MASTER'S THESIS

Preconditioners for linear parabolic optimal control problems

Tsang, Siu Chung

Date of Award:
2017

HONG KONG BAPTIST UNIVERSITY

Master of Philosophy

THESIS ACCEPTANCE

DATE: October 11, 2017

STUDENT'S NAME: TSANG Siu Chung

THESIS TITLE: Preconditioners for Linear Parabolic Optimal Control Problems

This is to certify that the above student's thesis has been examined by the following panel members and has received full approval for acceptance in partial fulfillment of the requirements for the degree of Master of Philosophy.

Chairman:	Dr. Chu Xiaowen Associate Professor, Department of Computer Science, HKBU (Designated by Dean of Faculty of Science)
Internal Members:	Dr. Liu Hongyu Associate Professor, Department of Mathematics, HKBU (Designated by Head of Department of Mathematics)
	Dr. Kwok Wing Hong Felix Assistant Professor, Department of Mathematics, HKBU
External Members:	Prof. Chung Eric Tsz Shun Associate Professor Department of Mathematics The Chinese University of Hong Kong

Issued by Graduate School, HKBU

Preconditioners for Linear Parabolic Optimal Control Problems

TSANG Siu Chung

A thesis submitted in partial fulfillment of the requirements

for the degree of

Master of Philosophy

Principal Supervisor:

Dr. Kwok Wing Hong, Felix (Hong Kong Baptist University)

October 2017

DECLARATION

I hereby declare that this thesis represent my own work which has been done after registration for the degree of MPhil at Hong Kong Baptist University, and has not been previously included in a thesis, dissertation submitted to this or other institution for a degree, diploma or other qualifications.

I have read the University's current research ethics guidelines, and accept responsibility for the conduct of procedures in accordance with the University's Committee on the Use of Human & Animal Subjects in Teaching and Research (HASC). I have attempted to identify all risks related to this research that may arise in conducting this research, obtained the relevant ethical and/or safety approval (where applicable), and acknowledged my obligations and the rights of the participants.

Signature: 

Date: October 2017

ABSTRACT

In this thesis, we consider the computational methods for linear parabolic optimal control problems. We wish to minimize the cost functional while fulfilling the parabolic partial differential equations (PDE) constraint. This type of problems arises in many fields of science and engineering. Since solving such parabolic PDE optimal control problems often lead to a demanding computational cost and time, an effective algorithm is desired. In this research, we focus on the distributed control problems. Three types of cost functional are considered: Target States problems, Tracking problems, and All-time problems. Our major contribution in this research is that we developed a preconditioner for each kind of problems, so our iterative method is accelerated.

In chapter 1, we gave a brief introduction to our problems with a literature review. In chapter 2, we demonstrated how to derive the first-order optimality conditions from the parabolic optimal control problems. Afterwards, we showed how to use the shooting method along with the *flexible generalized minimal residual* to find the solution. In chapter 3, we offered three preconditioners to enhance our shooting method for the problems with symmetric differential operator. Next, in chapter 4, we proposed another three preconditioners to speed up our scheme for the problems with non-symmetric differential operator. Lastly, we have the conclusion and the future development in chapter 5.

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my supervisor Dr. Felix Kwok. I am very fortunate to have a chance to work with such a supportive and knowledgeable supervisor. I am very grateful to him for bringing me into the rich field of scientific computing research. He has been very reliable and helpful since I became his graduate student. The suggestions he has made for my work have been very valuable.

I am also grateful to Dr. Henry Ngan. He has been encouraging me, both academically and personally, since I am an undergraduate student. He helped me to finish the first research project and the first publication in my life. My acknowledgment also goes to Prof. Michael Ng, who offered me some great advice in the academic life, as well as introducing me to Dr. Kwok. I also have to thank my colleague Mr. Siu Ka Wai. He is always there to discuss the research difficulties with me in the past two years.

I would like to thank Prof. Julien Salomon along with his graduate student Sebastian Reyes Riffo. I am thankful to Prof. Salomon for inviting me to visit Université Paris-Dauphine. The meeting and discussion with this research team in Paris have been inspiring.

I acknowledge the finical support from the University Grant Studentships offered by the Hong Kong Baptist University (HKBU). I am also very grateful for all the facilities, research activities, and seminars organized by the Mathematics Department of HKBU.

I am extremely thankful to my love, Cherry Lui, for being so amazing and supportive over the years. She has been a forever source of enjoyment, strength, and encouragement. Last but not least, I have to thank my family.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we will introduce the optimal control problems governed by linear parabolic partial differential equations. A background is given in section 1.1. We will show a motivational example in section 1.2. A literature review is presented in section 1.3. Then, we will address three types of linear parabolic optimal control problems in section 1.4: target states problems, tracking problems, and all-time problems.

1.1 Background

Partial differential equations (PDE) is perhaps one of the most fundamental types of problems in applied mathematics. PDE are widely used to describe many phenomenon in the real world. PDE related models appear in almost every area in our modern society such as physics, chemistry, biology, economics, engineering, medicine, and the list goes on. The solution of PDE is crucial in many kinds of practical problems. The numerical and analytical studies of PDE has been an active research topic over many decades. In the scientific computing community, the main focus has been developing numerical methods for solving the PDE. One major branch of the studies is to establish different strategies for constructing a discrete representation of the PDE. Finite differences method, finite elements method, and finite volume method are some of the established discretization schemes in this field. These methods will generate a matrix system that is equivalent to the corresponding PDE. Naturally, another major group of studies is interested in the numerical methods for solving

such matrix system. These numerical methods are classified into direct methods or iterative methods. Since the matrix system arose from the PDE is often very large and sparse, direct methods are unlikely to be feasible in many applications. With the help of the modern computer development in the past few decades, we are now capable of solving such large and sparse matrix system effectively by iterative solver. The performance of an iterative method is highly depended on the structure of the matrix system. Hence, preconditioning techniques are very useful to enhance the efficiency of an iterative method. Therefore, in this thesis, we are interested in developing preconditioners to accelerate the iterative method that solve the matrix system arose in our PDE optimal control problems.

Optimization is another type of problem that is broadly implemented in the scientific community. To solve optimization problem is to find a solution such that the objective function is maximized (or minimized) while fulfilling some constraint equations. The history of constrained optimization problems is more than three centuries, where Lagrange multiplier was proposed as a fundamental idea to solve such problems. In this research, we are interested in the PDE-constrained optimization problems, which mean the constraint equation of the optimization problem is a PDE. This class of problems is also known as PDE optimal control problems. A detailed historical review of this type of problems can be found in [15]. We will also present a literature review of PDE optimal control problems in section 1.3. Throughout this thesis, we will be focusing on parabolic PDE as our constraint equation particularly. Mathematically speaking, our concerned problem read as the following general form:

$$\begin{aligned} \min_{y,u} \quad & J(y,u) \\ \text{s.t.} \quad & \mathcal{D}y = u \quad \text{in} \quad [0,T]\times\Omega \\ & y = g \quad \text{in} \quad [0,T]\times\partial\Omega \end{aligned} \tag{1.1}$$

where J is the cost function, Ω is the space domain, $[0,T]$ is the time domain, $\mathcal{D}$ is some parabolic differential operator, y is the unknown variable of the PDE, u is known as the source term of the PDE, and g is the function corresponds to the boundary condition.

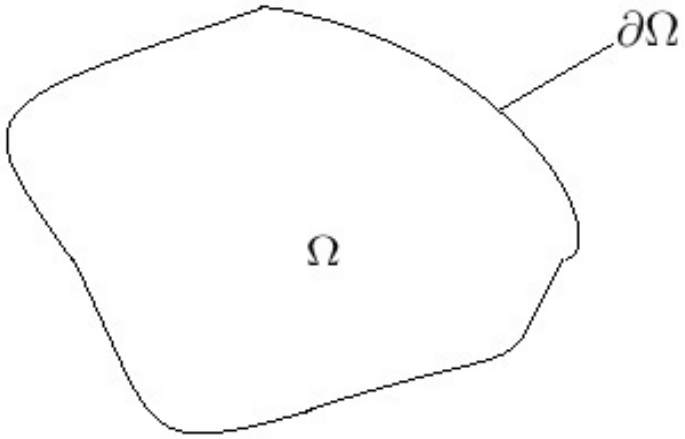


Figure 1.1: A visualization of domain Ω.

1.2 Motivational Example: Optimal Control of Distributed Heating

In this section, we aim to provide a real-life example of parabolic PDE optimal control problem as a motivation. If readers are interested, more examples could be found in [12] and [48]. Let say a company wanted to control the temperature distribution in a certain environment over a period of time, as they know that this temperature distribution can maintain their products in the best condition (For instances, chemical laboratory, food store, or nuclear reactor). We suppose that the company have a heater (or cooler) that is distributed all over the environment. Then, the company can achieve the desired temperature distribution by solving a PDE optimal control problem.

Assume that we have a spatial domain $\Omega \subseteq \mathbb{R}^i$ with $i = 1, 2, 3$ (See figure 1.1). We are interested in controlling the temperature distribution $y : [0, T] \times \Omega \longrightarrow [0, T] \times \mathbb{R}_0^+$ over a so-called time-space cylinder $[0, T] \times \Omega$, where $T > 0$ is the terminal time. We also assume that the temperature on the boundary $\partial\Omega$ is known, i.e., we have a Dirichlet boundary condition. Consider a simple case that we can control a heat source $u : [0, T] \times \Omega \longrightarrow [0, T] \times \mathbb{R}_0^+$ without any constraint (for example, microwave heating without any physical and technical limit). We wish to find u such that the actual temperature distribution y is "as close as possible" to our desired temperature

distribution $\hat{y} : [0,T] \times \Omega \longrightarrow [0,T] \times \mathbb{R}_0^+$. Note that the temperature y is governed by the heat equation. Therefore, we have our optimal control problem of distributed control heating

$$\begin{aligned} \min_{y,u} \quad & J(y,u) \\ \text{s.t.} \quad & \tfrac{\partial}{\partial t} y - c\Delta y = u \quad \text{in} \quad [0,T] \times \Omega \\ & y = g \quad \text{in} \quad [0,T] \times \partial\Omega \end{aligned} \tag{1.2}$$

where J is the cost function to be minimize, $c \in \mathbb{R}^+$ is the thermal conductivity, and g is function corresponds to the boundary condition. To achieve our goal "y as close as possible to $\hat{y}$", a common formulation of the cost functional J is

$$J(y,u) = \frac{1}{2}\int_0^T \int_\Omega u(x,t)^2 \mathrm{d}x\mathrm{d}t + \frac{\alpha_2}{2}\int_\Omega \left(y(x,T) - \hat{y}(x,T)\right)^2 \mathrm{d}x + \frac{\alpha_3}{2}\int_\Omega \left(y(x,0) - \hat{y}(x,0)\right)^2 \mathrm{d}x \tag{1.3}$$

for some non-negative real constant α_2 and α_3. Here, the first integral can be interpreted as a cost of the heating (or cooling) device that the user want to minimize, while the second and third integral means that the user want to drive y close to $\hat{y}$ at the initial time $t = 0$ and terminal time $t = T$. The constant α_i can be seen as the proportion that the user values the control terms. For example, if a user is more submitted into the final time control than the initial time and the heating device cost, then the user can set α_2 as a large value.

1.3 Literature Review

As mentioned briefly in section 1.1, our interested PDE optimal control problems is an active research area in applied mathematics. Glowinski and Lions [16] described this class of problems as "*at a given time horizon we want the system under study to behave exactly as we wish (or in a manner arbitrarily close to it).*" Over the past decade, the understanding of both theoretical and numerical aspects of this problem had a significant expansion [48][12]. The broad applicability of PDE-constrained optimization is one of the reasons why this problem receives so much interest in the research community. Such applications include shape optimization for engineering [46][47][31][22], optimal control in quantum physics [34][7][5][9][10][30][51], PDE-optimization of electromagnetic inverse scattering [20][21][38], optimal control

of reaction diffusion equation [1][19][4], medical imaging system [27], the control of bacterial chemotaxis system [36], the control of Simulated Moving Bed process [35], optimal control of surface acoustic wave [13], the control of fluid flow [11][49], and so on.

Solving the PDE optimal control problems often results in a very large and computationally expensive system. Numerous numerical methods have been developed to solve the PDE-constrained optimization problem numerically. Preconditioned all-at-once saddle point system and multigrid methods are two common approaches to reduce the computational cost. In these approaches, the optimal control problems are discretized into a so-called all-at-once linear system. It can be proved that this very large and sparse system is in a saddle point form. It is well-known that a block preconditioner could accelerate the iterative method for saddle point system. A famous paper from Benzi, Golub, and Liesen [2] gave a great review of saddle point systems and the corresponding numerical methods. In many applications, the multigrid method will be introduced to further reduce the computational cost. There are many researchers working on the development of this kind of methods. For instances, Sarkis proposed a block diagonal parareal preconditioner for parabolic PDE constraint [44], Rees offered an all-at-once preconditioner for elliptic PDE constraint [37], Pearson gave a block triangular preconditioner for time-dependent Stokes PDE constraints [32], and Schiela presented an operator preconditioner for inequality constrained optimal control problems [45]. It is worth to mention that it is extremely inefficient to form the saddle point matrix explicitly. A nature way to deal with this issue is to obtain the matrix-vector multiplication in each iteration step by solving the forward-backward coupled PDE. We will show this so-called shooting implementation in section 2.2 in this thesis.

One may also apply multigrid scheme directly to the PDE optimal control problems. Borzi introduced the Collective Smoothing Multigrid method for parabolic PDE-optimization problems in [6]. A comprehensive review of this family of methods could be found in the book [8].

Another idea is to employ domain decomposition method so that the large system can be solved in a parallel manner. For example, Gander and Kwok proposed a Schwarz methods in time for the control of parabolic PDE [14][28]. Heinkenschloss designed an iterative method with a Gauss–Seidel preconditioner for linear quadratic optimal control problems [23], and Lagnese proposed a time-domain decomposition scheme for wave equation constraint [29].

Despite all of the above-mentioned methods, the computing storage and time for the PDE optimal control problems could still be extremely demanding. The PDE optimal control problems involve a huge volume of data such that even advanced computer is not capable of calculating the solution efficiently. Therefore, in this research, the objective is to design a fast and robust algorithm to solve these problems in a feasible amount of computational cost. We wish to investigate the parabolic PDE optimal control problems via shooting method and preconditioning. Our preconditioners can also be used in the domain decomposition scheme, as the domain decomposition scheme would generate many small problems that fit with our setting [28]. Our preconditioners could be applied on these small problems and hence accelerate the entire domain decomposition scheme.

1.4 Linear Parabolic Optimal Control Problems

In this section, we are going to formulate and define our interested linear parabolic optimal control problems. Consider a parabolic partial differential equation as our constraint. By semi-discretization in space, we can represent the constraint as a system of ordinary differential equations:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) = \mathbf{u}(t) + \mathbf{f}(t), \quad t \in [0, T] \tag{1.4}$$

where A is a matrix arising from the discretization of partial differential operator in space, $\mathbf{y}(t) : [0, T] \longrightarrow \mathbb{R}^m$ is the state variable, $\mathbf{u}(t) : [0, T] \longrightarrow \mathbb{R}^m$ is the control variable, and $\mathbf{f}(t) : [0, T] \longrightarrow \mathbb{R}^m$ is some known source term. Note that m is the degree of freedom in the spatial discretization. It is worth to mention here that our

matrix A is a sparse and large matrix as it is obtained by finite difference method, finite element method, or finite volume method. Furthermore, since we are considering parabolic PDE throughout this thesis, we assume that our matrix A has positive semi-definite Hermitian part.

We assume that there is no limitation or constraint for control variable $\mathbf{u}(t)$. Our goal is to vary the control variable $\mathbf{u}(t)$ such that the cost functional $J(\mathbf{y}, \mathbf{u})$ is minimized. Our optimal control problems read as

$$\min_{\mathbf{y},\mathbf{u}} J(\mathbf{y}, \mathbf{u}) \quad \text{subject to} \quad \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) = \mathbf{u}(t) + \mathbf{f}(t), \quad t \in [0, T]$$

The cost functional $J(\mathbf{y}, \mathbf{u})$ vary with different applications. We will define our interested cases in next section. But before that, we first introduce the norm that we need to use in the cost functional. For any vector $\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \in \mathbb{R}^m$, we define the standard L2-norm

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + ... + x_m^2} = \sqrt{\mathbf{x}^T\mathbf{x}}$$

1.4.1 Target States Problems

The target states problems arise as a sub-problem in a new class of domain decomposition method for parabolic optimal control problems. This domain decomposition scheme is described in [28]. Suppose the desired optimal state at any time t is denoted by $\hat{\mathbf{y}}(t) : [0, T] \longrightarrow \mathbb{R}^m$. In many situations, the user is only interested in controlling the initial and terminal state of the state variable $\mathbf{y}(t)$, i.e. our goal is to make $\mathbf{y}(0)$ and $\mathbf{y}(T)$ become "as close as possible" to $\hat{\mathbf{y}}(0)$ and $\hat{\mathbf{y}}(T)$. A common formulation of such cost functional $J(\mathbf{y}, \mathbf{u})$ takes the form

$$J(\mathbf{y}, \mathbf{u}) = J_1(\mathbf{y}, \mathbf{u}) = \frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t + \frac{\alpha_2}{2}\|\mathbf{y}(T) - \hat{\mathbf{y}}(T)\|_2^2 + \frac{\alpha_3}{2}\|\mathbf{y}(0) - \hat{\mathbf{y}}(0)\|_2^2 \quad (1.5)$$

for some positive real constant α_2 and α_3.

Note that in this $J_1(\mathbf{y}, \mathbf{u})$ formulation, the intermediate state (i.e. when $t \in (0, T)$) of $\mathbf{y}(t)$ is ignored. This means that we do not "track" the intermediate time-steps

of $\mathbf{y}(t)$, and only interested in the initial and terminal state. In many applications, the initial (or terminal) state of $\mathbf{y}(t)$ may already be known. The user may only interested in controlling the terminal (or initial) state instead of controlling both. In these cases, we can simply put $\alpha_i = 0$ for some i to make the unnecessary components of $J_1(\mathbf{y}, \mathbf{u})$ vanish, and put the known initial (or terminal) condition of $\mathbf{y}(t)$ into the constraints. However, without loss of generality, we will consider both α_2 and α_3 is positive in this thesis.

1.4.2 Tracking Problems

The above cost functional $J_1(\mathbf{y}, \mathbf{u})$ is not the only formulation. The user may wish to control the entire time-interval of $\mathbf{y}(t)$, i.e. tracking all time-steps. In this case, we would like to drive $\mathbf{y}(t)$ "as close as possible" to $\hat{\mathbf{y}}(t)$. A typical formulation of this kind of cost functional read as

$$J(\mathbf{y}, \mathbf{u}) = J_2(\mathbf{y}, \mathbf{u}) = \frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|_2^2 \mathrm{d}t \tag{1.6}$$

where α_1 is a positive real constant. And we need the initial condition in the constraint too:

$$\mathbf{y}(0) = \mathbf{y}_0 \quad \text{in} \quad \Omega$$

We can see that it is more challenging problem compare to the target state problems in a computational point of view, as tracking the entire time domain means a tremendous amount of computational cost.

1.4.3 All-time Problems

Lastly, we consider the so-called all-time problems. This type of problems also arise as a sub-problem of the domain decomposition method mentioned in [28]. The corresponding cost functional of all-time problems read as

$$\begin{aligned} J(\mathbf{y}, \mathbf{u}) = J_3(\mathbf{y}, \mathbf{u}) = \frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|_2^2 \mathrm{d}t \\ + \frac{\alpha_2}{2}\|\mathbf{y}(T) - \hat{\mathbf{y}}(T)\|_2^2 + \frac{\alpha_3}{2}\|\mathbf{y}(0) - \hat{\mathbf{y}}(0)\|_2^2 \end{aligned} \tag{1.7}$$

In the rest of this thesis, we will see that these three cases can be translated into some systems of linear equations with different structure, and we will propose different

strategies to solve such systems. In next chapter, we will present the derivation of these systems.

Chapter 2

Methodology

In the last chapter, we studied the formulation of our interested parabolic optimal control problems. The next step is to derive the optimality system from it, which is presented in section 2.1. Section 2.2 will show how to use shooting method to solve such system effectively. Afterwards, an introduction on preconditioned iterative methods for the system of linear equations is given in section 2.3.

