A reduced order framework for optimal control of nonlinear partial differential equations

ROM-based Optimal Flow Control

Nicholas H. Nelsen

A Thesis Submitted to the
School of Mechanical & Aerospace Engineering
and the Honors College
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Science

Oklahoma State University
Stillwater, Oklahoma, USA
Primary Advisor: Omer San

May 2018
(Defended May 2, 2018)

To Jamey Jacob, without whom this work would not have even gotten off the ground

Acknowledgments

First and foremost, I want to thank my senior honors thesis advisor Professor Omer San in the Oklahoma State University (OSU) School of Mechanical and Aerospace Engineering (MAE) for taking me on as his first undergraduate student. Dr. San has been engaged with my learning process and is always supportive of me whenever I need technical assistance, direction, or advice.

I engaged in fruitful discussions with my co-advisor Professor Weiwei Hu in the Department of Mathematics on the more mathematical aspects of this thesis, and for that I am grateful; she has been a great collaborator and has taught me much of what I know about numerical analysis.

With regard to my numerous research experiences during these four years at OSU, I cannot forget to mention Jessica Sullins and the Office of Scholar Development and Undergraduate Research. Additionally, the support I have received from the OSU Honors College has been extremely encouraging during this process. Dean Keith Garbutt's belief in my ability to complete *two* honors college degrees has certainly helped keep me going all this time.

Most significantly, I am indebted to the John Hendrix Chair and Professor of Aerospace Engineering Jamey Jacob for wholeheartedly accepting my request to conduct an independent capstone project and senior honors thesis. I am grateful for Professor Jacob's help in getting this idea through the MAE administration and always supporting my drive to succeed in all aspects of my undergraduate career. As the first student in Aerospace Engineering to perform a research-based capstone, I hope my experience will set a precedent in enabling this option for students in future years. Relatedly, I thank Dr. James Kidd for allowing me some much needed flexibility in fulfilling all of the requirements for his MAE 4374 capstone course section, upon which my thesis was attached.

To the graduate students of the Computational Fluid Dynamics Laboratory (Romit Maulik, Mashfiqur Rahman, and Mansoor Ahmed), thank you for being so welcoming during my short tenure with you all and for broadening my knowledge of computational science. In particular, Romit has been a great friend and mentor; I enjoyed our long chats in the lab and the advice he has shared.

Lastly, I am grateful for the support of my family, Dan, Hannah, and Timmy Nelsen; never once were my ambitions questioned or stymied, no matter how outlandish these seemed at the time.

A reduced order framework for optimal control of nonlinear partial differential equations

ROM-based Optimal Flow Control

Thesis by
Nicholas H. Nelsen

In Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Science

Abstract

A variety of partial differential equations (PDE) can govern the spatial and time evolution of fluid flows; however, direct numerical simulation (DNS) of the Euler or Navier–Stokes equation or other traditional computational fluid dynamics (CFD) models can be computationally expensive and intractable. An alternative is to use model order reduction techniques, e.g., reduced order models (ROM) via proper orthogonal decomposition (POD) or dynamic mode decomposition (DMD), to reduce the dimensionality of these nonlinear dynamical systems while still retaining the essential physics. The objective of this work is to design a reduced order numerical framework for effective simulation and control of complex flow phenomena. To build our computational method with this philosophy, we first simulate the 1D Burgers' equation $u_t + uu_x - \nu u_{xx} = f(x,t)$, a well-known PDE modeling nonlinear advection-diffusion flow physics and shock waves, as a full order high resolution benchmark. We then apply canonical reduction approaches incorporating Fourier and POD modes with a Galerkin projection to approximate the solution to the posed initial boundary value problem. The control objective is simple: we seek the optimal (pointwise) input into the system that forces the spatial evolution of the PDE solution to converge to a preselected target state $u_T(x)$ at some final time $T > 0$. To implement an iterative control loop, we parametrize the unknown control function as a truncated Fourier series defined via a set of finite parameters. The performance of the POD ROM is compared to that of the Fourier ROM and full order model for six numerical experiments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Partial differential equations is one of the most diverse and ubiquitous fields of both pure and applied mathematics. A **partial differential equation** (PDE) is an equation involving an unknown function in two or variables and some order of partial derivatives of this function with respect to these variables. Not entirely unlike ordinary differential equations (ODE), the vast majority of PDE do not have explicit solutions or representations. However, in contrast to ODE, PDE do not have a comprehensive framework for their analysis and solution. Strikingly similar PDE can have erratically different behavior, and with different behavior comes the need to utilize an extensive repertoire of methods. To solve a PDE in practice, we typically turn to a computer and write computer codes using techniques from numerical analysis to obtain a *numerical solution*. Or if we desire to be solely theoretical, we deduce results about the solutions to PDE, such as existence and uniqueness, without actually writing down what the solution is. Here, mathematicians relax the *classical* sense of a smooth, continuous solution in favor of generalized *weak* solutions that are easier to work with in proofs. In either case, the study of PDE is one of the richest and most interdisciplinary fields of mathematics that permeates deeply into areas of fundamental and applied science.

