle-free implicit programming approach for a class of elliptic MPECs in function space”. In: *Mathematical Programming* 160.1 (2016), pp. 271–305.

[19] M. Hintermüller, K. Ito, and K. Kunisch. “The Primal-Dual Active Set Strategy as a Semismooth Newton Method”. In: *SIAM Journal on Optimization* 13.3 (2002), pp. 865–888.

[20] A. Jameson. “Aerodynamic design via control theory”. In: *Journal of scientific computing* 3.3 (1988), pp. 233–260.

[21] A. Jameson and E. Kreindler. “Inverse Problem of Linear Optimal Control”. In: *SIAM Journal on Control* 11.1 (1973), pp. 1–19.

[22] A. Jameson, L. Martinelli, and N.A. Pierce. “Optimum Aerodynamic Design Using the Navier–Stokes Equations”. In: *Theoretical and Computational Fluid Dynamics* 10.1 (1998), pp. 213–237.

[23] D. Kraft. “A software package for sequential quadratic programming”. In: *Tech. Rep. DFVLR-FB 88-28, DLR German Aerospace Center — Institute for Flight Mechanics* (1988).

[24] O. A. Ladyzenskaja and V. A. Solonnikov. “Linear and Quasilinear Equations of Parabolic Type.” In: *Communications on Pure and Applied Analysis.* 2.5 (1968), pp. 2115–2126.

[25] H.P. Langtangen. *Finite elements in 2D and 3D.* 2013. URL: `http://hplgit.github.io/INF5620/doc/pub/sphinx-fem`.

[26] G. Lebeau and L. Robbiano. “Controle exact de l’equation de la chaleur.” In: *Communication in partial differential equation.* 322.10 (1905), pp. 891–921.

[27] J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations.* Springer, New York, 1971, pp. 13–41.

[28] J.L. Lions. *Some Aspects of the Optimal Control of Distributed Parameter Systems.* 1972, pp. 8–48.

[29] J.L. Lions. *Équations Différentielles Opérationnelles et Problèmes aux Limites.* 1961.

[30] F. Mignot and J. P. Puel. “Optimal Control in Some Variational Inequalities”. In: *SIAM Journal on Control and Optimization* 22.3 (1984), pp. 466–476.

[31] M. E Monroe et al. “VIPER: an advanced software package to support high-throughput LC-MS peptide identification”. In: *Bioinformatics* 23.15 (2007), pp. 2021–2023.

[32] A. Münch and E. Zuazua. “Numerical approximation of null controls for the heat equation: ill-posedness and remedies”. In: *Inverse Problems* 26.8 (2010), p. 085018.

[33] J.C. Nédélec. “A new family of mixed finite elements in $\mathbb{R}^3$.” In: *Numer.Math.* 50 (1986), pp. 57–81.

[34] J.C. Nédélec. “Mixed finite elements in $\mathbb{R}^3$.” In: *Numer.Math.* 35 (1980), pp. 315–341.

[35] J. Nocedal and S.J. Wright. *Numerical Optimization.* Vol. 22. Springer, 2000, pp. 59–60.

[36] R.E. Perez, P.W. Jansen, and Joaquim R. R. A. Martins. “pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization”. In: *Structures and Multidisciplinary Optimization* 45.1 (2012), pp. 101–118.

[37] O. Pironneau. Optimal shape design for elliptic systems. In: *System Modeling and Optimization: Proceedings of the 10th IFIP Conference New York City, USA, August 31 – September 4, 1981*. Ed. by R. F. Drenick and F. Kozin. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982, pp. 42–66.

[38] A. Quarteroni et al. Numerical Approximation of a Control Problem for Advection-Diffusion Processes. In: *System Modeling and Optimization: Proceedings of the 22nd IFIP TC7 Conference held from July 18–22, 2005, in Turin, Italy*. Ed. by F. Ceragioli et al. Boston, MA: Springer US, 2006, pp. 261–273.

[39] M.E. Rognes et al. "Automating the solution of PDEs on the sphere and other manifolds in FEniCS 1.2." In: *Optimization theory Appl* (2013), pp. 1–65.

[40] D.C. Sorensen and R. Glowinski. "Computing the Eigenvalues of the Laplace-Beltrami Operator on the surface of a Torus:A Numerical Approach." In: *Computational Methods of Applied Sciences,Springer,Dordrecht.* 16.10 (2008), pp. 225–232.

[41] F. Tröltzsch. *Optimal Control of Partial Differential Equations*. Vol. 112. American Mathematical Society, 2010, pp. 119–179.