2.1 Optimality System

Here, we will use the standard Lagrange multiplier method to find the first order necessary conditions, and hence derive the optimality system for our interested optimization problems. Note that in our setting, we obviously have some convex optimization problems since we have the quadratic cost functional with a linear constraint. Hence, with the convexity assumption given, it is well-known that our first order necessary conditions are also the sufficient conditions. There are two approaches available to derive our desired optimality system, namely *discretize-then-optimize* and *optimize-then-discretize*. In the following, we will show that if a proper discretization scheme is selected, these two approaches will give an equivalent result but with a mismatch in the final time-step. It can be proved that this mismatch is proportional to time-step size τ. However, it is possible for these two approaches to have a completely different optimality system if some other discretization schemes are employed. In the rest of this thesis, we will only consider the optimality system obtained from

the *discretize-then-optimize* approach. Nevertheless, it is worth to mention that our proposed preconditioners in this thesis are also suitable for the system obtained from *optimize-then-discretize* approach or with some other discretization schemes. We will illustrate the idea of these two approaches with the all-time problems. We will not go through the derivation one by one for each problem as the processes are rather similar. We recommend [48] for a detailed Lagrangian derivation for optimality conditions in a general setting.

2.1.1 Discretize then Optimize

All-time Problems

We consider the cost functional with all-time control J_3 (1.7) subject to a system of ODE obtained from the discretization of parabolic PDE (1.4) as our illustration example. The same methodology can be applied to obtain the corresponding optimality system for other problems. Here, we use a uniform time-grid on $[0, T]$ with step-size τ. For $k = 0, ..., n$, we take $\mathbf{y}^{(k)}$, $\mathbf{u}^{(k)}$, $\boldsymbol{\lambda}^{(k)}$, $\hat{\mathbf{y}}^{(k)}$, and $\mathbf{f}^{(k)}$ be the k-th time-step of $\mathbf{y}(t)$, $\mathbf{u}(t)$, $\boldsymbol{\lambda}(t)$, $\hat{\mathbf{y}}(t)$, and $\mathbf{f}(t)$ respectively. Also, we define $\tilde{\mathbf{y}} = \left(\left(\mathbf{y}^{(1)}\right)^T, \ldots, \left(\mathbf{y}^{(n)}\right)^T\right)^T$, $\tilde{\mathbf{y}}' = \left(\left(\hat{\boldsymbol{y}}^{(1)}\right)^T, \ldots, \left(\hat{\boldsymbol{y}}^{(n)}\right)^T\right)^T$, $\tilde{\mathbf{u}} = \left(\left(\mathbf{u}^{(1)}\right)^T, \ldots, \left(\mathbf{u}^{(n)}\right)^T\right)^T$, and $\tilde{\boldsymbol{\lambda}} = \left(\left(\boldsymbol{\lambda}^{(1)}\right)^T, \ldots, \left(\boldsymbol{\lambda}^{(n)}\right)^T\right)^T$. Note that the adjoint variable $\boldsymbol{\lambda}(t) : [0, T] \to \mathbb{R}^m$ is also known as the Lagrange multiplier. Our discretization approach is to employ implicit Euler's method for the differentiation with respect to time for our ODE constraint (1.4). It is perhaps the most common and easiest way to discretize such system. For readers who are unfamiliar with this method, a detailed review can be found in [26].

By implicit Euler's method, the constraint (1.4) can be discretize as

$$\frac{\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}}{\tau} + A\mathbf{y}^{(k+1)} = \mathbf{u}^{(k+1)} + \mathbf{f}^{(k+1)}$$

which yield

$$(\tau A + I)\,\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)} - \tau\mathbf{u}^{(k+1)} = \tau\mathbf{f}^{(k+1)}$$

Hence, we can formulate

$$K\tilde{\mathbf{y}} - \tau\tilde{\mathbf{u}} = \tilde{\mathbf{f}}_0 \tag{2.1}$$

where

$$K = \begin{pmatrix} (\tau A + I) & & & & & \\ -I & (\tau A + I) & & & & \\ & \ddots & \ddots & & & \\ & & & -I & (\tau A + I) & \\ & & & & -I & (\tau A + I) \end{pmatrix}, \quad \text{and} \quad \tilde{\mathbf{f}}_0 = \begin{pmatrix} \mathbf{y}^{(0)} + \tau \mathbf{f}^{(1)} \\ \mathbf{f}^{(2)} \\ \vdots \\ \mathbf{f}^{(n)} \end{pmatrix} \tag{2.2}$$

Afterwards, we would like to discretize the cost functional J_3 (1.7). Firstly, we have

$$\frac{\alpha_2}{2}\|\mathbf{y}(T) - \hat{\mathbf{y}}(T)\|_2^2 + \frac{\alpha_3}{2}\|\mathbf{y}(0) - \hat{\mathbf{y}}(0)\|_2^2 = \frac{\alpha_2}{2}\|\mathbf{y}^{(n)} - \hat{\mathbf{y}}^{(n)}\|_2^2 + \frac{\alpha_3}{2}\|\mathbf{y}^{(0)} - \hat{\mathbf{y}}^{(0)}\|_2^2 \tag{2.3}$$

Secondly, by right-rectangular rule, the first term of J_3 (1.7) can be discretized as

$$\frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t = \frac{\tau}{2}\sum_{k=1}^{n} \|\mathbf{u}^{(k)}\|_2^2 = \frac{\tau}{2}\|\tilde{\mathbf{u}}\|_2^2 \tag{2.4}$$

Lastly, we use right-rectangular rule again on the second term of J_3 (1.7)

$$\frac{\alpha_1}{2}\int_0^T \|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|_2^2 \mathrm{d}t = \frac{\alpha_1 \tau}{2}\sum_{k=1}^{n} \|\mathbf{y}^{(k)} - \hat{\mathbf{y}}^{(k)}\|_2^2 = \frac{\alpha_1 \tau}{2}\|\tilde{\mathbf{y}} - \tilde{\mathbf{y}}'\|_2^2 \tag{2.5}$$

Combining (2.1), (2.3), (2.4), and (2.5), we have a discretized version of all-time problems

$$\min_{\mathbf{y},\mathbf{u}} J_3(\mathbf{y}, \mathbf{u}) = \frac{\tau}{2}\|\tilde{\mathbf{u}}\|_2^2 + \frac{\alpha_1 \tau}{2}\|\tilde{\mathbf{y}} - \tilde{\mathbf{y}}'\|_2^2 + \frac{\alpha_2}{2}\|\mathbf{y}^{(n)} - \hat{\mathbf{y}}^{(n)}\|_2^2 + \frac{\alpha_3}{2}\|\mathbf{y}^{(0)} - \hat{\mathbf{y}}^{(0)}\|_2^2$$

Subject to a constraint

$$K\tilde{\mathbf{y}} - \tau\tilde{\mathbf{u}} = \tilde{\mathbf{f}}_0$$

Consider the Lagrangian function

$$\mathcal{L} = J_3 + \tilde{\boldsymbol{\lambda}}^T \left(K\tilde{\mathbf{y}} + \tau\tilde{\mathbf{u}} - \tilde{\mathbf{f}}_0 \right)$$

It is well known that if we have our optimal solutions of state variable $\mathbf{y}(t)$, adjoint variable $\boldsymbol{\lambda}(t)$, and control variable $\mathbf{u}(t)$, the following must hold

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{y}^{(i)}} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{u}^{(i)}} = \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\lambda}^{(i)}} = 0, \quad \text{for,} \quad i = 0, ..., n \tag{2.6}$$

This first-order necessary conditions (2.6) is also known as the *Karush-Kuhn-Tucker (KKT) conditions* of optimization problems. Therefore, with $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{y}^{(i)}} = 0$, we have

$$\begin{cases} \alpha_3 \left(\mathbf{y}^{(0)} - \hat{\mathbf{y}}^{(0)}\right) - \boldsymbol{\lambda}^{(0)} = 0 \\ K^T\tilde{\boldsymbol{\lambda}} + \alpha_1\tau\left(\tilde{\mathbf{y}} - \tilde{\mathbf{y}}'\right) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \alpha_2\left(\mathbf{y}^{(n)} - \hat{\mathbf{y}}^{(n)}\right) \end{pmatrix} = 0 \implies K^T\tilde{\boldsymbol{\lambda}} + B\tilde{\mathbf{y}} = \tilde{\mathbf{g}} \end{cases} \tag{2.7}$$

where

$$B = \begin{pmatrix} \alpha_1\tau I & & & \\ & \ddots & & \\ & & \alpha_1\tau I & \\ & & & (\alpha_1\tau + \alpha_2)I \end{pmatrix}, \text{and} \quad \tilde{\mathbf{g}} = \begin{pmatrix} \alpha_1\tau\hat{\mathbf{y}}^{(1)} \\ \vdots \\ \alpha_1\tau\hat{\mathbf{y}}^{(n-1)} \\ (\alpha_1\tau + \alpha_2)\hat{\mathbf{y}}^{(n)} \end{pmatrix}$$

Next, if we differentiate the Lagrangian function with respect to $\mathbf{u}^{(i)}$, we can have

$$\tau\tilde{\mathbf{u}} - \tau\tilde{\boldsymbol{\lambda}} = 0 \implies \tilde{\mathbf{u}} = \tilde{\boldsymbol{\lambda}} \tag{2.8}$$

Last but not least, from $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\boldsymbol{\lambda}^{(i)}} = 0$, we recover our constraint

$$K\tilde{\mathbf{y}} - \tau\tilde{\mathbf{u}} = \tilde{\mathbf{f}}_0 \tag{2.9}$$

Combining (2.8), (2.9), and the initial condition found in (2.7), we obtain

$$K^T\tilde{\mathbf{y}} - C\tilde{\boldsymbol{\lambda}} = \tilde{\mathbf{f}}$$

where,

$$C = \begin{pmatrix} (\frac{1}{\alpha_3} + \tau)I & & & \\ & \tau I & & \\ & & \ddots & \\ & & & \tau I \end{pmatrix}, \quad \text{and} \quad \tilde{\mathbf{f}} = \begin{pmatrix} \hat{\mathbf{y}}^{(0)} + \tau\mathbf{f}^{(1)} \\ \tau\mathbf{f}^{(2)} \\ \vdots \\ \tau\mathbf{f}^{(n)} \end{pmatrix}$$

Hence, we have an all-at-once system as our first order necessary conditions for the tracking problems.

$$\begin{pmatrix} K^T & B \\ -C & K \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\lambda}} \\ \tilde{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{g}} \\ \tilde{\mathbf{f}} \end{pmatrix} \tag{2.10}$$

We define the the first row of (2.10) as the *backward adjoint equation*

$$K^T\tilde{\boldsymbol{\lambda}} + B\tilde{\mathbf{y}} = \tilde{\mathbf{g}} \tag{2.11}$$

and the *forward state equation* read as

$$K\tilde{\mathbf{y}} - C\tilde{\boldsymbol{\lambda}} = \tilde{\mathbf{f}} \tag{2.12}$$

Note that (2.12) and (2.11) correspond to a coupled forward-backward ODE in the *optimize-then-discretize* approach. Before we show that, we will first present the optimality system for the target state problems and tracking problems.

Target States Problems

In the previous subsection, we demonstrated how to derive the discretized optimality system of the all-time problem via *discretize-then-optimize* approach. The methodology for the target states case is the same as the all-time case. We do not show the entire derivation here for the sake of simplicity. For target states problems (i.e. cost functional J_1 (1.5) subject to the system of ODE (1.4)), we can use the same approach to obtain the corresponding optimality system

$$\begin{pmatrix} K^T & B_0 \\ -C & K \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\lambda}} \\ \tilde{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{g}}_0 \\ \tilde{\mathbf{f}} \end{pmatrix} \tag{2.13}$$

where K, C, $\tilde{\mathbf{y}}$, $\tilde{\boldsymbol{\lambda}}$, and $\tilde{\mathbf{f}}$ are defined in (2.10), and

$$B_0 = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & \alpha_2 I \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{g}}_0 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \alpha_2 \hat{\mathbf{y}}^{(n)} \end{pmatrix}$$

Similar, we define the *backward adjoint equation* as

$$K^T\tilde{\boldsymbol{\lambda}} + B_0\tilde{\mathbf{y}} = \tilde{\mathbf{g}}_0 \tag{2.14}$$

and the *forward state equation* as

$$K\tilde{\mathbf{y}} - C\tilde{\boldsymbol{\lambda}} = \tilde{\mathbf{f}} \tag{2.15}$$

Tracking Problems

Again, we will only show the resultant optimality system for the tracking problems. We have

$$\begin{pmatrix} K^T & B_1 \\ -C_1 & K \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\lambda}} \\ \tilde{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{g}}_1 \\ \tilde{\mathbf{f}}_1 \end{pmatrix} \tag{2.16}$$

where K, $\tilde{\mathbf{y}}$, and $\tilde{\boldsymbol{\lambda}}$ are defined in (2.10), with

$$B_1 = \begin{pmatrix} \alpha_1\tau I & & \\ & \ddots & \\ & & \alpha_1\tau I \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{g}}_1 = \begin{pmatrix} \alpha_1\tau\hat{\mathbf{y}}^{(1)} \\ \vdots \\ \alpha_1\tau\hat{\mathbf{y}}^{(n)} \end{pmatrix}$$

and

$$C_1 = \begin{pmatrix} \tau I & & \\ & \ddots & \\ & & \tau I \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{f}}_1 = \begin{pmatrix} \mathbf{y}_0 + \tau\mathbf{f}^{(1)} \\ \tau\mathbf{f}^{(2)} \\ \vdots \\ \tau\mathbf{f}^{(n)} \end{pmatrix}$$

Similar, we define the *backward adjoint equation* as

$$K^T\tilde{\boldsymbol{\lambda}} + B_1\tilde{\mathbf{y}} = \tilde{\mathbf{g}}_1 \tag{2.17}$$

and the *forward state equation* as

$$K\tilde{\mathbf{y}} - C_1\tilde{\boldsymbol{\lambda}} = \tilde{\mathbf{f}}_1 \tag{2.18}$$

2.1.2 Optimize then Discretize

In this section, we will derive the optimality system with *optimize-then-discretize* approach. Moreover, we are going to show that this approach match with the *discretize-then-optimize* approach with an error of $O(\tau)$ in the final condition. Again, we consider the all-time problems, i.e., the cost functional with tracking term J_3 (1.7) subject to the system of ODE (1.4), as our example. Introducing the Lagrangian function,

$$\mathcal{L}(\mathbf{y}, \mathbf{u}, \boldsymbol{\lambda}) = J_3(\mathbf{y}, \mathbf{u}) + \int_0^T \boldsymbol{\lambda}(t)^T \left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) - \mathbf{u}(t) - \mathbf{f}(t) \right) \mathrm{d}t$$

where $\boldsymbol{\lambda}(t) : [0,T] \longrightarrow \mathbb{R}^m$ is the adjoint variable. Firstly, we take the Fréchet derivative with respect to $\mathbf{y}(t)$ in some direction $\mathbf{z}_1(t)$, then we obtained

$$\begin{aligned} D_y\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = \alpha_1 \int_0^T \mathbf{z}_1(t)^T \left[\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right] \mathrm{d}t + \alpha_2 \mathbf{z}_1(t)^T \left[\mathbf{y}(T) - \hat{\mathbf{y}}(T)\right] \\ + \alpha_3 \mathbf{z}_1(0)^T \left[\mathbf{y}(0) - \hat{\mathbf{y}}(0)\right] + \int_0^T \boldsymbol{\lambda}(t)^T \left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{z}_1(t) + A\mathbf{z}_1(t)\right) \mathrm{d}t \end{aligned} \tag{2.19}$$

Note that with integration by parts and some simple algebra, we have

$$\begin{cases} \int_0^T \boldsymbol{\lambda}(t)^T \frac{\mathrm{d}\mathbf{z}_1(t)}{\mathrm{d}t} \mathrm{d}t = \mathbf{z}_1(T)^T \boldsymbol{\lambda}(T) - \mathbf{z}_1(0)^T \boldsymbol{\lambda}(0) - \int_0^T \mathbf{z}_1(t)^T \frac{\mathrm{d}\boldsymbol{\lambda}(t)}{\mathrm{d}t} \mathrm{d}t \\ \boldsymbol{\lambda}(t)^T A \mathbf{z}_1(t) = \mathbf{z}_1(t)^T A^T \boldsymbol{\lambda}(t) \end{cases}$$

Thus, (2.19) can be rewritten as

$$\begin{aligned} D_y\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = \int_0^T \mathbf{z}_1(t)^T \left(-\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\lambda}(t) + A^T\boldsymbol{\lambda}(t) + \alpha_1 \left(\mathbf{y}(t) - \alpha_1 \hat{\mathbf{y}}(t)\right)\right) \mathrm{d}t \\ + \mathbf{z}_1(T)^T \left(\alpha_2 \mathbf{y}(T) - \alpha_2 \hat{\mathbf{y}}(T) + \boldsymbol{\lambda}(t)\right) \\ + \mathbf{z}_1(0)^T \left(\alpha_3 \mathbf{y}(0) - \alpha_3 \hat{\mathbf{y}}(0) - \boldsymbol{\lambda}(0)\right) \end{aligned} \tag{2.20}$$

Next, we take the Fréchet derivative with respect to $\mathbf{u}(t)$ in some direction $\mathbf{z}_2(t)$, then we have

$$D_u\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = \int_0^T \mathbf{z}_2(t)^T \left(\mathbf{u}(t) - \boldsymbol{\lambda}(t)\right) \mathrm{d}t \tag{2.21}$$

Lastly, we take the Fréchet derivative with respect to $\boldsymbol{\lambda}(t)$ in some direction $\mathbf{z}_3(t)$, then

$$D_\lambda\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = \int_0^T \mathbf{z}_3(t)^T \left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) - \mathbf{u}(t) - \mathbf{f}(t)\right) \mathrm{d}t \tag{2.22}$$

If we have our optimal solutions of state variable $\mathbf{y}(t)$, adjoint variable $\boldsymbol{\lambda}(t)$, and control variable $\mathbf{u}(t)$, then for any allowable directions $\mathbf{z}_i(t)$, the following must hold

$$D_y\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = D_u\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = D_\lambda\mathcal{L}(\mathbf{y},\mathbf{u},\boldsymbol{\lambda}) = 0 \tag{2.23}$$

Hence, we can see that (2.20), (2.21), and (2.22) vanish with all allowable $\mathbf{z}_i(t)$. Therefore, we can conclude the following first-order necessary optimality conditions

of our interested optimization problems

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) = \mathbf{u}(t) + \mathbf{f}(t) & , t \in [0,T] \\ \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\lambda}(t) - A^T\boldsymbol{\lambda}(t) = \alpha_1\left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right) & , t \in [0,T] \\ \boldsymbol{\lambda}(t) = \mathbf{u}(t) & , t \in [0,T] \\ \alpha_3\mathbf{y}(0) - \boldsymbol{\lambda}(0) = \alpha_3\hat{\mathbf{y}}(0) \\ \alpha_2\mathbf{y}(T) + \boldsymbol{\lambda}(T) = \alpha_2\hat{\mathbf{y}}(T) \end{cases} \tag{2.24}$$

Note that one can eliminate the control variable $\mathbf{u}(t)$ easily to form a coupled forward-backward system of ODE as the optimality system. We have

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\boldsymbol{\lambda}(t)\\ \mathbf{y}(t)\end{pmatrix} + \begin{pmatrix}-A^T & -\alpha_1 I\\ -I & A\end{pmatrix}\begin{pmatrix}\boldsymbol{\lambda}(t)\\ \mathbf{y}(t)\end{pmatrix} = \begin{pmatrix}-\alpha_1\hat{\mathbf{y}}(t)\\ \mathbf{f}(t)\end{pmatrix} \tag{2.25}$$

with initial and terminal conditions

$$\begin{cases} \alpha_3\mathbf{y}(0) - \boldsymbol{\lambda}(0) = \alpha_3\hat{\mathbf{y}}(0) \\ \alpha_2\mathbf{y}(T) + \boldsymbol{\lambda}(T) = \alpha_2\hat{\mathbf{y}}(T) \end{cases} \tag{2.26}$$

We define the first row of (2.25) with the terminal condition in (2.26) as the *backward adjoint equation*:

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\lambda}(t) - A^T\boldsymbol{\lambda}(t) = \alpha_1\left(\mathbf{y}(t) - \hat{\mathbf{y}}(t)\right) \\ \alpha_2\mathbf{y}(T) + \boldsymbol{\lambda}(T) = \alpha_2\hat{\mathbf{y}}(T) \end{cases} \tag{2.27}$$

The second row of (2.25) with the initial condition in (2.26) is defined as the *forward state equation*:

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) = \boldsymbol{\lambda}(t) + \mathbf{f}(t) \\ \alpha_3\mathbf{y}(0) - \boldsymbol{\lambda}(0) = \alpha_3\hat{\mathbf{y}}(0) \end{cases} \tag{2.28}$$

Now, we need to discretize (2.27) and (2.28) as we want to solve this coupled forward-backward system of ODE numerically. Again, we will apply the implicit Euler's method for the discretization with respect to time.