In describing the flow of fluids, PDE such as the Navier–Stokes equation have captured the attention of mathematicians and the general science community alike. There is an outstanding one-million dollar Hilbert Millennium Prize problem for proof or disproof of existence and global regularity for the 3D Navier–Stokes equation [21]. Recently, progress has been made on the side of disproof: Buckmaster and Vicol [12] have proven that a certain (wild) class of weak solutions develop singularities in finite time, termed “finite time blow up”. These extremely esoteric results were even communicated to a broader scientific audience via online news and magazine outlets such as WIRED and Quanta. Clearly, the mathematical fluid dynamics community is quite active and truly hard problems are still outstanding in this field.

Much of modern science, and in particular scientific computing, is devoted to solving PDE with relevant initial and boundary conditions that model complex physical processes. However, multi-scale nonlinear dynamical systems often emerge from spatial discretization of these equations that quickly overwhelm even the most impressive computational resources. In the case of fluid dynamics, the complexity of the Navier–Stokes equation calls for an approach to simplify or approximate the full system for the purpose of understanding fundamental mechanisms that describe fluid flow evolution. Since high resolution direct numerical simulations of PDE impose prohibitive computational overhead for realistic governing laws, the field of reduced order modeling has emerged as a candidate solution.

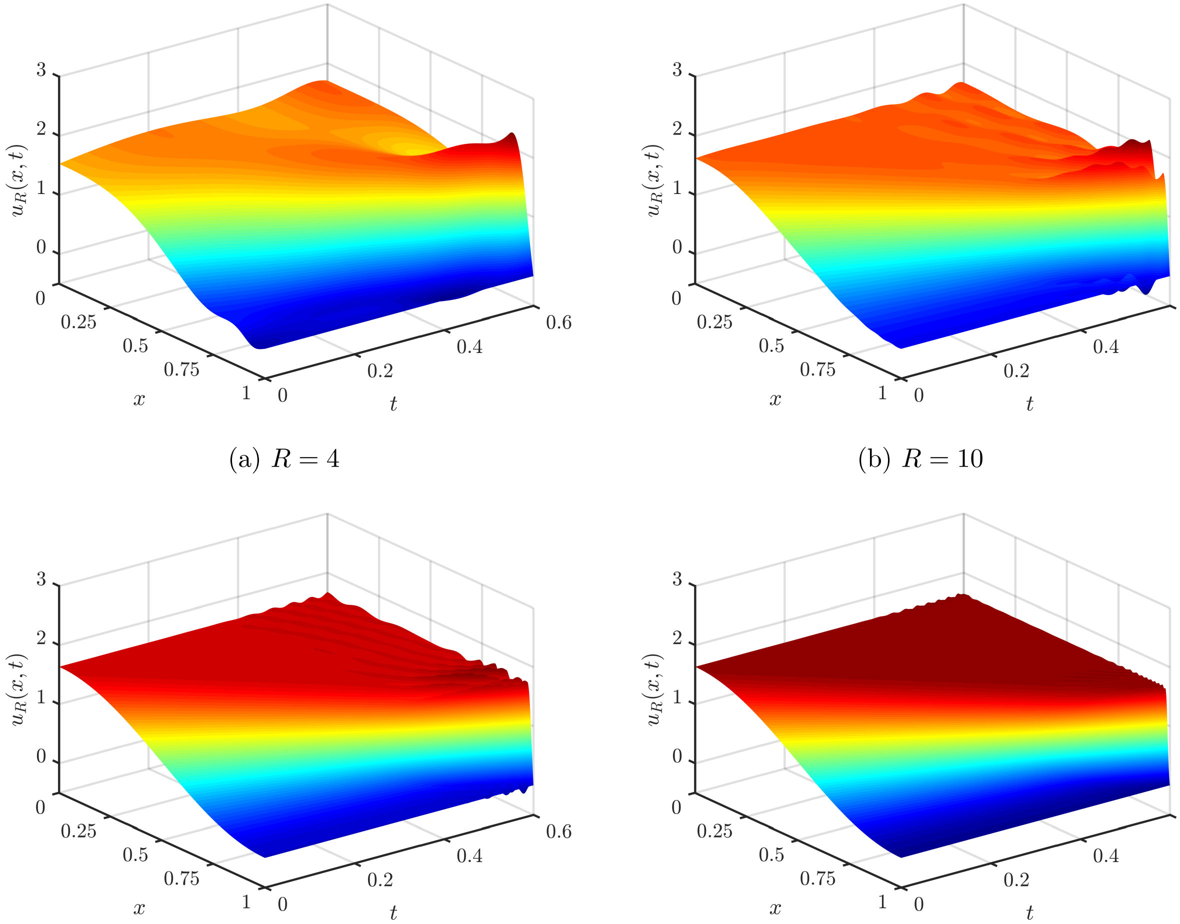
1.1 Model Order Reduction

By simplifying the computational complexity of an underlying governing law, typically a PDE or system of PDE, reduced order models (ROMs) display great promise in many prediction, identification, design, optimization, and control applications. Model reduction approaches are especially effective in settings where traditional methods require repeated model evaluations over a similar range of input values. However, ROMs suffer from lack of robustness with respect to large parameter changes and high upfront costs to construct them. Additionally, a truncation procedure inherent in the process of reducing system dimensionality in effect tosses out high order information that may play a crucial role in capturing the coupled nonlinear behavior of convection-dominant or turbulent flows [53, 54, 55].

The underlying concept of reduced order modeling is the reduction of the number of degrees of freedom (*modes*) in a dynamical system (e.g., from $D = \mathcal{O}(10^6)$ to $R = 50 \ll D$). This directly translates into savings of computational cost, especially when the physical dimension of the system is increased (i.e., from 1D to 2D or 3D). The degrees of freedom, or "dimension" here, can be loosely thought of as how many ordinary differential equations are needed to fully describe the system of interest. Approximating a complex dynamical system with very few degrees of freedom is especially of relevance to feedback flow control laws and related applications.

Projection-based reduced order modeling offers a simple framework to accomplish this objective. Here, low dimensional representations of the high dimensional governing law are constructed in a subspace spanned by a carefully selected orthonormal basis. For example, these basis functions can be the standard Fourier series or Chebyshev or Legendre polynomials. It is still an open challenge to guarantee that the dynamics in this transformed space accurately reflect the behavior of the true physical system governed by the PDE.

After a suitable basis is chosen, the *Galerkin projection* is one leading approach used to project the dynamics of the equation onto the subspace spanned by a truncated set of basis functions. This variational (weak form) approach yields a finite system of ODE from the original PDE; the number of ODE is equal to the number of basis functions selected to approximate the original function space in which the PDE was posed. The accuracy of reduced order models will certainly improve with an increased number of modes [see Figure 1.1]; this is due to the nature of the Galerkin method used to construct the ROM. Yet, there is a threshold beyond which more modes fail to noticeably contribute to accuracy. Further, it is not entirely understood how to choose the number of modes that will guarantee a specified error bound for arbitrary dynamical systems; in practice, a few trial runs is sufficient to determine an optimal number of modes for the problem at hand.



(a) $R = 4$

(b) $R = 10$

(c) $R = 20$

(d) $R = 40$

Figure 1.1: Improvement in a POD–G–ROM approximation of a PDE as $R \to \infty$.