Similar to the *discretize-then-optimize* approach, we let $\mathbf{y}^{(k)}$, $\mathbf{u}^{(k)}$, $\boldsymbol{\lambda}^{(k)}$, $\hat{\mathbf{y}}^k$, and $\mathbf{f}^{(k)}$ be the k-th time-step of $\mathbf{y}(t)$, $\mathbf{u}(t)$, $\boldsymbol{\lambda}(t)$, $\hat{\mathbf{y}}(t)$, and $\mathbf{f}(t)$ respectively, where $k = 0, 1, ..., n$.

We can discretize the *forward state equation* (2.12) by backward Euler's method in time. We obtain

$$\begin{cases} (\tau A + I)\,\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)} - \tau\boldsymbol{\lambda}^{(k+1)} = \tau\mathbf{f}^{(k+1)} \\ \alpha_3\mathbf{y}^{(0)} - \boldsymbol{\lambda}^{(0)} = \alpha_3\hat{\mathbf{y}}^{(0)} \end{cases} \tag{2.29}$$

where τ is the step-size for our time domain $[0, T]$. One can observe that this iterative equations (2.29) is exactly the same as the second row of the all-at-once system (2.10). Therefore, we successfully matched the *forward state equation* in two approaches. However, this is not the case for the *backward adjoint equation*. By forward Euler's method, one can discretize the *backward adjoint equation* (2.12) and obtain.

$$\begin{cases} (\tau A^T + I)\boldsymbol{\lambda}^{(k-1)} - \boldsymbol{\lambda}^{(k)} + \alpha_1\tau\mathbf{y}^{(k-1)} = \alpha_1\tau\hat{\mathbf{y}}^{(k-1)} \\ \alpha_2\mathbf{y}^{(n)} + \boldsymbol{\lambda}^{(n)} = \alpha_2\hat{\mathbf{y}}^{(n)} \end{cases} \tag{2.30}$$

We can see that the above iterative equations (2.30) match with the first row of the all-at-once system (2.10) except the final time-step. In fact, we can show that this so-called shifted final condition satisfy the first order accuracy in τ. Hence, the final conditions of these two approach will be matched as $\tau \to 0$. Note that the final time-step in (2.10) read as

$$(\tau A^T + I)\boldsymbol{\lambda}^{(n)} + (\alpha_1\tau + \alpha_2)\mathbf{y}^{(n)} = (\alpha_1\tau + \alpha_2)\hat{\mathbf{y}}^{(n)} \tag{2.31}$$

If we define $\boldsymbol{\lambda}^{(n+1)}$ according to the Euler's scheme, we have

$$(\tau A^T + I)\boldsymbol{\lambda}^{(n)} - \boldsymbol{\lambda}^{(n+1)} + \alpha_1\tau\mathbf{y}^{(n)} = \alpha_1\tau\hat{\mathbf{y}}^{(n)} \tag{2.32}$$

Pulling (2.31) and (2.32) together, one can deduce

$$\alpha_2\mathbf{y}^{(n)} + \boldsymbol{\lambda}^{(n+1)} = \alpha_2\hat{\mathbf{y}}^{(n)}$$

as a final condition for the all-at-once system (2.10) in the *discretize-then-optimize* approach, which is off by one time-step compare to the final condition (2.30) in the *optimize-then-discretize* approach. Since the Euler's scheme used in (2.32) have the following property

$$\boldsymbol{\lambda}^{(n+1)} = \boldsymbol{\lambda}^{(n)} + \tau\left(A^T\boldsymbol{\lambda}^{(n)} + \alpha_1\mathbf{y}^{(n)} - \alpha_1\hat{\mathbf{y}}^{(n)}\right) = \boldsymbol{\lambda}^{(n)} + O(\tau)$$

We can conclude that the *discretize-then-optimize* approach and the *optimize-then-discretize* approach is shifted by an error of $O(\tau)$ in the final condition, and hence two approaches will match exactly when $\tau \to 0$.

We can see that the difference between this approach is neglectable since most likely a small time-steps is applied. As mentioned previously, we chose to use the all-at-once system (2.10), (2.13), and (2.16) obtained from the *discretize-then-approach* in the rest of this thesis.

2.2 Shooting Method

In section 2.1, we can observe that the optimality systems that we want to solve ((2.10), (2.13), and (2.16)) could be very large depends on the grid-size of space domain and time domain. We propose the shooting method to reduce the size of such system. It also work well with the iterative solvers as it would output a matrix-vector multiplication. The idea is to shoot the state variable backward and then forward in time via the *forward state equation* and *backward adjoint equation* for our problems. Notice that the shooting output should match with the input if we have the correct initial guess. Furthermore, the mapping should be linear as we are considering linear parabolic PDE as our constraints. Therefore, we can formulate a linear system in much lower dimension and work well with the iterative methods.

2.2.1 Shooting Method for Target States Problems

Consider the target states problems in this subsection. We use shooting method to solve the optimality linear system (2.13). Consider the mapping $\mathcal{P}_1$ that take the final state $\mathbf{y}^{(n)}$ as an input. This mapping returns $\mathcal{P}_1\left(\mathbf{y}^{(n)}\right) = \left[\tilde{\boldsymbol{\lambda}}\right]$ by integrating the *backward adjoint equation* (2.14) backward in time. Next, we have another mapping $\mathcal{Q}_1$ that takes $\tilde{\boldsymbol{\lambda}}$ as an input. $\mathcal{Q}_1$ will return an output $\mathcal{Q}_1\left(\tilde{\boldsymbol{\lambda}}\right) = \left[\mathbf{y}^{(n)}\right]$ by solving the *forward state equation* (2.15). Finally, we define a mapping $\mathcal{R}_1$ as $\mathcal{R}_1 = \mathcal{Q}_1(\mathcal{P}_1(\mathbf{x}))$ for some vector $\mathbf{x} \in \mathbb{R}^m$. Since the mapping $\mathcal{R}_1$ is linear, we have $\mathcal{R}_1(\mathbf{x}) = L_1\mathbf{x} + \mathbf{r_1}$

for some matrix operator L_1 and vector $\mathbf{r_1}$. For consistency, we have

$$\mathcal{R}_1(\mathbf{y}^{(n)}) = \mathbf{y}^{(n)}$$

Hence, we have a linear system

$$(L_1 - I)\mathbf{y}^{(n)} = -\mathbf{r_1} \tag{2.33}$$

We will apply the *flexible generalized minimal residual* (FGMRES) method to solve (2.33), and hence solve the optimal control problem by calculating the control variable and adjoint variable with correct $\mathbf{y}^{(n)}$. An introduction for FGMRES is given in section 2.3.

2.2.2 Shooting Method for Tracking Problems and All-time Problems

We move on to discus the shooting method for tracking problems and all-time problems. We use shooting method to solve the optimality linear system (2.10) and (2.16). The formulation is very similar to the target state problems, except that we now need to shoot the entire time-steps $\tilde{\mathbf{y}}$ instead of just $\mathbf{y}^{(n)}$ since the tracking term exist. Denote $\mathcal{P}_2$ as a mapping that take $\tilde{\mathbf{y}}$ as input. It will return $\mathcal{P}_2(\tilde{\mathbf{y}}) = \left[\tilde{\boldsymbol{\lambda}}\right]$ by solving the *backward adjoint equation* (2.17) (Or a similar mapping $\mathcal{P}_3$ for the all-time case solving (2.11)). And denote $\mathcal{Q}_2$ as a mapping that take $\tilde{\boldsymbol{\lambda}}$ as input. This mapping will gives $\mathcal{Q}_2(\tilde{\boldsymbol{\lambda}}) = [\tilde{\mathbf{y}}]$ by solving the *forward state equation* (2.18) (Or a similar mapping $\mathcal{Q}_3$ for the all-time case solving (2.12)). We define the mapping $\mathcal{R}_i$ as $\mathcal{R}_i = \mathcal{Q}_i(\mathcal{P}_i(\mathbf{x}))$, $i = 2, 3$, for some vector $\mathbf{x} \in \mathbb{R}^{mn}$. Since the mapping $\mathcal{R}_i$ is linear, we have $\mathcal{R}_i(\mathbf{x}) = L_i\mathbf{x} + \mathbf{r_i}$ for some matrix operator L_i and vector $\mathbf{r_i}$. For consistency, we have

$$\mathcal{R}_i(\tilde{\mathbf{y}}) = \tilde{\mathbf{y}}$$

Hence, we have a linear system for tracking problems

$$(L_2 - I)\tilde{\mathbf{y}} = -\mathbf{r_2} \tag{2.34}$$

and a linear system for all-time problems

$$(L_3 - I)\tilde{\mathbf{y}} = -\mathbf{r_3} \tag{2.35}$$

Again, we apply FGMRES to solve (2.34) and (2.35), and hence solve the optimal control problem by calculating the control variable and adjoint variable with correct $\tilde{\mathbf{y}}$.

2.2.3 Computation of Shooting Method

In this section, we will demonstrate how to apply FGMRES method on the linear system with the given mapping. A detailed review on FGMRES method is presented in the next section 2.3. It is well-known that with FGMRES method, we do not have to compute the entire matrix $L_i - I$ explicitly. Here in this section, we will present two algorithms that are used to find the matrix handle $(L_i - I)\mathbf{x}$ with given input vector $\mathbf{x}$, and vector $\mathbf{r_i}$ for $i = 1, 2, 3$. We take $i = 1$ for target state problems, $i = 2$ for tracking problems, and $i = 3$ for all-time problems.

Algorithm 1: Right-hand-side vector for FGMRES

Input : zero vector $\mathbf{0}$

Output: $\mathbf{r_i}$

$\mathbf{a} = \mathcal{P}_i(\mathbf{0})$;

$\mathbf{b} = \mathcal{Q}_i(\mathbf{a})$;

$\mathbf{r_i} = -\mathbf{b}$;

Algorithm 2: Function handle of left-hand-side matrix for FGMRES

Input : $\mathbf{x}$, $\mathbf{r_i}$

Output: $\mathbf{z}$, where $\mathbf{z} = (L_i - I)\,\mathbf{x}$

$\mathbf{a} = \mathcal{P}_i(\mathbf{x})$;

$\mathbf{b} = \mathcal{Q}_i(\mathbf{a})$;

$\mathbf{z} = \mathbf{b} - \mathbf{r_i} - \mathbf{x}$;

2.3 Iterative Methods for System of Linear Equations

In previous sections, we showed how to derive and discretize the first order necessary conditions from our interested parabolic optimal control problems. These first order necessary conditions are represented as a large optimality system, which is then

reduced to a smaller system of linear equations, (2.33), (2.34) or (2.35), with the help of shooting method. Now, in this section, we are going to introduce an iterative method to solve this system of linear equations. There are many iterative solvers available in the literature. In fact, it is a very active research topic in the scientific computing community [43]. Among all the available solvers, the *generalized minimal residual* (GMRES) method is perhaps one of the most popular and reliable methods as it is well suitable for the general large and sparse systems. In this research, we chose the *flexible generalized minimal residual* (FGMRES) method to solve our linear systems. We will give a brief review on this method as well as the preconditioning technique in this section.

If readers are interested in studying the iterative methods for linear systems and the preconditioning theories, a comprehensive introduction could be found in the book [17], [18], or [42].

2.3.1 Krylov Subspace Method

GMRES and FGMRES is in the class of the Krylove subspace methods. We will give a short review on the Krylov subspace method here. Consider a system of linear equation

$$\mathcal{A}\mathbf{x} = \mathbf{b}$$

where $\mathcal{A} \in \mathbb{R}^{N\times N}$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^N$ for some positive integer N. Assume that we have an initial guess of the solution, say $\mathbf{x}_0$. Let $\mathbf{v}_0 = b - \mathcal{A}\mathbf{x}_0$ be the residual. The Krylov subspace is defined as

$$\mathcal{K}_m(\mathcal{A}, \mathbf{v_0}) = \text{span}\{\mathbf{v}_0, \mathcal{A}\mathbf{v}_0, ..., \mathcal{A}^{m-1}v_0\}$$

Krylov subspace method look for an approximate solution $\mathbf{x}_m$, which is in the Krylov subspace $\mathcal{K}_m$, by imposing the Petrov-Galerkin condition

$$\mathbf{b} - \mathcal{A}\mathbf{x}_m \perp \mathcal{Q}_m$$

where $\mathcal{Q}_m$ is some subspace of dimension m. Different Krylov subspace method means a different choice of $\mathcal{Q}_m$. For example, the GMRES method takes $\mathcal{Q}_m = \mathcal{A}\mathcal{K}_m$.

2.3.2 Preconditioning and FGMRES

As mentioned previously, the FGMRES method is our choice in this research for solving our linear systems. FGMRES method is a variant of the GMRES method. The GMRES method is first presented in Saad's famous paper [40]. The convergence rate of GMRES is controlled by the enigenvalues distribution of the matrix $\mathcal{A}$ as we can see in the following theorem (1).

Theorem 1. *(Saad and Schultz, [40]). Suppose that $\mathcal{A} \in \mathbb{R}^{N\times N}$ is diagonalizable. Let $\mathcal{A} = \mathcal{U}\mathcal{D}\mathcal{U}^{-1}$, where $\mathcal{D}$ is a diagonal matrix with entries $d_1, ..., d_m, d_{m+1}, ..., d_N$. Also assume that $\{d_1, ..., d_m\}$ has non-positive real parts and $\{d_{m+1}, ..., d_N\}$ is bounded in a circle, where this circle is centered at $c > 0$ with a radius r such that $r < c$. Then the residual $\mathbf{v}_{i+1}$ at $(i+1)$-th step will satisfy*

$$\|\mathbf{v}_{i+1}\|_2 \leq \|\mathcal{U}\|_2\|\mathcal{U}^{-1}\|_2 \left(\frac{d_{max}}{d_{min}}\right)^m \left(\frac{r}{c}\right)^{i-m} \|\mathbf{v}_0\|_2$$

where $d_{min} = \min\limits_{1\leq i\leq m} |d_i|$ and $d_{max} = \max\limits_{1\leq i\leq m\leq j\leq n} |d_i - d_j|$.

We aim to improve the convergence rate of GMRES with a preconditioner $\mathcal{P}^{-1}$. The idea of preconditioning is to alter the linear system $\mathcal{A}\mathbf{x} = \mathbf{b}$ into an equivalent form

$$\mathcal{A}\mathcal{P}^{-1}\left(\mathcal{P}\mathbf{x}\right) = \mathbf{b}$$

If we design our preconditioner properly, the preconditioned matrix $\mathcal{A}\mathcal{P}^{-1}$ would have a better eigenvalues distribution compare to the original matrix $\mathcal{A}$. It is well-known that a good preconditioner $\mathcal{P}^{-1}$ for GMRES should have the following two properties:

- The spectrum of $\mathcal{A}\mathcal{P}^{-1}$ is clustered around one.
- The computational cost of $\mathcal{P}^{-1}$ is low.

Note that in our implementation, the preconditioner $\mathcal{P}^{-1}$ would not be found explicitly. Instead, we will have an operation to output $\mathcal{P}^{-1}\mathbf{z}$ for an arbitrary vector $\mathbf{z}$. As discussed in [42], the GMRES method may not be suitable in our implementation. Therefore, we would use FGMRES [39] throughout this research. A standard algorithm of FGMRES could be found in (Chapter 9, [42]).

Chapter 3

Preconditioners for Parabolic Optimal Control Problems with Symmetric Differential Operator

The linear systems that we want to solve ((2.33), (2.34), and (2.35)) are difficult to compute in many scenarios. Both direct and iterative methods could be ineffective to solve these systems. One can observe that the linear system obtained by shooting method could be very large in practical problems. The dimension of the system grows significantly when a finer mesh is used. In fact, the linear system (2.34) and (2.35) has a size of mn by mn, where m and n are the degree of freedom of space discretization and time discretization respectively. Even in the target states problems, where we only shoot the final time slice $\mathbf{y}^{(n)}$, the linear system (2.33) could still be very large (with a size of n by n to be precise) if we apply a thin space grid. Therefore, a faster or more economical method is desired. As we described in section 2.3, a good preconditioner could seal the deal. Note that a good preconditioner is highly depended on the structure of the linear system. Therefore, we will have different approach if the differential matrix operator A in our optimal control problems has different properties. In this chapter, we will consider the case that A is symmetric (and positive semi-definite, as pre-assumed in section 1.4). In section 3.1, we will develop a preconditioner for the target states problems with symmetric operator. Another preconditioner will be proposed for tracking problems with symmetric oper-

ator in section 3.2. The preconditioner for all-time problems is given in section 3.3. Lastly, we will discuss the performances of our preconditioners in section 3.4.

3.1 Preconditioner for Target States Problems with Symmetric Differntial Operator

Recall our target state optimal control problem defined in section 1.4.1.

$$J(\mathbf{y},\mathbf{u}) = J_1(\mathbf{y},\mathbf{u}) = \frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t + \frac{\alpha_2}{2}\|\mathbf{y}(T)-\hat{\mathbf{y}}(T)\|_2^2 + \frac{\alpha_3}{2}\|\mathbf{y}(0)-\hat{\mathbf{y}}(0)\|_2^2 \quad (1.5)$$

subject to a system of ODE obtained by applying spatial discretization on a parabolic PDE

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) = \mathbf{u}(t) + \mathbf{f}(t), \quad t \in [0,T] \quad (1.4)$$

In this section, we will take A as a symmetric matrix, i.e., $A^T = A$. As we already demonstrated in chapter 2, we are going to solve the corresponding linear system

$$(L_1 - I)\,\mathbf{y}^{(n)} = -\mathbf{r_1} \quad (2.33)$$

in order to find the optimality conditions for our desired optimal control problem. In the following section, we will derive a preconditioner P_1^{-1} for (2.33). The idea is to integrate $\mathbf{y}^{(n)}$ backward and then forward by solving the *backward adjoint equation* and the *forward state equation* by integrating factor method directly. Hence, we can write the matrix $L_1 - I$ in terms of A. The preconditioner P_1^{-1} is formulated by the approximation of $(L_1 - I)^{-1}$ when the matrix is at high frequency and low frequency.