Model reduction has been extremely successful in the field of fluid dynamics, where massive experimental or numerical data sets are commonplace. Galerkin projections for linear systems are trivial and their extension to simple nonlinear systems modeling incompressible flow has been well-studied [47]. In more challenging dynamical systems, such as the fully compressible Navier–Stokes equations, corresponding ROMs have also been derived and implemented [5, 48]. One of the most popular approaches is Proper Orthogonal Decomposition (POD), which extracts the highest energy structures or dominant statistical characteristics from a flow field to construct a basis. However, it is known that low energy, high frequency modes can have a profound influence on the dynamics [49]. Thus, some low-dimensional POD ROMs do not accurately represent the original PDE (e.g., in fluid systems with high transient growth such as shear flows). Care must be taken when choosing to use a POD, as it is not the optimal choice in all model reduction problems. Other techniques of note include Balanced Truncation, Balanced Proper Orthogonal Decomposition, Goman-Khrabrov Models, eigensystem realization algorithms, and Dynamic Mode Decomposition and its variants [49, 55].

Computation of reduced order models typically occur in what are known as the "offline" and "online" stages. In the offline stage, the user provides input data taken from experimental measurements or high fidelity numerical simulations. The goal of the offline stage is to set up the reduced order model by decomposing the flow field, creating snapshot vectors, calculating expensive coefficients that arise from the Galerkin projection, and computing basis functions for the specific model reduction method used. The cost for this process is $\mathcal{O}(N^3)$ where N is the number of modes; this points to the fact that using reduced order models is actually prohibitive when N is larger than say $\mathcal{O}(10^1)$ and certainly ill-advised when $N^3 = \mathcal{O}(D)$ where D is the rank of the corresponding full order problem or data set.

The online stage follows in a simple input/output manner. Since the basis and coefficients are pre-computed, the dynamics can be temporally evolved in the modal space. For example, an initial condition is prescribed and the reduced system of coupled ODE is then integrated in time to produce the output solution. In this work, the online stage composes the not only the solution step to the system of ODE, but also the iterative control loop (see Section 3.2) for state function matching.

1.2 Optimal Control

In this work, we design a reduced order computational framework for effective simulation and optimal control of challenging flows. The use of reduced order models in control theoretic applications has been surging recently [see, e.g.: 2, 8, 15, 16, 29, 40, 46]. A thorough discussion of computational flow control is given in the reviews [9, 49, 71]. The field is also ever expanding in scope and original ideas. Proctor, Brunton, and Kutz in [44] extend the linear theory of Dynamic Mode Decomposition to the control of complex nonlinear systems with the aid of the Koopman operator. The novelty in this example is the equation-free architecture of the method; that is, the data-driven approach can predict future states of dynamical systems with only input data. Additionally, waves of popularity in machine learning have spurred interest in flow prediction and points to future optimal control frameworks [45, 55, 56, 57, 58].

The modern mathematical treatment of optimal control theory, which includes proving existence and uniqueness of controllers and minimizing objective functionals with the calculus of variations, has paved the way for now state-of-the-art computational approaches. An overview of some current issues in ROM-based control can be found in [49, 51]. In particular, uncertainty quantification and basis selection are open questions in this field. We intend to partially address the latter in this work.

1.3 Preface

We emphasize that this thesis was inspired by the desire to improve upon the results of KUCUK and SADEK in [26] with a novel reduced order framework. While there is no claim that all results in this document are wholly original, deliberate effort has been placed in carefully interpreting both theoretical and computational studies, and all computer codes were written from scratch in the software MATLAB. When required, we made use of standard numerical packages for solving ODE and eigenvalue problems. Additionally, the following notation is used throughout this work:

Notation 1.3.0.1. We use the acronym "PDE" to refer to both a single *partial differential equation* and multiple *partial differential equations*. The meaning should be clear from the context, and this notation is common practice in the mathematical sciences.

Notation 1.3.0.2. The variable "x" and other related symbols are used in the context of both 1-dimensional ($\mathbb{R}$) and d-dimensional ($\mathbb{R}^d$) Euclidean spaces; we do not invoke bold face text to denote vectors, save for a few exceptions. Instead, the meaning should be obvious from the context. When appropriate, the dot symbol "$\cdot$" is used to denote the standard Euclidean inner product.