3.1.1 Derivation of the Preconditioner

Now we start to derive our preconditioner P_1^{-1}. Consider the eigenvalue decomposition $A = QDQ^{-1}$, and we let $\boldsymbol{\mu}(t) = Q^{-1}\boldsymbol{\lambda}(t)$. We have a substituted version of the *backward adjoint equation* (2.14)

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\mu}(t) - D\boldsymbol{\mu}(t) = 0 \\ \boldsymbol{\mu}(T) = \alpha_2 Q^{-1}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right) \end{cases} \quad (3.1)$$

Suppose we take d_i to be the eigenvalues of A, and we use the notation $[\mathbf{x}]_i$ to denote the i-th element of vector $\mathbf{x}$. We have n-copies of ODE from the above (3.1)

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\mu}_i - d_i\boldsymbol{\mu}_i = 0 \\ \boldsymbol{\mu}_i(T) = \alpha_2 \left[Q^{-1}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)\right]_i \end{cases}$$

By integrating factor method,

$$e^{-d_i t}\boldsymbol{\mu}_i = C_i$$

We can find the constant C_i with the terminal condition

$$e^{-d_i T}\left[\alpha_2 Q^{-1}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)\right]_i = C_i$$

So we obtained

$$\boldsymbol{\mu}_i = \alpha_2 e^{d_i(t-T)}\left[Q^{-1}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)\right]_i$$

Hence, we get the solution of (3.1)

$$\boldsymbol{\mu}(t) = \alpha_2 e^{A(t-T)}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)$$

Now, we let $\boldsymbol{\xi}(t) = Q^{-1}\mathbf{y}(t)$. Then again, we have a substituted version of the *forward state equation* (2.15)

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\xi}(t) + D\boldsymbol{\xi}(t) = \boldsymbol{\mu}(t) + Q^{-1}\mathbf{f}(t) \\ \boldsymbol{\xi}(0) = \frac{1}{\alpha_3}\boldsymbol{\mu}(0) + Q^{-1}\hat{\mathbf{y}}(0) \end{cases} \tag{3.2}$$

Similarly, we have n-copies of ODE from the above (3.2)

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\xi}_i + d_i\boldsymbol{\xi}_i = \alpha_2 e^{d_i(t-T)}\left[Q^{-1}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)\right]_i + \left[Q^{-1}\mathbf{f}(t)\right]_i$$

Again, by integrating factor method

$$\begin{aligned} (e^{d_i t}\boldsymbol{\xi}_i) &= \frac{\alpha_2}{2d_i}e^{d_i(2t-T)}\left[Q^{-1}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)\right]_i + \frac{1}{d_i}e^{d_i t}\left[Q^{-1}\mathbf{f}(t)\right]_i + C_i \\ \implies \boldsymbol{\xi}_i &= \frac{\alpha_2}{2d_i}e^{d_i(t-T)}\left[Q^{-1}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)\right]_i + \frac{1}{d_i}\left[Q^{-1}\mathbf{f}(t)\right]_i + e^{-d_i t}C_i \end{aligned}$$

From the initial condition, we can find the constant C_i

$$C_i = \left(\frac{\alpha_2}{\alpha_3} - \frac{\alpha_2}{2d_i}\right)e^{-d_i T}\left[Q^{-1}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)\right]_i + \left[Q^{-1}\hat{\mathbf{y}}(0)\right]_i - \frac{1}{d_i}\left[Q^{-1}\mathbf{f}(0)\right]$$

Therefore, we have

$$\boldsymbol{\xi}_i = \frac{\alpha_2}{2d_i} e^{d_i(t-T)} Q^{-1} \left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right) + \frac{1}{d_i}\left[Q^{-1}\mathbf{f}(t)\right]_i$$
$$+ \left(\frac{\alpha_2}{\alpha_3} - \frac{\alpha_2}{2d_i}\right) e^{-d_i(t+T)} \left[Q^{-1}\left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)\right]_i + e^{-d_i t}\left[Q^{-1}\hat{\mathbf{y}}(0)\right]_i - \frac{1}{d_i}\left[Q^{-1}\mathbf{f}(0)\right]$$

Hence, we have the solution

$$\mathbf{y}(t) = \frac{\alpha_2}{2} A^{-1} e^{A(t-T)} \left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right) + A^{-1}\mathbf{f}(t) + \frac{\alpha_2}{\alpha_3} e^{-A(t+T)} \left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right)$$
$$- \frac{\alpha_2}{2} A^{-1} e^{-A(t+T)} \left(\hat{\mathbf{y}}(T) - \mathbf{y}(T)\right) + e^{-At}\hat{\mathbf{y}}(0) - A^{-1}\mathbf{f}(0)$$

If we put $t = T$, we can obtain the following

$$\mathbf{y}(T) = \left[-\frac{\alpha_2}{2} A^{-1} - \frac{\alpha_2}{\alpha_3} e^{-2TA} + \frac{\alpha_2}{2} A^{-1} e^{-2TA}\right] \mathbf{y}(T)$$
$$+ \left[\frac{\alpha_2}{2} A^{-1} + \frac{\alpha_2}{\alpha_3} e^{-2TA} - \frac{\alpha_2}{2} A^{-1} e^{-2TA}\right] \hat{\mathbf{y}}(T) + A^{-1}\mathbf{f}(T) + e^{-AT}\hat{\mathbf{y}}(0) - A^{-1}\mathbf{f}(0)$$

By comparing to the linear system (2.33) that we formulated in section 2.2, one can observe that

$$\begin{cases} L_1 - I &= -\frac{\alpha_2}{2} A^{-1} - \frac{\alpha_2}{\alpha_3} e^{-2TA} + \frac{\alpha_2}{2} A^{-1} e^{-2TA} - I \\ &= \frac{\alpha_2}{2} A^{-1}\left(e^{-2TA} - I\right) - \frac{\alpha_2}{\alpha_3} e^{-2TA} - I \\ \mathbf{r_1} &= \left[\frac{\alpha_2}{2} A^{-1} + \frac{\alpha_2}{\alpha_3} e^{-2TA} - \frac{\alpha_2}{2} A^{-1} e^{-2TA}\right] \hat{\mathbf{y}}(T) + A^{-1}\mathbf{f}(T) + e^{-AT}\hat{\mathbf{y}}(0) - A^{-1}\mathbf{f}(0) \end{cases} \tag{3.3}$$

We can observe that all the known vectors in the problems, i.e., $\hat{\mathbf{y}}$ and $\mathbf{f}$, went to the right-hand-side vector $\mathbf{r_1}$. Now, we have found a expression of $L_1 - I$ in terms of A. We want to find a good approximation to $L_1 - I$ when A has high (or low) frequency so we can invert it nicely. Suppose A has high frequency, then

$$L_1 - I \approx -\frac{\alpha_2}{2} A^{-1} - I$$

Therefore we have an approximation of inverse of $L - I$ when A has high frequency

$$(L_1 - I)^{-1} \approx -\left(\frac{\alpha_2}{2} I + A\right)^{-1} A \tag{3.4}$$

On the other hand, if A has low frequency, then

$$\begin{aligned} L_1 - I &= -\alpha_2 A^{-1} e^{-TA} \left(\frac{e^{TA} - e^{-TA}}{2} \right) - \frac{\alpha_2}{\alpha_3} e^{-2TA} - I \\ &= -\alpha_2 A^{-1} e^{-TA} \sinh(TA) - \frac{\alpha_2}{\alpha_3} e^{-2TA} - I \\ &= -\alpha_2 T \left[\frac{A^{-1}}{T} \sinh(TA) \right] - \frac{\alpha_2}{\alpha_3} e^{-2TA} - I \\ &\approx -\left(\alpha_2 T + \frac{\alpha_2}{\alpha_3} + 1 \right) I \end{aligned}$$

Therefore we have the approximation of inverse of $L_1 - I$ when A has low frequency,

$$(L_1 - I)^{-1} \approx -\left(\frac{1}{\alpha_2 T + \frac{\alpha_2}{\alpha_3} + 1} \right) I \tag{3.5}$$

We approximate $(L_1 - I)^{-1}$ by (3.4) + (3.5), so our preconditioner would be

$$P_1^{-1} = -\left(\frac{\alpha_2}{2} I + A \right)^{-1} A - \left(\frac{1}{\alpha_2 T + \frac{\alpha_2}{\alpha_3} + 1} \right) I \tag{3.6}$$

Since both high and low frequencies are smoothed out, we expect the iteration method could be speeded up by this preconditioner P_1^{-1}

3.1.2 Eigenvalues Analysis of the Preconditioned System

In this section, we aim to find an eignevalues bound of our preconditioned system. Let x be the eigenvalues of A. Note that $x \geq 0$ as all eigenvalues of A should be greater than or equal to zero according to our assumption as A is positive semi-definite. We have the eigenvalues function f and g for the matrix $L_1 - I$ (3.3) and the preconditioner P_1^{-1} (3.6) respectively.

$$\begin{cases} f(x) = \frac{\alpha_2}{2x} \left(e^{-2Tx} - 1 \right) - \frac{\alpha_2}{\alpha_3} e^{-2Tx} - 1 \\ g(x) = \frac{-x}{\frac{\alpha_2}{2} + x} - \frac{1}{\alpha_2 T + \frac{\alpha_2}{\alpha_3} + 1} \end{cases}$$

We are going to find the upper and lower bound for both f and g, and hence find the bounds of $f \cdot g$. Firstly, since

$$-1 < \frac{-x}{\frac{\alpha_2}{2} + x} \leq 0, \quad \forall x \geq 0$$

We have

$$-1-\frac{1}{\alpha_2 T+\frac{\alpha_2}{\alpha_3}+1}<g(x)\leq-\frac{1}{\alpha_2 T+\frac{\alpha_2}{\alpha_3}+1},\quad \forall x\geq 0 \tag{3.7}$$

Secondly, for all $x>0$, consider

$$\begin{aligned}
f'(x)&=\frac{\alpha_2}{2}\left(\frac{-2Txe^{-2Tx}-e^{-2Tx}+1}{x^2}\right)+\frac{2\alpha_2 T}{\alpha_3}e^{-2Tx}\\
&=\frac{\alpha_2}{2}\left(\frac{e^{2Tx}-(1+2Tx)}{x^2e^{2Tx}}\right)+\frac{2\alpha_2 T}{\alpha_3}e^{-2Tx}\\
&>0
\end{aligned}$$

So f is strictly increasing on $(0,+\infty)$. Note that f can be represented by a hyperbolic function,

$$f(x)=\frac{\alpha_2}{2x}\left(e^{-2Tx}-1\right)-\frac{\alpha_2}{\alpha_3}e^{-2Tx}-1=-\alpha_2 T\left(\frac{\sinh(Tx)}{e^{Tx}(Tx)}\right)-\frac{\alpha_2}{\alpha_3}e^{-2Tx}-1$$

By L'Hospital's rule, we obtain

$$\lim_{x\to 0}\left(\frac{\sinh(Tx)}{e^{Tx}(Tx)}\right)=\lim_{x\to 0}\left(\frac{\cosh(Tx)}{e^{Tx}+Txe^{Tx}}\right)=1$$

Therefore,

$$\lim_{x\to 0}f(x)=-\alpha_2 T-\frac{\alpha_2}{\alpha_3}-1 \tag{3.8}$$

On the other hand,

$$\lim_{x\to\infty}f(x)=\lim_{x\to\infty}\left(\frac{\alpha_2}{2x}\left(e^{-2Tx}-1\right)-\frac{\alpha_2}{\alpha_3}e^{-2Tx}-1\right)=-1 \tag{3.9}$$

Consider the limit of f (3.8) and (3.9) with the fact that f is strictly increasing on $(0,+\infty)$, we have

$$-\alpha_2 T-\frac{\alpha_2}{\alpha_3}-1\leq f(x)<-1,\quad \forall x\geq 0 \tag{3.10}$$

Combining (3.7) and (3.10), we have the bounds of the preconditioned system

$$\frac{1}{\alpha_2 T+\frac{\alpha_2}{\alpha_3}+1}<f\cdot g<\alpha_2 T+\frac{\alpha_2}{\alpha_3}+2,\quad \forall x\geq 0$$

This bound allows us to estimate eigenvalues of the error propagation matrix, which in turn sheds light on the performance of our method. The idea here is based on the fact that the FGMRES method will always has same or better convergence performance compare to the stationary method. Therefore, if our preconditioner has

certain convergence rate for stationary method, then it should also has the same or better convergence rate for FGMRES method. Consider the stationary iterative method for the linear system $\mathcal{A}\mathbf{x} = \mathbf{b}$ with preconditioner $\mathcal{P}^{-1}$.

$$\mathcal{P}\mathbf{x}^{k+1} = \mathcal{P}\mathbf{x}^k + \mathbf{b} - \mathcal{A}\mathbf{x}^k$$

We have the following iterative formula

$$\mathbf{x}^{k+1} = (I - \mathcal{P}^{-1}\mathcal{A})\mathbf{x}^k + \mathcal{P}^{-1}\mathbf{b}$$

Let assume e^k to be the error of k-th step, then

$$e^{k+1} = (I - \mathcal{P}^{-1}\mathcal{A})e^k$$

The matrix $\mathcal{E} = I - \mathcal{P}^{-1}\mathcal{A}$ is known as the error propagation matrix of stationary method. We can observe that the error propagation matrix of our interested system read as

$$\mathcal{E} = I - P_1^{-1}(L_1 - I)$$

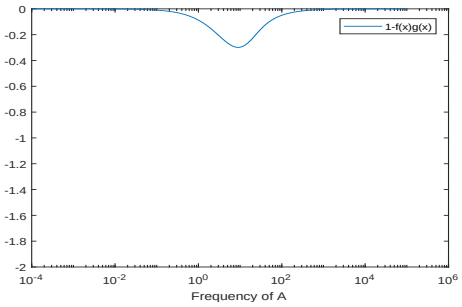
Hence the contraction factor $\rho = \max_{x \in \sigma(A)} |1 - f(x)g(x)|$ could indicate the convergence performance of our preconditioner. We can see this contraction factor against the eigenvalues of A in figure (3.1). We can observe that in most of these testing parameters, our preconditioner smooth out the high and low frequencies successfully. Our numerical example showed that the performance is still good even for the cases that some frequencies are not smoothed out, as those frequencies are handled by FGMRES.

3.1.3 Computational Cost of P_1^{-1}

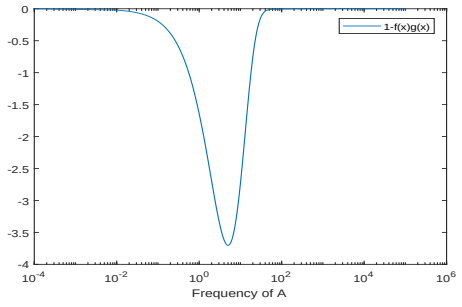
Computational cost is an important issue for preconditioner. Obviously, we would like to have a preconditioner with a low computational cost. Recall our preconditioner

$$P_1^{-1} = -\left(\frac{\alpha_2}{2}I + A\right)^{-1} A - \left(\frac{1}{\alpha_2 T + \frac{\alpha_2}{\alpha_3} + 1}\right) I \tag{3.6}$$

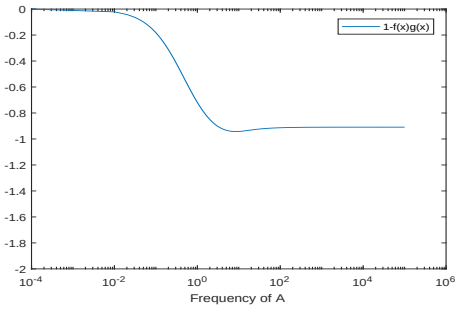
The computation of $P_1^{-1}\mathbf{x}$ include a direct multiplication $\mathbf{v} = A\mathbf{x}$ and solving a linear system $\left(\frac{\alpha_2}{2}I + A\right)\mathbf{w} = \mathbf{v}$, for some vector $\mathbf{x}$, $\mathbf{v}$, and $\mathbf{w}$. Of course there is a constant addition too. As the matrix A is a sparse matrix obtained from the



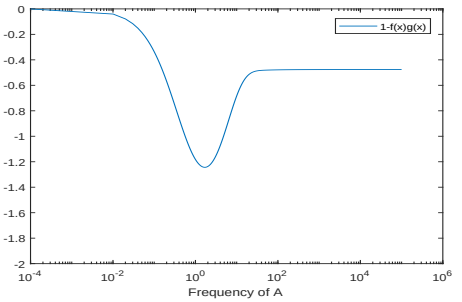
(a) $T = 0.1$, $\alpha_2 = 10^5$, $\alpha_3 = 10^5$



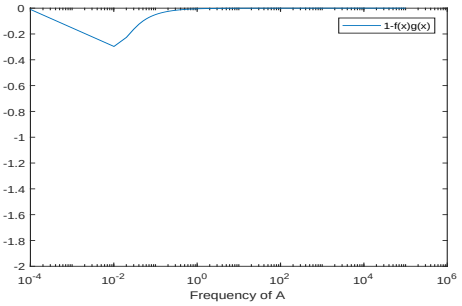
(b) $T = 0.1$, $\alpha_2 = 10^5$, $\alpha_3 = 1$



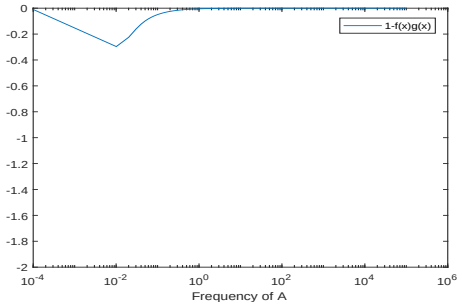
(c) $T = 0.1$, $\alpha_2 = 1$, $\alpha_3 = 10^5$



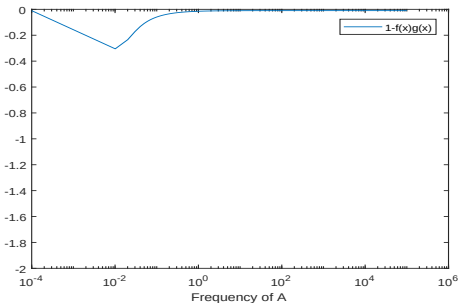
(d) $T = 0.1$, $\alpha_2 = 1$, $\alpha_3 = 1$



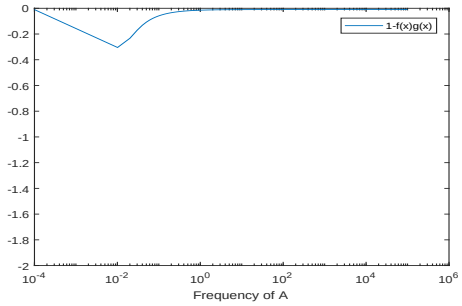
(e) $T = 100$, $\alpha_2 = 10^5$, $\alpha_3 = 10^5$



(f) $T = 100$, $\alpha_2 = 10^5$, $\alpha_3 = 1$



(g) $T = 100$, $\alpha_2 = 1$, $\alpha_3 = 10^5$



(h) $T = 100$, $\alpha_2 = 1$, $\alpha_3 = 1$

Figure 3.1: Contraction factor $\rho = \max_{x \in \sigma(A)} |1 - f(x)g(x)|$ against engienvalues of A.

spatial discretization of differential operator, the computation of $\left(\frac{\alpha_2}{2}I + A\right)^{-1}\mathbf{v}$ for any vector $\mathbf{v}$ is equivalent to an elliptic solver. As there exist some efficient elliptic solver (e.g. multi-grid method), the computational cost of our preconditioner P_1^{-1} is reasonable.

3.2 Preconditioner for Tracking Problems with Symmetric Differential Operator

In this section, we will propose a preconditioner for the tracking problem. Recall our cost functional define in section 1.4.2,

$$J(\mathbf{y},\mathbf{u}) = J_2(\mathbf{y},\mathbf{u}) = \frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|_2^2 \mathrm{d}t \tag{1.6}$$

subject to a system of ODE arose from the space-discretization of parabolic PDE with known initial condition

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) = \mathbf{u}(t) + \mathbf{f}(t), \quad t \in [0,T] \\ \mathbf{y}(0) = \mathbf{y}_0 \end{cases} \tag{3.11}$$

Once again, we take $A^T = A$ in this section. Our goal is to find a good preconditioner for the linear system (2.34).

$$(L_2 - I)\tilde{\mathbf{y}} = -\mathbf{r_2} \tag{2.34}$$

The idea of our preconditioner P_2^{-1} is inspired by the work from Pearson [33]. He constructed a block diagonal preconditioner to his fairly large all-at-once system. In this thesis, we followed his idea to formulate a preconditioner for our smaller linear system (2.34) obtained by shooting method. Another contribution of this section is that we proposed a more robust proof for the eignevalues bound.

3.2.1 Linear System of Shooting Method

Recall our all-at-once system for tracking problems

$$\begin{pmatrix} K^T & B_1 \\ -C_1 & K \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\lambda}} \\ \tilde{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{g}}_1 \\ \tilde{\mathbf{f}}_1 \end{pmatrix} \tag{2.16}$$

In this subsection, we are going to show that the shooting linear system (2.34) can be written in terms of the matrices and vectors in the all-at-once system. We can see that solving the *backward adjoint equation* (2.17) with input $\tilde{\mathbf{y}}$ (i.e. the mapping $\mathcal{P}_2$ in section 2.2) is equivalent to computing

$$\tilde{\boldsymbol{\lambda}} = K^{-T}(-B_1\tilde{\mathbf{y}} + \tilde{\mathbf{g}}_1) \tag{3.12}$$

Similarly, solving the *forward state equation* (2.18) with input $\tilde{\boldsymbol{\lambda}}$ (i.e. the mapping $\mathcal{Q}_2$ in section 2.2) is equivalent to computing

$$\tilde{\mathbf{y}} = K^{-1}(C_1\tilde{\boldsymbol{\lambda}} + \tilde{\mathbf{f}}_1) \tag{3.13}$$

Put (3.12) into (3.13) and make $\tilde{\mathbf{y}}$ as subject, we can formulate the shooting system

$$\left(-K^{-1}C_1K^{-T}B_1 - I\right)\tilde{\mathbf{y}} = -K^{-1}C_1K^{-T}\tilde{\mathbf{g}}_1 - K^{-1}\tilde{\mathbf{f}}_1$$

Therefore, the linear system in (2.34) can be written as

$$\begin{cases} L_2 - I = -K^{-1}C_1K^{-T}B_1 - I \\ \mathbf{r_2} = K^{-1}C_1K^{-T}\tilde{\mathbf{g}}_1 + K^{-1}\tilde{\mathbf{f}}_1 \end{cases} \tag{3.14}$$

3.2.2 Preconditioner and Eigenvalue Analysis

We now want to find a good approximation P_2 to

$$L_2 - I = -K^{-1}C_1K^{-T}B_1 - I$$

So we can apply P_2^{-1} as a preconditioner for the linear system (2.34). Also, we would like to find an eigenvalues bound for the preconditioned system. To do so, we are going to need the following theorem. This theorem is inspired by the work in [33].