Notation 1.3.0.3. To denote the standard partial derivatives, we mainly utilize the notation $\frac{\partial}{\partial x_i}$ and $\frac{\partial}{\partial t}$ in Leibniz form or the more short hand $(\cdot)_x$ and $(\cdot)_t$. The notation ∂_x and ∂_t is seldom used here. We denote the Laplacian operator as Δ instead of the perhaps more common ∇^2 seen in physics.

This thesis is organized as follows. In Chapter 2 we detail the motivation for studying Burgers' equation and solve the initial value problem analytically, showcasing the challenges of dealing with the deceptively simple quadratic nonlinearity. We also pose the initial boundary value problem that is the test bed for our numerical treatment of reduced order models and their control. Chapter 3 discusses the relatively expensive full order model and its optimal control results. Two reduced order models, Fourier–ROM and POD–G–ROM, are derived in Chapter 4 and performance is compared. Finally, in Chapter 5 we provide some concluding remarks and address avenues for future research.

Chapter 2

Burgers' Equation

Burgers' equation is the simplest nonlinear PDE one can study, and it comes in two standard variants: inviscid and viscous. The inviscid Burgers' equation is a hyperbolic conservation law (of the form $\theta_t + \nabla F(\theta) = 0$, where F is a flux) exhibiting finite time singularities in its solution or gradient, making this equation difficult to simulate numerically without shock-capturing methods:

$$\begin{cases} \frac{\partial u}{\partial t} + \nabla \left(\frac{1}{2}u^2\right) = 0, \ x \in \mathbb{R}^d, \ t > 0 \\ u(x,0) = u_0(x) \end{cases} \tag{2.1}$$

In particular, it can be proven that shocks form if the initial data satisfies $\nabla u_0 < 0$ on some interval; specifically, the breaking time is $t_b = \frac{1}{\min_{x \in \mathbb{R}^d}\{\nabla u_0\}}$.

On the other hand, the viscous Burgers' equation is of the parabolic-type, with a nice diffusion term $-\nu\Delta u$ that dissipates high frequency, high gradient content:

$$\begin{cases} \frac{\partial u}{\partial t} + u \cdot \nabla u - \nu \Delta u = f, \ x \in \mathbb{R}^d, \ t > 0 \\ u(x,0) = u_0(x) \end{cases} \tag{2.2}$$

where $\nu > 0$. This version of the equation is well studied in the literature and is called the *viscous Burgers' equation.* In contrast to Eqn. (2.1), the IVP (2.2) is globally well-posed with smooth solutions due to the presence of viscosity. The viscous term provides enough regularity to overcome the nonlinear convection and prevent the formation of a singularity.

This prototypical PDE, though simple, exhibits fascinating behavior that translates well into applications in science and engineering. For example, Burgers' equation is the first standard model problem to test new numerical methods for solving PDE, especially since there are exact solutions. It exhibits the same type of quadratic nonlinearity, $u \cdot \nabla u = \nabla(\frac{1}{2}u^2)$, shared with the Euler and

Navier–Stokes equations:

$$\begin{cases} \frac{\partial u}{\partial t} + u \cdot \nabla u - \nu \Delta u = -\nabla p, \quad x \in \mathbb{R}^d, \ t > 0 \\ \nabla \cdot u = 0 \\ u(x, 0) = u_0(x) \end{cases} \tag{2.3}$$

where $d = 2, 3$ and $\nu \geq 0$. The Navier–Stokes equation simply enforces the conservation of mass and momentum for an incompressible fluid, arising from Newton's Second Law and energy principles. The only difference between Burgers' equation and the Navier–Stokes equation is the pressure term and enforcement of incompressibility via the divergence-free condition, both in the latter.

It is clear why in some contexts, Burgers' equation is termed the "1D Navier–Stokes equation". Indeed, many popular fluid dynamics solvers are designed for complex 3D flows governed by the Navier–Stokes equation or its many coupled extensions (e.g.: radiative heat transfer, chemically reacting flows, particle problems). However, engineers test their numerical methods on the Burgers' equation first because if the code fails for this case, it will certainly fail for more challenging PDE such as Eqn. (2.3). We first gleam some insight into the structure of solutions by solving Eqn. (2.1) and Eqn. (2.2) analytically before turning to flow control applications with reduced order models.

2.1 Analytical Solution

We can solve the inviscid Burgers' equation (2.1) analytically using the method of characteristics. Shock front formation is a result of the characteristic curves intersecting; similarly, an expansion or rarefaction wave is generated by the characteristics diverging from one another. Considering the characteristic $x = \xi + u(x, t)t$ with variable speed, we find an implicit solution for the inviscid Burgers' equation to be

$$u(x, t) = u_0(x - u(x, t)t). \tag{2.4}$$

There is no closed form solution in general, but using this relationship to plot the characteristics in the x-t plane can provide insight on how the curves interact and intersect.

The viscous Burgers' equation can also be solved exactly for the whole space $\mathbb{R}^d$. We use a powerful technique thats converts the nonlinear PDE into an easy linear PDE, the **Cole-Hopf transformation** [19]. Many authors perform the *ansatz* setting $w = e^{\frac{-cu}{\nu}}$ and changing variables in Eqn. (2.2) to obtain the heat equation. We will take this approach, but first justify where the substitution came from. Consider the Cauchy problem for the parabolic PDE with a quadratic nonlinearity in $\mathbb{R}^d$:

$$\begin{cases} \frac{\partial u}{\partial t} - \nu \Delta u + c\left|\nabla u\right|^2 = 0 \\ u(x,0) = u_0(x) \end{cases} \tag{2.5}$$

where $\nu > 0$. We assume that u has sufficient regularity and assign the unknown function

$$\phi : \mathbb{R} \longrightarrow \mathbb{R} \quad \text{smooth}$$

with $w := \phi(u)$. The goal here is to find a smooth function ϕ such that we force w to solve a linear PDE. Computing derivatives with the chain rule, we obtain

$$\frac{\partial w}{\partial t} = \phi'(u)\frac{\partial u}{\partial t}$$

for time and

$$\Delta w = \phi'(u)\Delta u + \phi''(u)\left|\nabla u\right|^2$$

for space. But by our carefully selected system (2.5) above, we see that

$$\begin{aligned} \frac{\partial w}{\partial t} &= \phi'(u)\frac{\partial u}{\partial t} = \phi'(u)(\nu\Delta u - c\left|\nabla u\right|^2) \\ &= \nu\phi'(u)\Delta u - c\phi'(u)\left|\nabla u\right|^2 \\ &= \nu(\Delta w - \phi''(u)\left|\nabla u\right|^2) - c\phi'(u)\left|\nabla u\right|^2 \\ &= \nu\Delta w - (\nu\phi''(u) + c\phi'(u))\left|\nabla u\right|^2 \end{aligned}$$

Clearly we obtain the linear heat equation if ϕ solves the ODE $\nu\phi'' + c\phi' = 0$. Thus, set

$$\phi(y) = e^{-\frac{cy}{\nu}} \tag{2.6}$$

as the *Cole-Hopf transform*. More specifically, if u solves the nonlinear equation (2.5), then $w = e^{-\frac{cu}{\nu}}$ solves the diffusion equation IVP

$$\begin{cases} \frac{\partial w}{\partial t} - \nu\Delta w = 0, \quad x \in \mathbb{R}^d,\ t > 0 \\ w(x,0) = e^{\frac{-cu_0(x)}{\nu}}, \quad x \in \mathbb{R}^d \end{cases} \tag{2.7}$$

The unique solution for this problem [see 19, for a complete mathematical treatment] is given by the convolution with the heat kernel

$$w(x,t) = \frac{1}{(4\pi\nu t)^{\frac{d}{2}}} \int_{\mathbb{R}^d} e^{-\frac{|x-y|^2}{4\nu t}} e^{-\frac{cu_0(x)}{\nu}}\ dy$$

and hence by the Cole-Hopf transform we may recover u by

$$u(x,t) = -\frac{\nu}{c}\log(w(x,t)) = -\frac{\nu}{c}\log\left(\frac{1}{(4\pi\nu t)^{\frac{d}{2}}}\int_{\mathbb{R}^d} e^{-\frac{|x-y|^2}{4\nu t}} e^{-\frac{cu_0(x)}{\nu}}\,dy\right). \tag{2.8}$$