Theorem 2. *Let A be a symmetric matrix and K is the matrix defined in (2.2). Suppose $\Delta = blkdiag(a_1I, ..., a_nI)$ where $a_i > 0$ for all $i = 1, ..., n$. Also assume that $a_1 \geq ... \geq a_n$. Then $\Delta K + K^T\Delta$ is positive definite.*

Proof. Let $\tilde{\mathbf{w}} := (\mathbf{w}_1^T, ..., \mathbf{w}_n^T)^T$ be any real vector. We have

$$
\begin{aligned}
&\tilde{\mathbf{w}}^T(\Delta K + K^T\Delta)\tilde{\mathbf{w}} \\
&= \begin{pmatrix} \mathbf{w}_1^T & \dots & \mathbf{w}_n^T \end{pmatrix}
\begin{pmatrix}
2a_1(\tau A + I) & -a_2 I & & \\
-a_2 I & \ddots & \ddots & \\
& \ddots & \ddots & -a_n I \\
& & -a_n I & 2a_n(\tau A + I)
\end{pmatrix}
\begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_n \end{pmatrix} \\
&= \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(2a_j(\tau A + I)\right) \mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\mathbf{w}_{j-1}^T(-a_j I)\mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\mathbf{w}_j^T(-a_j I)\mathbf{w}_{j-1}\right] \\
&= 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A\right) \mathbf{w}_j\right] + 2\sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j I\right) \mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\mathbf{w}_{j-1}^T(-a_j I)\mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\mathbf{w}_j^T(-a_j I)\mathbf{w}_{j-1}\right] \\
&= 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A\right) \mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\mathbf{w}_{j-1}^T \left(a_{j-1} I\right) \mathbf{w}_{j-1} - \mathbf{w}_{j-1} \left(a_j I\right) \mathbf{w}_j - \mathbf{w}_j^T \left(a_j I\right) \mathbf{w}_{j-1} + \mathbf{w}_j^T \left(a_j I\right) \mathbf{w}_j\right] \\
&\quad + \mathbf{w}_1^T \left(a_1 I\right) \mathbf{w}_1 + \mathbf{w}_n^T \left(a_n I\right) \mathbf{w}_n \\
&= 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A\right) \mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)^T \left(a_j I\right) \left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)\right] \\
&\quad + \sum_{j=2}^{n} \left[\mathbf{w}_{j-1}^T \left(\left(a_{j-1} - a_j\right) I\right) \mathbf{w}_{j-1}\right] + \mathbf{w}_1^T \left(a_1 I\right) \mathbf{w}_1 + \mathbf{w}_n^T \left(a_n I\right) \mathbf{w}_n \\
&\geq 0
\end{aligned}
$$

The equality sign holds only if all components are zeros. From

$$\mathbf{w}_1^T \left(a_1 I\right) \mathbf{w}_1 = \mathbf{w}_n^T \left(a_n I\right) \mathbf{w}_n = 0$$

we have $\mathbf{w}_1 = \mathbf{w}_n = 0$. Since we also have

$$\sum_{j=2}^{n} \left[\left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)^T \left(a_j I\right) \left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)\right] = 0$$

We can deduce that $\mathbf{w}_{j-1} = \mathbf{w}_j$ for $j = 2, ..., n$. By induction, we have $\tilde{\mathbf{w}} = (\mathbf{w}_1^T, ..., \mathbf{w}_n^T)^T = 0$. Since the equality sign only holds when $\tilde{\mathbf{w}} = 0$, the matrix $K\Delta + \Delta K^T$ is positive definite. □

With the above theorem, we are all set to derive our preconditioner now. Define

$$P_2 = T_1 N_1$$

where

$$N_1 = (K + B_1^{1/2} C_1^{1/2})^T C_1^{-1} (K + B_1^{1/2} C_1^{1/2})$$

and

$$T_1 = -K^{-1} C_1 K^{-T}$$

Notice that our preconditioner involves computing the matrix square root of B_1 and C_1. Although matrix square root is fairly complicated to compute generally, the matrix square root of B_1 and C_1 are extremely easy to compute given the fact that both B_1 and C_1 are diagonal matrices with non-negative entries. We are going to show that $\rho(P_2^{-1}(L_2 - I))$ is bounded by $[\frac{1}{2}, 1)$. Consider the eigenvalues of our preconditioned system

$$P_2^{-1}(L_2 - I)\mathbf{v} = \lambda \mathbf{v} \implies T_1^{-1}(L_2 - I)v = \lambda N_1 \mathbf{v}$$

Hence,

$$\lambda = \frac{\mathbf{v}^T (T_1^{-1}(L_2 - I))\mathbf{v}}{\mathbf{v}^T (N_1)\mathbf{v}}$$

So we have the Rayleigh quotient

$$\begin{aligned} R &= \frac{\mathbf{v}^T (T_1^{-1}(L_2 - I))\mathbf{v}}{\mathbf{v}^T (N_1)\mathbf{v}} \\ &= \frac{\mathbf{v}^T (K^T C_1^{-1} K + B_1)\mathbf{v}}{\mathbf{v}^T (K + B_1^{1/2} C_1^{1/2})^T C_1^{-1} (K + B_1^{1/2} C_1^{1/2})\mathbf{v}} \\ &= \frac{\mathbf{v}^T (K^T C_1^{-1} K + B_1)\mathbf{v}}{\mathbf{v}^T (K^T C^{-1} K + B_1 + K^T C_1^{-1/2} B_1^{1/2} + B_1^{1/2} C_1^{-1/2} K)\mathbf{v}} \end{aligned}$$

Let $\beta = C_1^{-1/2} K\mathbf{v}$ and $\gamma = B_1^{1/2}\mathbf{v}$. The above Rayleigh quotient can be denoted as

$$R = \frac{\beta^T \beta + \gamma^T \gamma}{\beta^T \beta + \gamma^T \gamma + \beta^T \gamma + \gamma^T \beta}$$

We first prove the upper bound of this Rayleigh quotient. Recall that

$$C_1 = \begin{pmatrix} \tau I & & \\ & \ddots & \\ & & \tau I \end{pmatrix}, \quad \text{and} \quad B_1 = \begin{pmatrix} \alpha_1 \tau I & & \\ & \ddots & \\ & & \alpha_1 \tau I \end{pmatrix}$$

Obviously, we have $C_1^{-1/2} B_1^{1/2} = B_1^{1/2} C_1^{-1/2} = blkdiag(a_1 I, ..., a_n I)$ for some real numbers a_i. In fact, we have

$$a_1 = ... = a_n = \sqrt{\alpha_1} \geq 0$$

Hence, according to theorem 2, we can conclude that $K^T C_1^{-1/2} B_1^{1/2} + B_1^{1/2} C_1^{-1/2} K$ is positive definite. Hence, $\beta^T\gamma + \gamma^T\beta > 0$. So

$$R = \frac{\beta^T\beta + \gamma^T\gamma}{\beta^T\beta + \gamma^T\gamma + \beta^T\gamma + \gamma^T\beta} < 1$$

Next, we can find the lower bound of R easily. Consider

$$\begin{aligned} & \frac{1}{2}\left((\beta-\gamma)^T(\beta-\gamma)\right) \geq 0 \\ \iff & \frac{1}{2}(\beta^T\beta - \gamma^T\beta - \beta^T\gamma + \gamma^T\gamma) \geq 0 \\ \iff & \beta^T\beta + \gamma^T\gamma \geq \frac{1}{2}(\beta^T\beta + \gamma^T\beta + \beta^T\gamma + \gamma^T\gamma) \\ \iff & R \geq \frac{1}{2} \end{aligned} \tag{3.15}$$

Therefore, we have $\rho(P_2^{-1}(L_2 - I)) \in [\frac{1}{2}, 1)$. And our proposed preconditioner read as

$$P_2^{-1} = N_1^{-1}T_1^{-1} = \left[(K + B_1^{1/2}C_1^{1/2})^{-1}C_1(K + B_1^{1/2}C_1^{1/2})^{-T}\right]\left[-K^T C_1^{-1}K\right] \tag{3.16}$$

3.2.3 Computational Cost of P_2^{-1}

In this section, we will describe our preconditioner P_2^{-1} in a so-called ODE form. So one can have a rough idea about the computational cost of this preconditioner. Note that applying the first part of our preconditioner $T^{-1}\mathbf{x} = -K^T C_1^{-1} K\mathbf{x}$ is approximately equivalent to the computation of the differentiation

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} + A\mathbf{x} = \mathbf{v}$$

to find $\mathbf{v}$. Then compute the differentiation

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v} - A\mathbf{v} = \mathbf{w}$$

to find $\mathbf{w}$. Next, the second part of our preconditioner $N^{-1}\mathbf{w} = (K + B_1^{1/2}C_1^{1/2})^{-1}C_1(K + B_1^{1/2}C_1^{1/2})^{-T}\mathbf{w}$ is approximately equivalent to solving the ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{z} - A\mathbf{z} + \alpha_1\mathbf{z} = \mathbf{w}$$

to find $\mathbf{z}$. And then solve one last ODE to find the output $\mathbf{q}$,

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q} + A\mathbf{q} + \alpha_1\mathbf{q} = \mathbf{z}$$

One can see that the computation our preconditioner is approximately equivalent to the backward-forward shooting, which implied that the computational cost of each iteration steps of FGMRES is roughly doubled after the preconditioner is applied. In our numerical examples, our preconditioners reduce the number of iterations more that a factor of two (see table (3.1)). Therefore, we can conclude that our preconditioners is efficient.

3.3 Preconditioner for All-time Problems with Symmetric Differential Operator

In this section, we will discuss the preconditioner for the all-time case, where the problems read as

$$\begin{aligned} J(\mathbf{y},\mathbf{u}) = J_3(\mathbf{y},\mathbf{u}) = \frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \|\mathbf{y}(t) - \hat{\mathbf{y}}(t)\|_2^2 \mathrm{d}t \\ + \frac{\alpha_2}{2}\|\mathbf{y}(T) - \hat{\mathbf{y}}(T)\|_2^2 + \frac{\alpha_3}{2}\|\mathbf{y}(0) - \hat{\mathbf{y}}(0)\|_2^2 \end{aligned} \tag{1.7}$$

subject to constraint

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) = \mathbf{u}(t) + \mathbf{f}(t), \quad t \in [0,T] \tag{1.4}$$

This type of problems arise as a sub-problem when we apply a new class of domain decomposition method on parabolic optimal control [28]. The idea is similar to the preconditioner for the tracking case. Since the matrices in all-time case do not satisfy the assumption in theorem 2, we need another way to show that the matrix $K^T\Delta + \Delta K$ is positive definite. However, it will also introduce a limitation to the proof of our eigenvalues bound. The eigenvalues bound will only hold for some α_i or some shifted time-steps in our shooting scheme. We will discuss this issue in the followings.

3.3.1 Linear System of Shooting Method

Recall our all-at-once system for the all-time problems

$$\begin{pmatrix} K^T & B \\ -C & K \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{\lambda}} \\ \tilde{\mathbf{y}} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{g}} \\ \tilde{\mathbf{f}} \end{pmatrix} \tag{2.10}$$

Following the idea in section 3.2.1, the linear system (2.35) can be written as

$$\begin{cases} L_3 - I = -K^{-1}CK^{-T}B - I \\ \mathbf{r_3} = K^{-1}CK^{-T}\tilde{\mathbf{g}} + K^{-1}\tilde{\mathbf{f}} \end{cases} \tag{3.17}$$

3.3.2 Preconditioner and Eigenvalues Analysis

In this section, we want to find a good approximation P_3 to

$$L_3 - I = -K^{-1}CK^{-T}B - I$$

and prove the corresponding eigenvalues bound. So we can apply P_3^{-1} as a preconditioner for the linear system (2.35). We want to follow the idea in the tracking case to formulate our preconditioner P_3^{-1}. We take

$$P_3^{-1} = N^{-1}T^{-1} = \left[(K + B^{1/2}C^{1/2})^{-1}C(K + B^{1/2}C^{1/2})^{-T}\right]\left[-K^TC^{-1}K\right] \tag{3.18}$$

Obviously, this preconditioner will give us the same eignevalues lower bound as P_2^{-1} (3.16). Notice that if we take $\beta = C^{-1/2}K\mathbf{v}$ and $\gamma = B^{1/2}\mathbf{v}$, we have the Rayleigh quotient

$$R = \frac{\beta^T\beta + \gamma^T\gamma}{\beta^T\beta + \gamma^T\gamma + \beta^T\gamma + \gamma^T\beta} \geq \frac{1}{2}$$

However, if we look into our matrices in the all-time case

$$C = \begin{pmatrix} (\frac{1}{\alpha_3} + \tau)I & & & \\ & \tau I & & \\ & & \ddots & \\ & & & \tau I \end{pmatrix}, \quad B = \begin{pmatrix} \alpha_1\tau I & & & \\ & \ddots & & \\ & & \alpha_1\tau I & \\ & & & (\alpha_1\tau + \alpha_2)I \end{pmatrix}$$

and take

$$C^{-1/2}B^{1/2} = B^{1/2}C^{-1/2} = \Delta = blkdiag(a_1I, ..., a_nI)$$

we have

$$\begin{cases} a_1 = \sqrt{\frac{\alpha_1\alpha_3\tau}{1+\alpha_3\tau}} \\ a_i = \sqrt{\alpha_1} \qquad\qquad \text{for} \quad i = 2, ..., n-1 \\ a_n = \sqrt{\frac{\alpha_1\tau+\alpha_2}{\tau}} \end{cases} \tag{3.19}$$

Therefore, the condition $a_1 \geq ... \geq a_n$ in theorem 2 would not be satisfy. Hence, we could not use theorem 2 to show that $K^T\Delta + \Delta K$ is positive definite and prove the upper bound. Nevertheless, we can still show this by introducing some limitation on α_i and the time-steps size. Followed from the proof in theorem 2 together with (3.19), for any vector $\tilde{\mathbf{w}} = \left(\mathbf{w}_1^T, ..., \mathbf{w}_n^T\right)^T$, we have

$$\begin{aligned}
&\tilde{\mathbf{w}}^T(\Delta K + K^T\Delta)\tilde{\mathbf{w}} \\
&= 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A\right) \mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)^T \left(a_j I\right) \left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)\right] \\
&\quad + \sum_{j=2}^{n} \left[\mathbf{w}_{j-1}^T \left(\left(a_{j-1} - a_j\right) I\right) \mathbf{w}_{j-1}\right] + \mathbf{w}_1^T \left(a_1 I\right) \mathbf{w}_1 + \mathbf{w}_n^T \left(a_n I\right) \mathbf{w}_n \\
&= 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A\right) \mathbf{w}_j\right] + \sum_{j=2}^{n-1} \left[a_j \left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)^T \left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)\right] \\
&\quad + \sum_{j=3}^{n-1} \left[(a_{j-1} - a_j)\mathbf{w}_{j-1}^T\mathbf{w}_{j-1}\right] + (2a_1 - a_2)\mathbf{w}_1^T\mathbf{w}_1 + a_{n-1}\mathbf{w}_{n-1}^T\mathbf{w}_{n-1} - 2a_n\mathbf{w}_{n-1}^T\mathbf{w}_n + 2a_n\mathbf{w}_n^T\mathbf{w}_n \\
&\geq (2a_1 - a_2)\mathbf{w}_1^T\mathbf{w}_1 + a_{n-1}\mathbf{w}_{n-1}^T\mathbf{w}_{n-1} - 2a_n\mathbf{w}_{n-1}^T\mathbf{w}_n + 2a_n\mathbf{w}_n^T\mathbf{w}_n \\
&= (2a_1 - a_2)\mathbf{w}_1^T\mathbf{w}_1 + a_{n-1}\left(\mathbf{w}_{n-1} - \frac{a_n}{a_{n-1}}\mathbf{w}_n\right)^T\left(\mathbf{w}_{n-1} - \frac{a_n}{a_{n-1}}\mathbf{w}_n\right) + \left(2a_n - \frac{a_n^2}{a_{n-1}}\right)\mathbf{w}_n^T\mathbf{w}_n \\
&\geq (2a_1 - a_2)\mathbf{w}_1^T\mathbf{w}_1 + \left(2a_n - \frac{a_n^2}{a_{n-1}}\right)\mathbf{w}_n^T\mathbf{w}_n
\end{aligned}$$

Hence, $K^T\Delta + \Delta K$ is positive definite if the following inequalities hold

$$\begin{cases} 2a_1 \geq a_2 \\ 2a_{n-1} \geq a_n \end{cases} \tag{3.20}$$

If we set $\alpha_3 \geq \frac{1}{3\tau}$, then

$$\begin{aligned} & \alpha_3 \geq \frac{1}{3\tau} \\ \iff & 4\alpha_3\tau \geq 1 + \alpha_3\tau \\ \iff & 4 \geq \frac{1+\alpha_3\tau}{\alpha_3\tau} \\ \iff & 4\left(\frac{\alpha_1\alpha_3\tau}{1+\alpha_3\tau}\right) \geq \alpha_1 \\ \iff & 2a_1 \geq a_2 \end{aligned} \tag{3.21}$$

Also notice that

$$\begin{aligned} & \frac{\alpha_1}{\alpha_2} \geq \frac{1}{3\tau} \\ \iff & 4\alpha_1\tau \geq \alpha_1\tau + \alpha_2 \\ \iff & 2\sqrt{\alpha_1} \geq \sqrt{\frac{\alpha_1\tau+\alpha_2}{\tau}} \\ \iff & 2a_{n-1} \geq a_n \end{aligned} \tag{3.22}$$

Combining (3.21) and (3.22), we can deduce that if

$$\begin{cases} \alpha_3 \geq \frac{1}{3\tau} \\ \frac{\alpha_1}{\alpha_2} \geq \frac{1}{3\tau} \end{cases} \tag{3.23}$$

then the matrix $K^T\Delta + \Delta K$ is positive definite and $\beta^T\gamma + \gamma^T\beta > 0$. Hence, we have the upper bound

$$R = \frac{\beta^T\beta + \gamma^T\gamma}{\beta^T\beta + \gamma^T\gamma + \beta^T\gamma + \gamma^T\beta} < 1$$

3.3.3 Restriction of Parameters

Our proof for eigenvalues bound $[\frac{1}{2}, 1)$ only valid when the inequalities in (3.23) hold. There are two ways to make sure (3.23) hold. Firstly, if we are provided with a fixed time-steps size τ. We can simply set α_i, $i = 1, 2, 3$ such that our inequalities (3.23) are satisfied. However, we may not be able to chose α_i freely in many real-world applications. Therefore, alternatively, we can use a so-called shifted time-steps size

τ_1 and τ_2 at the initial and terminal time-steps. With given α_i, we choose τ_1 and τ_2 such that

$$\begin{cases} \alpha_3 \geq \frac{1}{3\tau_1} \\ \frac{\alpha_1}{\alpha_2} \geq \frac{1}{3\tau_2} \end{cases}$$

In other words, we need a large τ_1 and τ_2. With the above restrictions, our eigenvalues bound $[\frac{1}{2}, 1)$ can be guaranteed. Nevertheless, our numerical examples in next section showed that the FGMRES still converge nicely even the above restrictions are violated.

On the other hand, one can observe that the computational cost of P_3^{-1} is basically the same as that of P_2^{-1}. We would not describe the computational cost of P_3^{-1} explicitly here.

3.4 Numerical Result

3.4.1 Target States Problems

In this section, we will test the performance of our preconditioner

$$P_1^{-1} = -\left(\frac{\alpha_2}{2}I + A\right)^{-1} A - \left(\frac{1}{\alpha_2 T + \frac{\alpha_2}{\alpha_3} + 1}\right) I \tag{3.6}$$

on the linear system $(L_1 - I)\mathbf{y}^{(n)} = -\mathbf{r_1}$ (2.33) obtained by shooting method. Consider the cost functional for target states problems

$$\min_{y,u} J(y,u) = \frac{1}{2}\int_0^T \int_\Omega (u(x,t))^2 \mathrm{d}x\mathrm{d}t + \frac{\alpha_2}{2}\int_\Omega (y(x,T) - \hat{y}(x,T))^2 \mathrm{d}x + \frac{\alpha_3}{2}\int_\Omega (y(x,0) - \hat{y}(x,0))^2 \mathrm{d}x$$

subject to a two-dimensional diffusion equation with Neumann boundary condition

$$\begin{cases} \frac{\partial}{\partial t} y(x,t) + \nabla \cdot (a(x)\nabla y(x,t)) = u(x,t) + f(x,t) & \text{in} [0,T] \times \Omega \\ \frac{\partial y}{\partial n} = 0 & \text{in} [0,T] \times \partial\Omega \end{cases}$$

Here, we obtain the discrete partial differential operator A by applying finite element method with triangular mesh shown in figure (3.2), which is constructed by dividing the spacial domain $\Omega = (x_1, x_2)$ into n_x^2 triangular elements. And we apply Euler's method in time using n_t time steps. In our numerical experiment, we set $\Omega = [0,1]^2$,

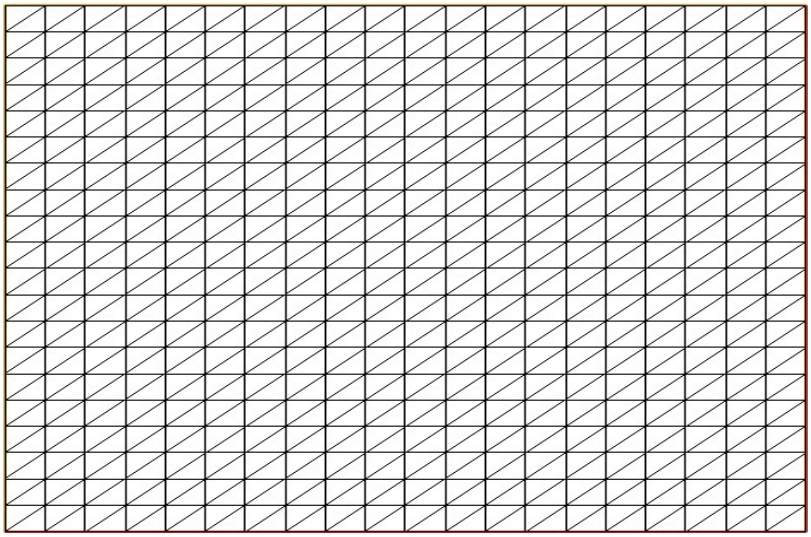


Figure 3.2: Triangular mesh used in the finite element discretization.