We can now proceed to use the Cole-Hopf transform to solve the 1D viscous Burgers' equation on the whole real line. Our system of interest is

$$\begin{cases} u_t + uu_x - \nu u_{xx} = 0, \; x \in \mathbb{R}, \; t > 0 \\ u(x,0) = u_0(x) \end{cases} \tag{2.9}$$

Our goal is to solve this equation using the above procedure. The first step is to rewrite equation (2.9) into the form of the parabolic PDE (2.5). We begin with the not so obvious change of variables

$$h(x,t) := \int_{-\infty}^{x} u(y,t)dy. \tag{2.10}$$

Applying this transformation is straightforward for the linear terms of Eqn. (2.9). For the nonlinear term uu_x, we rewrite it in conservation form as $\frac{\partial}{\partial x}(u^2/2)$ so that

$$\int_{-\infty}^{x}\frac{\partial}{\partial x}(u(y,t)^2/2)dy = \frac{d}{dx}\int_{-\infty}^{x}(u(y,t)^2/2)dy = u(x,t)^2/2 = \frac{1}{2}\left(\frac{\partial}{\partial x}h\right)^2$$

where $u(x,t) = \frac{\partial}{\partial x}h$ is obvious from Eqn. (2.10) and the fundamental theorem of calculus. Then, the transformed PDE is

$$\begin{cases} h_t + \frac{1}{2}(h_x)^2 - \nu h_{xx} = 0, \; x \in \mathbb{R}, \; t > 0 \\ h(x,0) = \int_{-\infty}^{x} u_0(\eta)\,d\eta \end{cases} \tag{2.11}$$

which indeed is of the form (2.5) with $c = 1/2$. Using the formula (2.8), the solution for the auxiliary variable h is

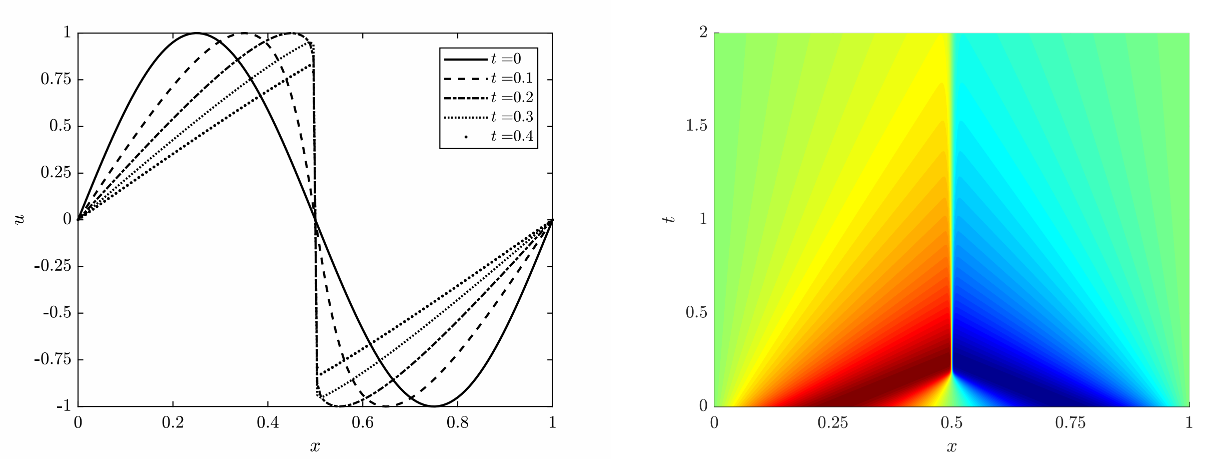
$$h(x,t) = -2\nu\log\left(\frac{1}{(4\pi\nu t)^{\frac{1}{2}}}\int_{\mathbb{R}} e^{-\frac{|x-y|^2}{4\nu t}} e^{-\frac{\int_{-\infty}^{y} u_0(\eta)\,d\eta}{2\nu}}\,dy\right).$$