$T = 0.2$, $n_x = 3721$, $n_t = 32$, $\tau = 0.00625$, $f = 0$, $\hat{y}(x,t) = e^t \sin(x_1)\cos(x_2)$, and $a(x) = 2 + \sin(10\pi x_1) + \sin(10\pi x_2)$. FGMRES, which is truncated when the residual norm is less than 10^{-7}, is applied to show the performance of our preconditioner with different α_2 and α_3. The performances of our preconditioner (3.6) to the this testing example is showed in figure (3.3) and table (3.1).

3.4.2 Tracking Problems

We are going to test the performance of our preconditioner in this section

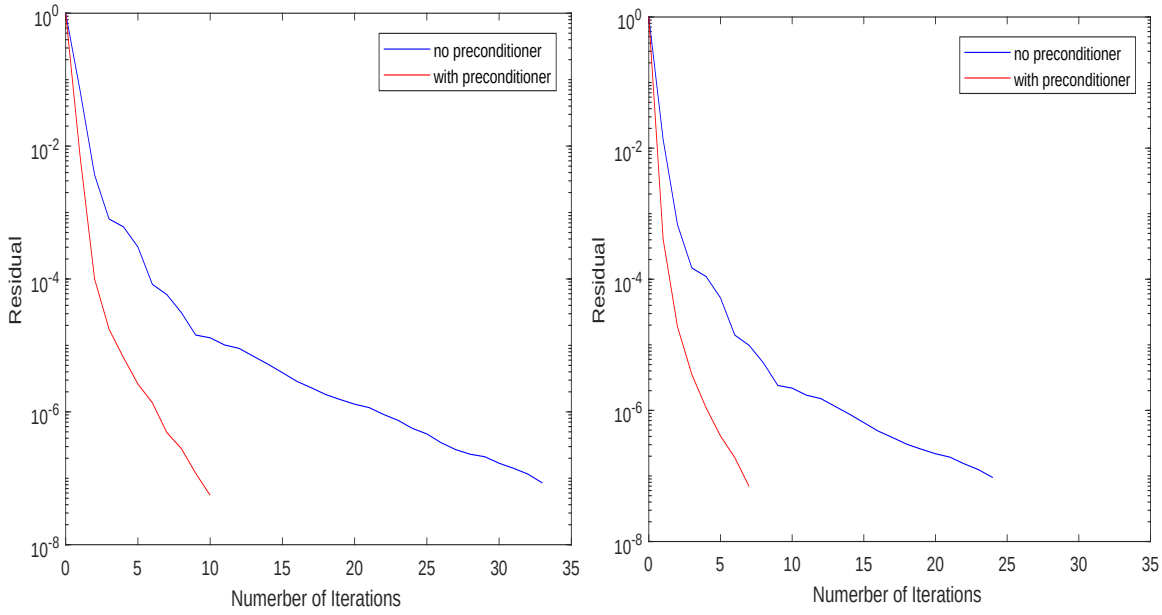
$$P_2^{-1} = N^{-1}T^{-1} = \left[(K + B_1^{1/2}C_1^{1/2})^{-1}C_1(K + B_1^{1/2}C_1^{1/2})^{-T}\right]\left[-K^T C_1^{-1}K\right] \quad (3.16)$$

This preconditioner is applied onto the linear system $(L_2 - I)\tilde{\mathbf{y}} = -\mathbf{r_2}$. Consider a distributed tracking control of two-dimensional heat equation as an example.

$$\min_{y,u} J(y,u) = \frac{1}{2}\int_0^T \int_\Omega (u(x,t))^2 \mathrm{d}x\mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \int_\Omega (y(x,t) - \hat{y}(x,t))^2 \mathrm{d}x\mathrm{d}t$$

subjected to the two-dimensional heat equation with Dirichlet boundary condition

$$\begin{cases} \frac{\partial}{\partial t}y(x,t) - c\nabla^2 y(x,t) = u(x,t) + f(x,t) & \text{in } [0,T]\times\Omega \\ y(x,t) = 0 & \text{in } [0,T]\times\partial\Omega \\ y(x,0) = y_0 & \text{in } \Omega \end{cases}$$



(a) $\alpha_2 = 10^5$, $\alpha_3 = 10^5$ (b) $\alpha_2 = 10^5$, $\alpha_3 = 1$

Figure 3.3: Performances of preconditioner P_1^{-1} (3.6) on the linear system (2.33).

Our spatial discretization, i.e., the formulation of our discrete partial differential operator A, use finite difference method on a uniform grid which is constructed by dividing $\Omega = (x_1, x_2)$ into n_x^2 squares of equal size. And we apply Euler's method in time using n_t time steps. In our computation, we set $\Omega = [0,1]^2$, $T = 0.2$, $c = 1/2\pi^2$, $n_x = 16$, $n_t = 32$, $\tau = 0.00625$, $f(x,t) = 0$, $y_0 = 0$ and $\hat{y}(x,t) = \min(x_1, x_2, 1-x_1, 1-x_2)$. So our problems is similar to the second example in [23]. Again, FGMRES with residual norm toleration less than 10^{-7} is applied. Figure (3.4) and table (3.1) showed the performance of P_2^{-1} (3.16) on the linear system (2.34) with this test example.

3.4.3 All-time Problems

We test our preconditioner

$$P_3^{-1} = N^{-1}T^{-1} = \left[(K + B^{1/2}C^{1/2})^{-1}C(K + B^{1/2}C^{1/2})^{-T}\right]\left[-K^T C^{-1}K\right] \qquad (3.18)$$

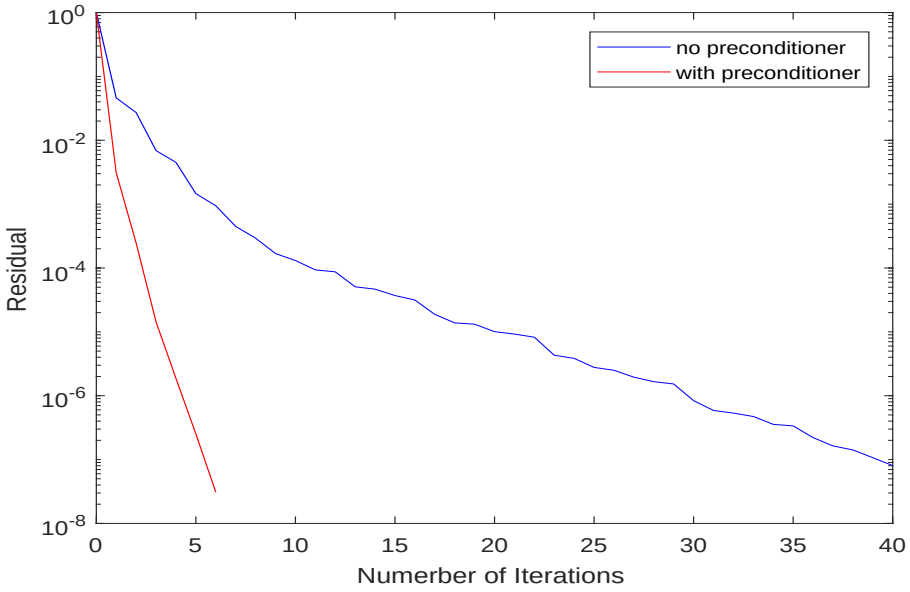


Figure 3.4: Performances of preconditioner P_2^{-1} (3.16) on the linear system (2.34) for $\alpha_1 = 10^5$

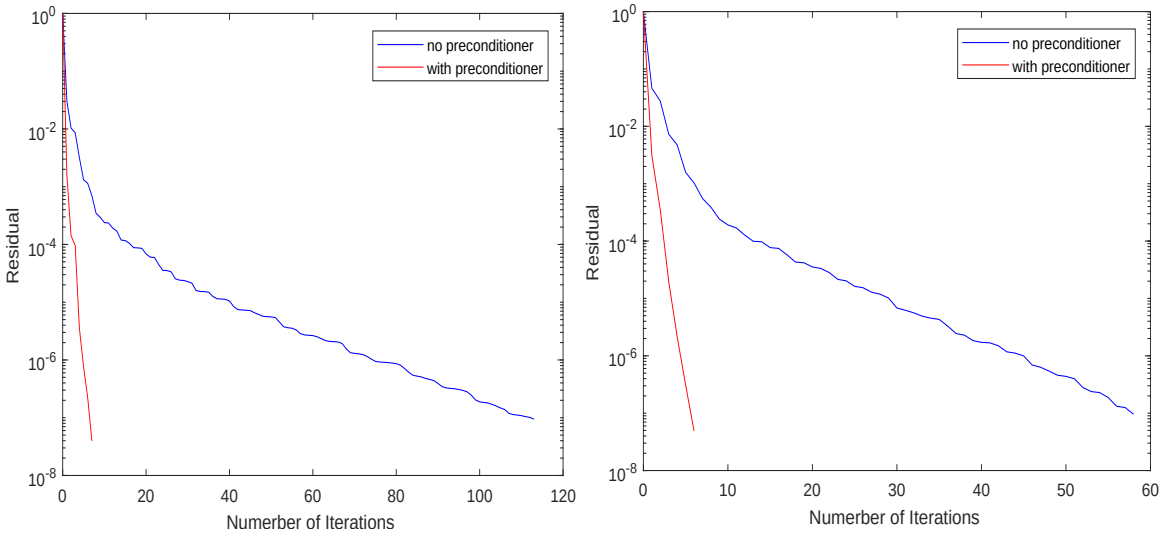
We will use the two-dimensional heat equation as an testing example again. Consider the all-time cost functional

$$\min_{y,u} J(y,u) = \frac{1}{2}\int_0^T \int_\Omega (u(x,t))^2 \mathrm{d}x\mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \int_\Omega (y(x,t) - \hat{y}(x,t))^2 \mathrm{d}x\mathrm{d}t + \frac{\alpha_2}{2}\int_\Omega (y(x,T) - \hat{y}(x,T))^2 \mathrm{d}x + \frac{\alpha_3}{2}\int_\Omega (y(x,0) - \hat{y}(x,0))^2 \mathrm{d}x$$

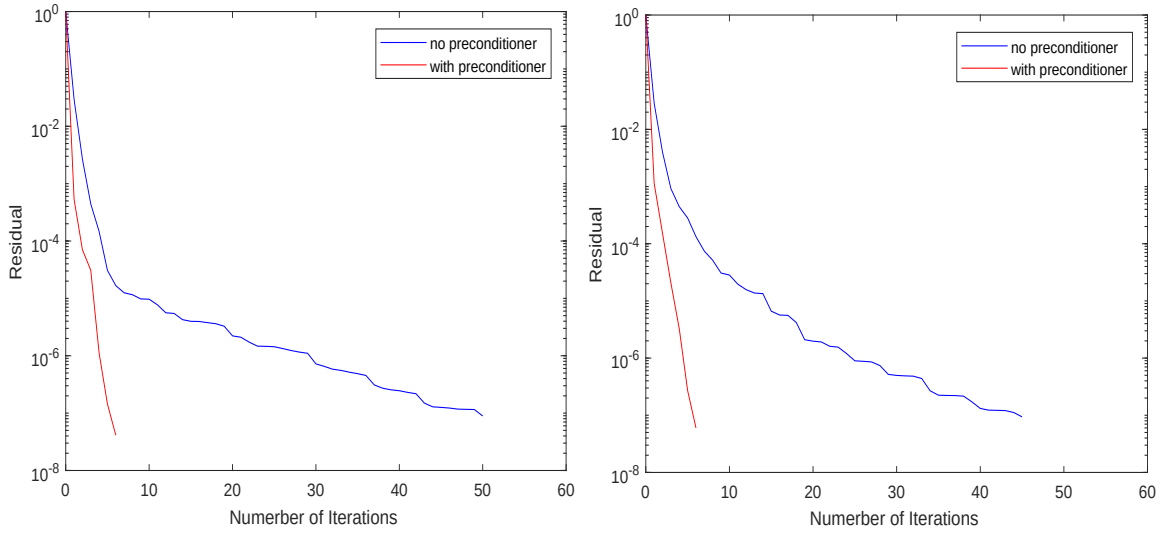
subjected to the two-dimensional heat equation with Dirichlet boundary condition

$$\begin{cases} \frac{\partial}{\partial t} y(x,t) - c\nabla^2 y(x,t) = u(x,t) + f(x,t) & \text{in } [0,T] \times \Omega \\ y(x,t) = 0 & \text{in } [0,T] \times \partial\Omega \end{cases}$$

We will use the same setting as in section 3.4.2. We test the performance of the preconditioner with different α_1, α_2, and α_3. Figure (3.5) and table (3.1) show the numerical result of our preconditioner P_3^{-1} apply on the linear system (2.35).



(a) $\alpha_1 = 10^5$, $\alpha_2 = 10^5$, $\alpha_3 = 10^5$ (b) $\alpha_1 = 10^5$, $\alpha_2 = 1$, $\alpha_3 = 10^5$



(c) $\alpha_1 = 10^5$, $\alpha_2 = 10^5$, $\alpha_3 = 1$ (d) $\alpha_1 = 10^5$, $\alpha_2 = 1$, $\alpha_3 = 1$

Figure 3.5: Performances of preconditioner P_3^{-1} (3.18) on the linear system (2.35).

Symmetric A	α_1	α_2	α_3	No Preconditioner	With Preconditioner
Target States	N/A	10^5	10^5	33	10
Target States	N/A	1	10^5	3	3
Target States	N/A	10^5	1	24	7
Tracking	10^5	N/A	N/A	40	6
Tracking	1	N/A	N/A	3	5
All-time	10^5	10^5	10^5	113	7
All-time	10^5	1	10^5	58	6
All-time	10^5	10^5	1	50	6
All-time	10^5	1	1	45	6

Table 3.1: Number of iterations for FGMRES for symmetric A with different parameters.

Chapter 4

Preconditioners for Parabolic Optimal Control Problems with Non-symmetric Differential Operator

As mentioned previously, our matrix A, which is arose from the space-discretization of parabolic PDE, is not always symmetric. The preconditioners of last chapter are designed for the symmetric case. In this chapter, we will move on to a more challenging case, which is when A is not symmetric. The structure in this chapter is similar to the last chapter. In section 4.1, we will introduce a preconditioner for the target states problems with non-symmetric A. Next, in section 4.2, a preconditioner for the tracking problems and a preconditioner for the all-time problems are presented. Lastly, we will show the numerical performance in 4.3

4.1 Preconditioner for Target States Problems with Non-symmetric Differntial Operator

Recall our target states optimal control problem defined in section 1.4.1.

$$J(\mathbf{y},\mathbf{u}) = J_1(\mathbf{y},\mathbf{u}) = \frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t + \frac{\alpha_2}{2}\|\mathbf{y}(T)-\hat{\mathbf{y}}(T)\|_2^2 + \frac{\alpha_3}{2}\|\mathbf{y}(0)-\hat{\mathbf{y}}(0)\|_2^2 \quad (1.5)$$

subject to a system of ODE obtained by applying spatial discretization on a parabolic PDE

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) = \mathbf{u}(t) + \mathbf{f}(t), \quad t \in [0, T] \tag{1.4}$$

In this section, we will consider A as a non-symmetric matrix, i.e., $A^T \neq A$. Our goal is to solve the corresponding linear system

$$(L_1 - I)\,\mathbf{y}^{(n)} = -\mathbf{r_1} \tag{2.33}$$

with a preconditioner P_4^{-1}. Similar to the symmetric case, the main idea is to integrate $\mathbf{y}^{(n)}$ backward and then forward by solving the *backward adjoint equation* and the *forward state equation* without time-discretization. However, given that A is non-symmetric, it is hard to formulate the approximation of high and low frequency this time. Instead, we proved that the inverse of $L_1 - I$ in (2.33) satisfy a matrix equation known as *Continous Algebraic Riccati Equation* (CARE). Hence, solving this matrix equation CARE will give a preconditioner to our linear system. Note that the numerical methods for solving such matrix equation is still an active research field. Interested readers could visit [3] for more numerical methods. Of course, this preconditioner will also be applicable to the symmetric case. Nevertheless, this would not be a good choice for symmetric case as we will see that the preconditioner in this chapter would have a higher computation cost compared to P_1^{-1} (3.6).

4.1.1 Derivation of the Preconditioner

Before we start our derivation, we assume that $\hat{\mathbf{y}}(t) = \mathbf{f}(t) = 0$ in our derivation for the sake of simplicity, as we already demonstrated in section 3.1 that these terms will fall into the right-hand-side vector instead of left-hand-side matrix in our linear system (2.33). We will follow the same idea in section 3.1 to derive our desired preconditioner. We first let $A = QDQ^{-1}$ (so $A^T = Q^{-T}DQ^T$) and $\boldsymbol{\mu} = Q^T\boldsymbol{\lambda}$, then the *backward adjoint equation* (2.14) can be written as

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\mu}(t) - D\boldsymbol{\mu}(t) = 0 \\ \boldsymbol{\mu}(T) = -\alpha_2 Q^T\mathbf{y}(T) \end{cases}$$

So we have n-copies of ODE,

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\mu}_i - d_i\boldsymbol{\mu}_i = 0 \\ \boldsymbol{\mu}_i(T) = -\alpha_2 \left[Q^T\mathbf{y}(T)\right]_i \end{cases}$$

where d_i is the i-th eigenvalue of A. Note that again our notation $[x]_i$ represent the i-th element in vector x. Now, we can solve the ODE by integrating factor method and obtain

$$\boldsymbol{\mu}_i = -\alpha_2 e^{d_i(t-T)} \left[Q^T\mathbf{y}(T)\right]_i$$

Therefore, we obtained the solution of the *backward adjoint equation*,

$$\boldsymbol{\lambda(t)} = -\alpha_2 e^{A^T(t-T)}\mathbf{y}(T)$$

Now, let $\boldsymbol{\xi} = Q^{-1}\mathbf{y}$, the *forward state equation* (2.15) would become

$$\begin{cases} \frac{\partial}{\partial t}\boldsymbol{\xi} + D\boldsymbol{\xi} = Q^{-1}\boldsymbol{\lambda} \\ \boldsymbol{\xi}(0) = \frac{1}{\alpha_3}Q^{-1}\boldsymbol{\lambda}(0) \end{cases}$$

Similarly, we have n-copies of ODE,

$$\begin{cases} \frac{\partial}{\partial t}\boldsymbol{\xi}_i + d_i\boldsymbol{\xi}_i = [Q^{-1}\boldsymbol{\lambda}(t)]_i \\ \boldsymbol{\xi}_i(0) = \left[\frac{1}{\alpha_3}Q^{-1}\boldsymbol{\lambda}(0)\right]_i \end{cases}$$

By integrating factor method we can obtain

$$e^{d_i t}\boldsymbol{\xi}_i = \int_0^t e^{d_i s}\left[Q^{-1}\boldsymbol{\lambda}(s)\right]_i \mathrm{d}s + C_i$$

With the initial condition, we can find C_i and hence we have

$$\boldsymbol{\xi}_i = \int_0^t e^{d_i(s-t)}\left[Q^{-1}\boldsymbol{\lambda}(s)\right]_i \mathrm{d}s + e^{-d_i t}\left[\frac{1}{\alpha_3}Q^{-1}\boldsymbol{\lambda}(0)\right]_i$$

So we have the solution of the *forward state equation*

$$\mathbf{y}(t) = \int_0^t e^{A(s-t)}\boldsymbol{\lambda}(s)\mathrm{d}s + \frac{1}{\alpha_3}e^{-At}\boldsymbol{\lambda}(0)$$

Since we have the solution of $\boldsymbol{\lambda}(t)$, we obtained

$$\mathbf{y}(t) = \left[-\alpha_2\int_0^t e^{A(s-t)}e^{A^T(s-T)}\mathrm{d}s - \frac{\alpha_3}{\alpha_3}e^{-tA}e^{-TA^T}\right]\mathbf{y}(T)$$

Therefore, by putting $t = T$, we formulated the linear mapping $\mathcal{R}_1$ in section 2.2. Hence

$$L_1 - I = -\alpha_2 \int_0^T e^{A(s-T)} e^{A^T(s-T)} \mathrm{d}s - \frac{\alpha_2}{\alpha_3} e^{-TA} e^{-TA^T} - I$$

Since our shooting matrix $L_1 - I$ for non-symmetric case consist of an integral involving matrix exponential, we aim to find a way to compute it now. Consider

$$\begin{aligned}
M &= \int_0^T e^{As} e^{A^T s} \mathrm{d}s \\
&= \frac{1}{2\gamma} \int_0^T e^{(A-\gamma I)s} e^{(A^T-\gamma I)s} \mathrm{d}\left(e^{2\gamma s}\right) \\
&= \frac{1}{2\gamma} \left(e^{TA} e^{TA^T} - I\right) \\
&\quad - \frac{1}{2\gamma} \int_0^T e^{2\gamma s} \left[(A-\gamma I)\, e^{(A-\gamma I)s} e^{(A^T-\gamma I)s} + e^{(A-\gamma I)s} e^{(A^T-\gamma I)s} \left(A^T - \gamma I\right)\right] \mathrm{d}s \\
&= \frac{1}{2\gamma} \left[e^{TA} e^{TA^T} - I - (A-\gamma I)M - M(A^T - \gamma I)\right]
\end{aligned}$$

So we obtained a matrix equation,

$$AM + MA^T = e^{TA} e^{TA^T} - I$$

Notice that we have $L_1 - I = -\alpha_2 e^{-TA} M e^{-TA^T} - \frac{\alpha_2}{\alpha_3} e^{-TA} e^{-TA^T} - I$, we get

$$A(L_1 - I) + (L_1 - I)A^T = -Q \tag{4.1}$$

where

$$Q = A + A^T + \alpha_2 I + e^{-TA} \left[\frac{\alpha_2}{\alpha_3} A + \frac{\alpha_2}{\alpha_3} A^T - \alpha_2 I\right] e^{-TA^T}$$

Therefore, our preconditioner $P_4 \approx (L-I)^{-1}$ satisfy the *Continuous Algebraic Riccati Equation (CARE)*:

$$A^T X + XA + XQX = 0 \tag{4.2}$$

4.1.2 Computation of the Preconditioner

Schur decomposition method is one of the classical algorithm to solve CARE. There are many other algorithms in the literature. Numerical methods for solving such matrix equation is an active research topic. As our focus is on the preconditioner

for shooting method, we will just use a classical way to solve our CARE. One of our future direction is to find the best numerical method to solve the CARE in our case. However, as of this moment, we will just employ the Schur decomposition method to see the convergence performance of our Riccati preconditioner P_4^{-1} (4.2). A MALAB algorithm, found in [3], for Schur decomposition method is given here.

Algorithm 3: MATLAB code to Solve CARE: $C + XA + A^TX - XBX = 0$, and apply X as a preconditioner in GMRES

Input : A, B, C, v

Output: z

```
1  n=size(B,1);
2  H=[A,-B;-C,-A'];
3  [U,T]=schur(H,'real') ;
4  e=ordeig(T);
5  [es,is]=sort(real(e),'ascend');
6  sel=zeros(2*n,1);
7  sel(is(1:n));
8  Q=ordschur(U,T,sel);
9  X=Q(n+1:2*n,1:n)/Q(1:n,1:n);
10 z=Xv;
```

We can see that the CARE (4.2) is fairly expensive to solve. Moreover, the matrix Q in (4.2) involves the computation of matrix exponential. Similar to the CARE, the computation of matrix exponential itself is still an active research topic [50][41][24]. A detailed introduction can be found on the book [25]. Since the computation of our preconditioner P_4^{-1} (4.2) is quite expensive, we suggest that one could go to the coarse grid to solve it. Therefore, the two-grid methods or multi-grid methods would be considered as one of our future directions.

4.2 Preconditioner for Tracking Problems and All-time Problems with Non-symmetric Differntial Operator

We are going to present a preconditioner for the tracking problems and all-time problems in this section. Recall our cost functional for tracking problems,

$$J(\mathbf{y},\mathbf{u}) = J_2(\mathbf{y},\mathbf{u}) = \frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \|\mathbf{y}(t)-\hat{\mathbf{y}}(t)\|_2^2 \mathrm{d}t \tag{1.6}$$

Also recall our cost functional for all-time problems,

$$\begin{aligned} J(\mathbf{y},\mathbf{u}) = J_2(\mathbf{y},\mathbf{u}) = &\frac{1}{2}\int_0^T \|\mathbf{u}(t)\|_2^2 \mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \|\mathbf{y}(t)-\hat{\mathbf{y}}(t)\|_2^2 \mathrm{d}t \\ &+ \frac{\alpha_2}{2}\|\mathbf{y}(T)-\hat{\mathbf{y}}(T)\|_2^2 + \frac{\alpha_3}{2}\|\mathbf{y}(0)-\hat{\mathbf{y}}(0)\|_2^2 \end{aligned} \tag{1.7}$$

Both cost functionals are subject to a system of ODE arose from the space-discretization of parabolic PDE

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y}(t) + A\mathbf{y}(t) = \mathbf{u}(t) + \mathbf{f}(t), \quad t \in [0,T] \tag{1.4}$$

In this section, we take $A^T \neq A$. We are going to developed a preconditioner P_5^{-1} for tracking problems and P_6^{-1} for all-time problems. The idea of our preconditioner P_5^{-1} and P_6^{-1} is very similar to that of symmetric case. Indeed, we would use the preconditioner with the same structure as the symmetric case

$$P_5^{-1} = N_1^{-1}T_1^{-1} = \left[(K + B_1^{1/2}C_1^{1/2})^{-1}C_1(K + B_1^{1/2}C_1^{1/2})^{-T}\right]\left[-K^T C_1^{-1}K\right] \tag{4.3}$$

and

$$P_6^{-1} = N^{-1}T^{-1} = \left[(K + B^{1/2}C^{1/2})^{-1}C(K + B^{1/2}C^{1/2})^{-T}\right]\left[-K^T C^{-1}K\right] \tag{4.4}$$

except, of course, we have $A \neq A^T$ here.

4.2.1 Preconditioner and Engienvalues Analysis

To show that our preconditioner P_5^{-1} and P_6^{-1} work for the non-symmetric case, we need to show that theorem 2 also holds for $A \neq A^T$. In fact, one can observe that theorem 2 is a special case for the following theorem 3:

Theorem 3. *Let A be a non-symmetric matrix and K is the matrix defined in (2.2). Suppose $\Delta = blkdiag(a_1 I, ..., a_n I)$ where $a_i > 0$ for all $i = 1, ..., n$. Also assume that $a_1 \geq ... \geq a_n$. Then $\Delta K + K^T \Delta$ is positive definite.*

Proof. Let $\tilde{\mathbf{w}} := (\mathbf{w}_1^T, ..., \mathbf{w}_n^T)^T$ be any real vector. We have

$$
\begin{aligned}
&\tilde{\mathbf{w}}^T(\Delta K + K^T\Delta)\tilde{\mathbf{w}} \\
&= \begin{pmatrix} \mathbf{w}_1^T & \dots & \mathbf{w}_n^T \end{pmatrix}
\begin{pmatrix}
a_1(\tau A + \tau A^T + 2I) & -a_2 I & & \\
-a_2 I & \ddots & \ddots & \\
& \ddots & \ddots & -a_n I \\
& & -a_n I & a_n(\tau A + \tau A^T + 2I)
\end{pmatrix}
\begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_n \end{pmatrix} \\
&= \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j(\tau A + \tau A^T + 2I)\right) \mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\mathbf{w}_{j-1}^T(-a_j I)\mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\mathbf{w}_j^T(-a_j I)\mathbf{w}_{j-1}\right] \\
&= 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A\right) \mathbf{w}_j\right] + 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A^T\right) \mathbf{w}_j\right] \\
&\quad + 2\sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j I\right) \mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\mathbf{w}_{j-1}^T(-a_j I)\mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\mathbf{w}_j^T(-a_j I)\mathbf{w}_{j-1}\right] \\
&= 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A\right) \mathbf{w}_j\right] + 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A^T\right) \mathbf{w}_j\right] \\
&\quad + \sum_{j=2}^{n} \left[\mathbf{w}_{j-1}^T \left(a_{j-1} I\right) \mathbf{w}_{j-1} - \mathbf{w}_{j-1} \left(a_j I\right) \mathbf{w}_j - \mathbf{w}_j^T \left(a_j I\right) \mathbf{w}_{j-1} + \mathbf{w}_j^T \left(a_j I\right) \mathbf{w}_j\right] \\
&\quad + \mathbf{w}_1^T \left(a_1 I\right) \mathbf{w}_1 + \mathbf{w}_n^T \left(a_n I\right) \mathbf{w}_n \\
&= 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A\right) \mathbf{w}_j\right] + 2\tau \sum_{j=1}^{n} \left[\mathbf{w}_j^T \left(a_j A^T\right) \mathbf{w}_j\right] + \sum_{j=2}^{n} \left[\left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)^T \left(a_j I\right) \left(\mathbf{w}_{j-1} - \mathbf{w}_j\right)\right] \\
&\quad + \sum_{j=2}^{n} \left[\mathbf{w}_{j-1}^T \left(\left(a_{j-1} - a_j\right) I\right) \mathbf{w}_{j-1}\right] + \mathbf{w}_1^T \left(a_1 I\right) \mathbf{w}_1 + \mathbf{w}_n^T \left(a_n I\right) \mathbf{w}_n \\
&\geq 0
\end{aligned}
$$

With the same argument in the proof of theorem (2), we can prove that the equality sign only holds when $\tilde{\mathbf{w}} = 0$. Therefore, the matrix $K\Delta + \Delta K^T$ is positive definite. □

Hence, with this theorem 3, we can see that the prove in section 3.2 and 3.3 also hold for the non-symmetric case. Hence, both $\rho(P_5^{-1}(L_2 - I))$ and $\rho(P_6^{-1}(L_3 - I))$ are bound by $[\frac{1}{2}, 1)$ for the non-symmetric case. Note that the bound for all-time problems is valid only when the inequalities (3.23) hold.

4.3 Numerical Result

4.3.1 Target States Problems

In this section, we will test the performance of our preconditioner P_4^{-1} on the linear system $(L_1 - I)\mathbf{y}^{(n)} = -\mathbf{r_1}$ (2.33) obtained by shooting method. Consider the cost functional for non-tracking problems

$$\min_{y,u} J(y,u) = \frac{1}{2}\int_0^T \int_\Omega (u(x,t))^2 \mathrm{d}x\mathrm{d}t + \frac{\alpha_2}{2}\int_\Omega (y(x,T) - \hat{y}(x,T))^2 \mathrm{d}x + \frac{\alpha_3}{2}\int_\Omega (y(x,0) - \hat{y}(x,0))^2 \mathrm{d}x$$

subject to a two-dimensional advection-diffusion equation

$$\begin{cases} \frac{\partial}{\partial t} y(x,t) - \nabla \cdot (\nu \nabla y(x,t) + \mathbf{b} y(x,t)) = u(x,t) & \text{in} [0,T] \times \Omega \end{cases}$$

where $\mathbf{b} = (b_1, b_2)$ is a vector field satisfying the Stokes system

$$\begin{cases} \Delta b_1 + p_{x_1} = 0 \\ \Delta b_2 + p_{x_2} = 0 \\ \nabla \cdot \mathbf{b} = 0 \end{cases}$$

Our problems data are those of the final example in [28]. The space domain is a region defined by the x_1-axis at the bottom and the curve $x_2 = 1 - x_1^2$ at the top. Figure (4.1) demonstrated our domain Ω with the vector field $\mathbf{b}$. In our numerical experiment, we set $\nu = 0.1$, $T = 32$, $n_t = 128$, and $\tau = 0.25$. The advection-diffusion equation is discretized by an upwind finite volume scheme. The boundary condition is Neumann on the curved boundary, Dirichlet with y=0 on the negative x_1-axis and Neumann on the positive x_1-axis. Once again, FGMRES, which is truncated when the residual norm is less than 10^{-7}, is applied to show the performance of our preconditioner with different α_2 and α_3. The performances of our preconditioner P_4^{-1} (4.2) to the this testing example is showed in figure (4.2) and table (4.1).

4.3.2 Tracking Problems

We are going to test the performance of our preconditioner P_5^{-1} in this section

$$P_5^{-1} = N_1^{-1} T_1^{-1} = \left[(K + B_1^{1/2} C_1^{1/2})^{-1} C_1 (K + B_1^{1/2} C_1^{1/2})^{-T}\right] \left[-K^T C_1^{-1} K\right] \qquad (4.3)$$

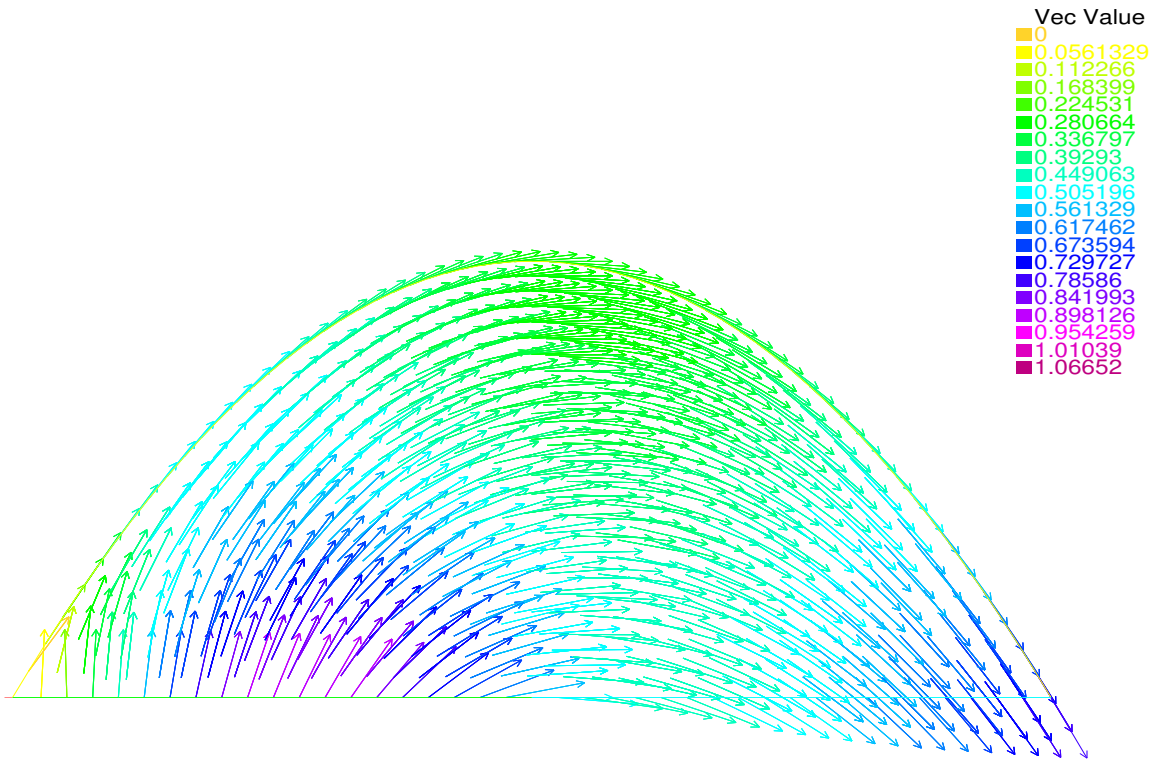
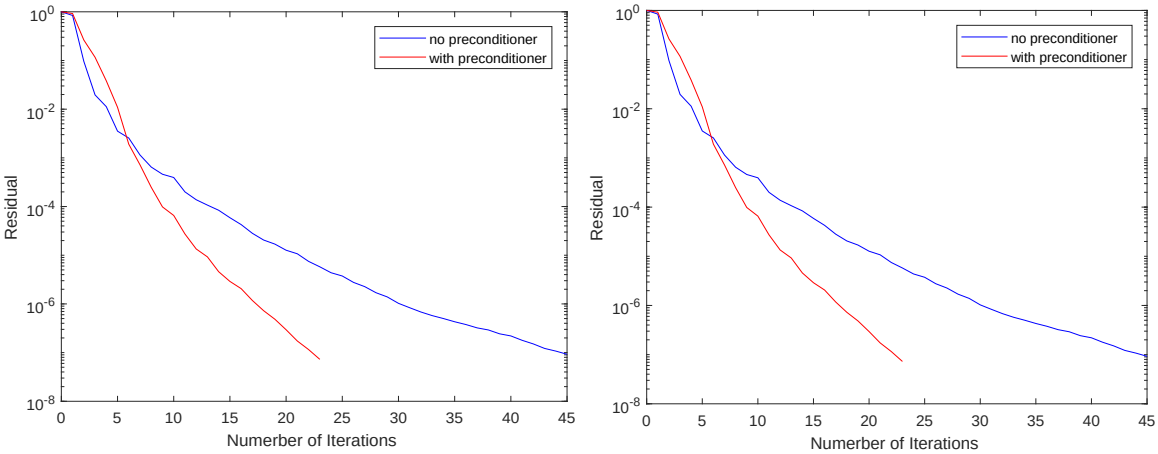


Figure 4.1: A visualization of domain Ω with vector field $\mathbf{b}$.



(a) $\alpha_2 = 10^5$, $\alpha_3 = 10^5$ (b) $\alpha_2 = 10^5$, $\alpha_3 = 1$

Figure 4.2: Performances of preconditioner P_4^{-1} (4.2) on the linear system (2.33).

This preconditioner is applied to the linear system $(L_2 - I)\tilde{\mathbf{y}} = -\mathbf{r_2}$. Consider again a distributed control of two-dimensional advection-diffusion equation as an example. The cost functional of tracking case read as

$$\min_{y,u} J(y,u) = \frac{1}{2}\int_0^T \int_\Omega (u(x,t))^2 \mathrm{d}x\mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \int_\Omega (y(x,t) - \hat{y}(x,t))^2 \mathrm{d}x\mathrm{d}t$$

This cost functional is subjected to a two-dimensional advection-diffusion equation

$$\begin{cases} \frac{\partial}{\partial t} y(x,t) - \nabla \cdot (\nu \nabla y(x,t) + \mathbf{b} y(x,t)) = u(x,t) & \text{in} \quad [0,T] \times \Omega \\ y(x,0) = y_0 & \text{in} \quad [0,T] \times \Omega \end{cases}$$

where $\mathbf{b} = (b_1, b_2)$ is a vector field satisfying the Stokes system

$$\begin{cases} \Delta b_1 + p_{x_1} = 0 \\ \Delta b_2 + p_{x_2} = 0 \\ \nabla \cdot \mathbf{b} = 0 \end{cases}$$

In this numerical experiment, we set $T = 0.2$, $n_t = 16$, $\tau = 0.0125$, and other settings are as same as section 4.3.1. Again, FGMRES with residual norm toleration less than 10^{-7} is applied. Figure (4.3) and table (4.1) showed the performance of P_5^{-1} on the linear system (2.34) with this test example.

4.3.3 All-time Problems

We test the performance of the preconditioner.

$$P_6^{-1} = N^{-1}T^{-1} = \left[(K + B^{1/2}C^{1/2})^{-1} C (K + B^{1/2}C^{1/2})^{-T}\right]\left[-K^T C^{-1} K\right] \qquad (4.4)$$

We consider the following cost functional of all-time case

$$\begin{aligned} \min_{y,u} J(y,u) = &\frac{1}{2}\int_0^T \int_\Omega (u(x,t))^2 \mathrm{d}x\mathrm{d}t + \frac{\alpha_1}{2}\int_0^T \int_\Omega (y(x,t) - \hat{y}(x,t))^2 \mathrm{d}x\mathrm{d}t \\ &+ \frac{\alpha_2}{2}\int_\Omega (y(x,T) - \hat{y}(x,T))^2 \mathrm{d}x + \frac{\alpha_3}{2}\int_\Omega (y(x,0) - \hat{y}(x,0))^2 \mathrm{d}x \end{aligned}$$

subject to the two-dimensional advection-diffusion equation

$$\begin{cases} \frac{\partial}{\partial t} y(x,t) - \nabla \cdot (\nu \nabla y(x,t) + \mathbf{b} y(x,t)) = u(x,t) & \text{in} [0,T] \times \Omega \end{cases}$$

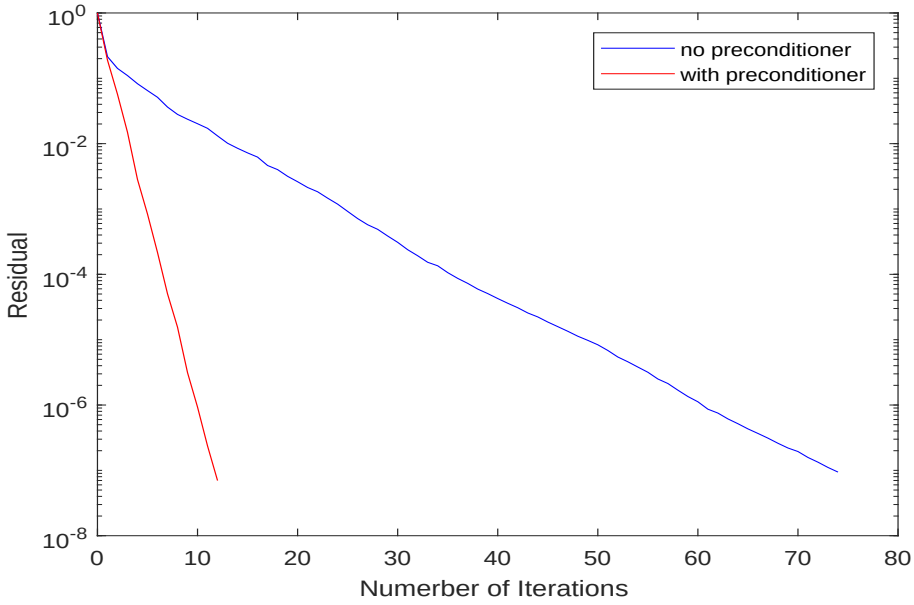
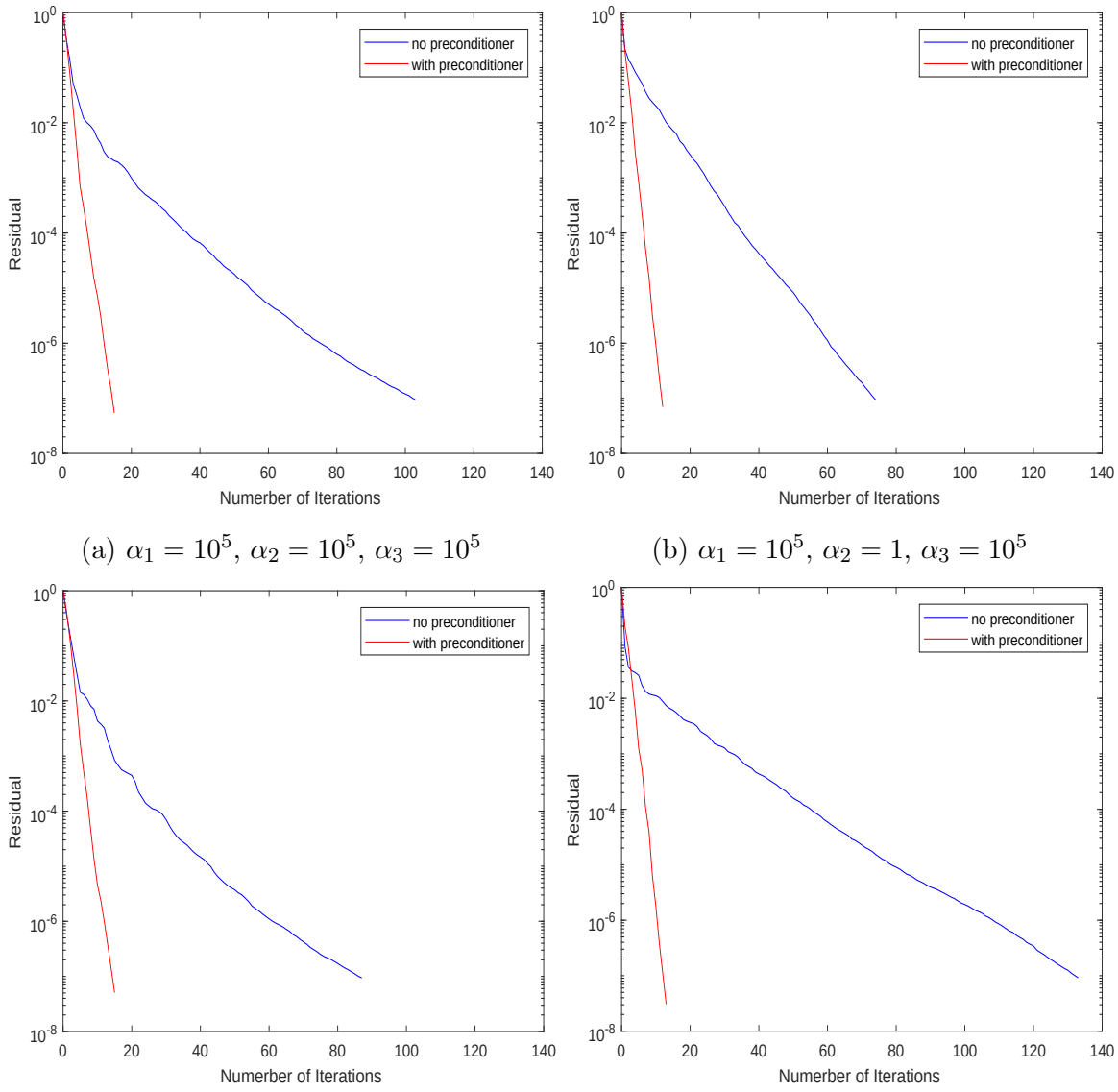


Figure 4.3: Performances of preconditioner P_5^{-1} (4.3) on the linear system (2.34) for $\alpha_1 = 10^5$

where $\mathbf{b} = (b_1, b_2)$ is a vector field satisfying the Stokes system

$$\begin{cases} \Delta b_1 + p_{x_1} = 0 \\ \Delta b_2 + p_{x_2} = 0 \\ \nabla \cdot \mathbf{b} = 0 \end{cases}$$

Again, we set $T = 0.2$, $n_t = 16$, $\tau = 0.0125$, and other settings are as same as section 4.3.1. FGMRES with residual norm toleration less than 10^{-7} is applied. The preconditioner P_6^{-1} is tested with different α_1, α_2, and α_3. Figure (4.4) and table (4.1) showed our numerical result.



(a) $\alpha_1 = 10^5$, $\alpha_2 = 10^5$, $\alpha_3 = 10^5$

(b) $\alpha_1 = 10^5$, $\alpha_2 = 1$, $\alpha_3 = 10^5$

(c) $\alpha_1 = 10^5$, $\alpha_2 = 10^5$, $\alpha_3 = 1$

(d) $\alpha_1 = 10^5$, $\alpha_2 = 1$, $\alpha_3 = 1$

Figure 4.4: Performances of preconditioner P_6^{-1} (4.4) on the linear system (2.35).

Symmetric A	α_1	α_2	α_3	No Preconditioner	With Preconditioner
Target States	N/A	10^5	10^5	45	23
Target States	N/A	1	10^5	6	7
Target States	N/A	10^5	1	45	23
Tracking	10^5	N/A	N/A	74	12
Tracking	1	N/A	N/A	3	5
All-time	10^5	10^5	10^5	103	15
All-time	10^5	1	10^5	74	12
All-time	10^5	10^5	1	87	15
All-time	10^5	1	1	133	13

Table 4.1: Number of iterations for FGMRES for non-symmetric A with different parameters.

Chapter 5

Conclusion and Future Development

5.1 Conclusion

In this thesis, we discussed the computational methods for the Linear Parabolic Optimal Control Problems. We considered three different types of problems. Namely, Target States Problems, Tracking Problems, and All-time Problems. For each of these types, we also addressed the case whether the differential matrix operator A is symmetric or not.

We have shown that with the *discretize-then-optimize* approach or *optimize-then-discretize* approach, one can obtain a system of linear equations from the problem. The solution of this system of linear equations is equivalent to the first-order necessary optimal condition of our interested optimal control problems. We proposed the shooting method and the *flexible generalized minimal residual* (FGMRES) method to reduce the computational cost of solving such linear system.

The major contribution of this thesis is that we developed a preconditioner to accelerate the FGMRES method for each case. For the target states problems with symmetric A, by integrating the coupled-ODE system backward and then forward, we obtained an expression of the shooting matrix in terms of A. Hence, we can have

a preconditioner by approximating the shooting matrix when A has high or low frequencies. On the other hand, when A is non-symmetric, our preconditioner became a solution of *Continuous Algebraic Riccati Equation* (CARE). So we can form a preconditioner by solving CARE with some standard algorithm available in the literature.

For the tracking problems with symmetric A, we expressed the shooting matrix in terms of K, B_1, and C_1, where B_1 and C_1 are generated from the *discretize-then-optimize* approach. We demonstrated that, in a reasonable amount of computational cost, we can form a preconditioner with eigenvalues bound $[\frac{1}{2}, 1)$. Moreover, we have shown that the same eigenvalues bound can be achieved with non-symmetric A.

The idea for the all-time problems is similar to the tracking problems. However, we notice that the eigenvalues bound $[\frac{1}{2}, 1)$ cannot be guaranteed. We explored the limitation of our preconditioner and we conclude that the eigenvalues bound can be achieved if we introduce some restrictions on our parameters. For a non-symmetric A, we proved that the same restrictions on the parameters can be applied to obtain the same eigenvalues bound.

To sum up, we proposed an effective iterative scheme to compute the optimal solution for our interested linear parabolic optimal control problems. And for each case, a corresponding preconditioner was developed to accelerate the convergence rate.

5.2 Future Development

In the future, there are many potential areas in this field for us to explore. In this section, we will name three major focuses of our future development.

Firstly, we would like to reduce the computational cost of our proposed preconditioners. It is well-known that we can apply multi-grid scheme on our preconditioned iterative method to achieve a lower computational cost. In the future, we will explore

the potential of such method and the corresponding acceleration rate.

Secondly, we believed that we should extend our methods to some more realistic problems. A real-world problem probably has more restrictions on the control variable comparing to what we considered in this thesis. Take the optimal heating control as an example, the heater should have some physical or technical limitations. Moreover, one may not be able to have the heating source distributed all-over the domain. Consider the domain described in figure 5.1. Assume that our heat source is only distributed in $\Omega_C \subseteq \Omega$. Then our tracking problems read as

$$\min_{y,u} \frac{1}{2} \int_0^T \int_{\Omega_C} u^2 \mathrm{d}x\mathrm{d}t + \frac{\alpha_1}{2} \int_0^T \int_{\Omega} (y - \hat{y})^2 \, \mathrm{d}x\mathrm{d}t$$

subject to the constraints

$$\begin{cases} \frac{\partial}{\partial t} y - c\Delta y = u + f & \text{in} \quad [0,T] \times \Omega_C \\ \frac{\partial}{\partial t} y - c\Delta y = f & \text{in} \quad [0,T] \times \Omega \backslash \Omega_C \\ y = g & \text{in} \quad [0,T] \times \partial\Omega \\ u_a \leq u \leq u_b & \text{in} \quad [0,T] \times \Omega_C \end{cases}$$

For this type of optimal control problems, the matrices arose from the discretization would have different structures comparing to our cases. Therefore, a new preconditioning method is required.

Last but not least, our preconditioners proposed in this thesis have a high chance to work well with other methods. We are interested in the parallel implementation in particular, since very often our parabolic optimal control problems would resulted in a extremely long computation time. The parallel methods should be able to greatly reduce our computational efforts. In [28], a parallel method is presented by Kwok for the parabolic optimal control problems. The problem is decomposed into many small problems that can be solved parallelly. Our preconditioning scheme for all-time problems is suitable for solving these small problems. Therefore, we believed that combing our preconditioners with the scheme in [28], a promising result could be achieved.

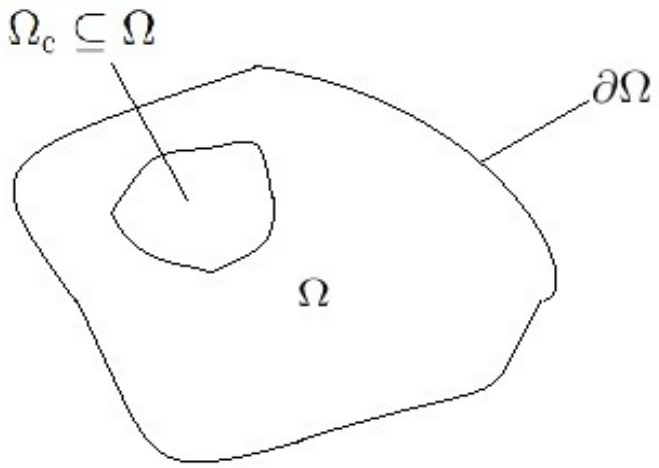


Figure 5.1: A visualization of domain $\Omega_C \subseteq \Omega$.

Bibliography

[1] Werner Barthel, Christian John, and Fredi Tröltzsch. Optimal boundary control of a system of reaction diffusion equations. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 90(12):966–982, 2010.

[2] Michele Benzi, Gene H Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[3] Dario A Bini, Bruno Iannazzo, and Beatrice Meini. *Numerical solution of algebraic Riccati equations*. SIAM, 2011.

[4] Luise Blank, M Hassan Farshbaf-Shaker, Claudia Hecht, Josef Michl, and Christoph Rupprecht. Optimal control of Allen-Cahn systems. In *Trends in PDE Constrained Optimization*, pages 11–26. Springer, 2014.

[5] A Borzı, Julien Salomon, and Stefan Volkwein. Formulation and numerical solution of finite-level quantum optimal control problems. *Journal of Computational and Applied Mathematics*, 216(1):170–197, 2008.

[6] Alfio Borzì. Multigrid methods for parabolic distributed optimal control problems. *Journal of Computational and Applied Mathematics*, 157(2):365–382, 2003.

[7] Alfio Borzì and Ulrich Hohenester. Multigrid optimization schemes for solving Bose–Einstein condensate control problems. *SIAM Journal on Scientific Computing*, 30(1):441–462, 2008.

[8] Alfio Borzì and Volker Schulz. *Computational optimization of systems governed by partial differential equations*. SIAM, 2011.

[9] Alfio Borzì, Georg Stadler, and Ulrich Hohenester. Optimal quantum control in nanostructures: Theory and application to a generic three-level system. *Physical Review A*, 66(5):053811, 2002.

[10] Ugo Boscain, Grégoire Charlot, Jean-Paul Gauthier, Stéphane Guérin, and Hans-Rudolf Jauslin. Optimal control in laser-induced population transfer for two and three level quantum systems. *Journal of Mathematical Physics*, 43(5):2107–2132, 2002.

[11] Haecheon Choi, Michael Hinze, and Karl Kunisch. Instantaneous control of backward-facing step flows. *Applied Numerical Mathematics*, 31(2):133–158, 1999.

[12] Juan Carlos De los Reyes. *Numerical PDE-constrained optimization*. Springer, 2015.

[13] Thomas Franke, Ronald HW Hoppe, Christopher Linsenmann, Lothar Schmid, and Achim Wixforth. Optimal control of surface acoustic wave actuated sorting of biological cells. In *Trends in PDE Constrained Optimization*, pages 505–519. Springer, 2014.

[14] Martin J Gander, Felix Kwok. Schwarz methods for the time-parallel solution of parabolic control problems. In *Domain Decomposition Methods in Science and Engineering XXII*, pages 207–216. Springer, 2016.

[15] Martin J Gander, Felix Kwok, and Gerhard Wanner. Constrained optimization: From Lagrangian mechanics to optimal control and PDE constraints. In *Optimization with PDE Constraints*, pages 151–202. Springer, 2014.

[16] Roland Glowinski and Jacques-Louis Lions. Exact and approximate controllability for distributed parameter systems. *Acta Numerica*, 4:159–328, 1995.

[17] Gene H Golub and Charles F Van Loan. *Matrix Computations*, volume 3. JHU Press, 2012.

[18] Anne Greenbaum. *Iterative Methods for Solving Linear Systems*, volume 17. Siam, 1997.

[19] Roland Griesse and Stefan Volkwein. A primal-dual active set strategy for optimal boundary control of a nonlinear reaction-diffusion system. *SIAM Journal on Control and Optimization*, 44(2):467–494, 2005.

[20] Mats Gustafsson and Sailing He. An optimization approach to two-dimensional time domain electromagnetic inverse problems. *Radio Science*, 35(2):525–536, 2000.

[21] Eldad Haber. A parallel method for large scale time domain electromagnetic inverse problems. *Applied Numerical Mathematics*, 58(4):422–434, 2008.

[22] Helmut Harbrecht and Johannes Tausch. On shape optimization with parabolic state equation. In *Trends in PDE Constrained Optimization*, pages 213–229. Springer, 2014.

[23] Matthias Heinkenschloss. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *Journal of Computational and Applied Mathematics*, 173(1):169–198, 2005.

[24] Nicholas J Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26(4):1179–1193, 2005.

[25] Nicholas J Higham. *Functions of Matrices: Theory and Computation*. SIAM, 2008.

[26] Arieh Iserles. *A First Course in the Numerical Analysis of Differential Equations*. Number 44. Cambridge University Press, 2009.

[27] Michael V Klibanov and Thomas R Lucas. Numerical solution of a parabolic inverse problem in optical tomography using experimental data. *SIAM Journal on Applied Mathematics*, 59(5):1763–1789, 1999.

[28] Felix Kwok. On the time-domain decomposition of parabolic optimal control problems. In *Domain Decomposition Methods in Science and Engineering XXIII*, pages 55–67. Springer, 2017.

[29] John E Lagnese and Günter Leugering. Time-domain decomposition of optimal control problems for the wave equation. *Systems and Control Letters*, 48(3):229–242, 2003.

[30] Yvon Maday and Gabriel Turinici. New formulations of monotonically convergent quantum control algorithms. *The Journal of Chemical Physics*, 118(18):8191–8196, 2003.

[31] Carlos E Orozco and ON Ghattas. Massively parallel aerodynamic shape optimization. *Computing Systems in Engineering*, 3(1-4):311–320, 1992.

[32] John W Pearson. Block triangular preconditioning for time-dependent Stokes control. *PAMM*, 15(1):727–730, 2015.

[33] John W Pearson, Martin Stoll, and Andrew J Wathen. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1126–1152, 2012.

[34] Anthony P Peirce, Mohammed A Dahleh, and Herschel Rabitz. Optimal control of quantum-mechanical systems: Existence, numerical approximation, and applications. *Physical Review A*, 37(12):4950, 1988.

[35] A Potschka, A Küpper, JP Schlöder, HG Bock, and S Engell. Optimal control of periodic adsorption processes: The Newton-Picard inexact SQP method. *Recent Advances in Optimization and its Applications in Engineering*, pages 361–378, 2010.

[36] Andreas Potschka. *A direct method for parabolic PDE constrained optimization problems*. Springer Science & Business Media, 2013.

[37] Tyrone Rees, Martin Stoll, and Andy Wathen. All-at-once preconditioning in PDE-constrained optimization. *Kybernetika*, 46(2):341–360, 2010.

[38] IT Rekanos. Time-domain inverse scattering using Lagrange multipliers: An iterative FDTD-based optimization technique. *Journal of Electromagnetic Waves and Applications*, 17(2):271–289, 2003.

[39] Youcef Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[40] Youcef Saad and Martin H Schultz. Gmres: A generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[41] Yousef Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM Journal on Numerical Analysis*, 29(1):209–228, 1992.

[42] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[43] Yousef Saad and Henk A Van Der Vorst. Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1):1–33, 2000.

[44] Marcus Sarkis, Christian E Schaerer, and Tarek Mathew. Block diagonal parareal preconditioner for parabolic optimal control problems. In *Domain Decomposition Methods in Science and Engineering XVII*, pages 409–416. Springer, 2008.

[45] Anton Schiela and Stefan Ulbrich. Operator preconditioning for a class of inequality constrained optimal control problems. *SIAM Journal on Optimization*, 24(1):435–466, 2014.

[46] Stephan Schmidt, Caslav Ilic, Volker Schulz, and Nicolas R Gauger. Three-dimensional large-scale aerodynamic shape optimization based on shape calculus. *AIAA journal*, 2013.

[47] Stephan Schmidt, Caslav Ilic, Volker Schulz, and Nicolas R Gauger. Three-dimensional large-scale aerodynamic shape optimization based on shape calculus. *AIAA journal*, 2013.

[48] Fredi Tröltzsch. Optimal control of partial differential equations. *Graduate Studies in Mathematics*, 112, 2010.

[49] Andreas Unger and Fredi Tröltzsch. Fast solution of optimal control problems in the selective cooling of steel. *ZAMM-Journal of Applied Mathematics and*

Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 81(7):447–456, 2001.

[50] Charles Van Loan. Computing integrals involving the matrix exponential. *IEEE transactions on automatic control*, 23(3):395–404, 1978.

[51] Gregory von Winckel, Alfio Borzi, and Stefan Volkwein. A globalized Newton method for the accurate solution of a dipole quantum control problem. *SIAM Journal on Scientific Computing*, 31(6):4176–4203, 2009.

CURRICULUM VITAE

Academic qualification of the thesis author, Mr. TSANG Siu Chung:

- Received the Degree of Bachelor of Science from Hong Kong Baptist University, November 2015

October 2017