Finally, taking a spatial derivative and canceling terms yields the exact solution to Eqn. (2.9),

$$u(x,t) = \frac{\displaystyle\int_{\mathbb{R}} \frac{x-y}{t}\, e^{\frac{-|x-y|^2}{4\nu t} - \frac{\int_{-\infty}^{y} u_0(\eta)d\eta}{2\nu}}\,dy}{\displaystyle\int_{\mathbb{R}} e^{\frac{-|x-y|^2}{4\nu t} - \frac{\int_{-\infty}^{y} u_0(\eta)d\eta}{2\nu}}\,dy} \qquad \forall x \in \mathbb{R}, \; t > 0. \tag{2.12}$$

While it is advantageous that an exact solution can be derived, this solution only holds for the IVP on the real line. For problems with boundary conditions, only a few exact solutions are known and even then, computing all of the integrals may be inconvenient. Hence, if an exact solution is not available to serve as a baseline result, we turn to standard direct numerical simulation (DNS) techniques for partial differential equations to obtain fine resolution data for Burgers' equation.

For example, prescribing the two mode sine wave $u_0(x) = \sin(2\pi x)$ on $[0, 1]$ in the Burgers' system leads to the well known shock behavior at $x = 0.5$ (Fig. 2.1a, solved with finite differences). The advection term causes $u(x, t)$ to progressively steepen with time. The viscosity term provides enough dissipation to prevent a singularity here, as we know that the solution is smooth for all time.



(a) Time evolution of $u(x, t)$ into a shock

(b) Contours of characteristic curves in x-t plane

Figure 2.1: Finite difference solution of Burgers' equation with $u_0(x) = \sin(2\pi x)$.

2.2 Initial Boundary Value Problem

The primary initial boundary value problem (IBVP) considered in this work is the 1D Burgers' system

$$\begin{cases} \text{PDE:} \quad u_t + uu_x - \nu u_{xx} = v(t)\delta(x-a), \quad (x,t) \in (0,1) \times (0,T] \\ \text{BC:} \quad u_x(0,t) = u(1,t) = 0, \quad t \in (0,T] \\ \text{IC:} \quad u(x,0) = \psi_1(x) = \sqrt{\tfrac{2}{3}}(1+\cos(\pi x)), \quad x \in (0,1) \end{cases} \tag{2.13}$$

where δ is the Dirac delta distribution, $\nu > 0$ is the kinematic viscosity, $a \in (0, 1)$ is the actuator position, T is the final time, $v(t)$ is a time-varying control input, and $\psi_n(x)$ are the orthonormal

Fourier basis functions [see 26] defined as

$$\psi_n(x) = \sqrt{\frac{4n-2}{2n+1}} \left(\frac{(-1)^{n+1}}{2n-1} + \cos(n\pi x) + \frac{2}{2n-1} \sum_{k=1}^{n-1} (-1)^{n+k+1} \cos(k\pi x) \right).$$

We denote the spatial domain as $\Omega = [0,1]$. The left endpoint boundary condition at $x = 0$ is of the Neumann type, $\frac{\partial u}{\partial x}|_{x=0} = 0$ and the right endpoint admits a Dirichlet boundary condition $u(x,t)|_{x=1} = 0$. The initial condition $u(x,0) = \psi_1(x)$ satisfies these homogeneous boundary conditions and is chosen to match that of [26].

The delta function, thought of as a unit "impulse" that assigns a single point unit mass and being so crucial to our pointwise control approach, is now defined as follows.
The **Dirac delta distribution** is a type of "generalized function" with the informal definition of

$$\delta(x-a) = \begin{cases} \infty, & x = a \\ 0, & x \neq a \end{cases} \tag{2.14}$$

for all $x, a \in \mathbb{R}^d$. More precisely, $\delta : \mathbb{R}^d \longrightarrow \mathbb{R}$ is a linear functional acting on test functions f with the fundamental properties that

$$\int_{\mathbb{R}^d} \delta(x)dx = 1$$

and

$$\int_{\mathbb{R}^d} f(x)\delta(x-a)dx = f(a).$$

Hence, the term $v(t)\delta(x-a)$ serves as a time-varying source or forcing (force density) that acts locally on the point $x = a$. Physically, this could represent gravity, magnetism, or some other force that is applied on a volume, and it is also analogous to a source or sink. This term is the mechanism that distributes the control function $v(t)$ to the PDE itself. The system (2.13) is called an *external distributed control* problem with force actuator $v(t)$. We interpret $u(x,t)$ as the velocity of a fluid particle constrained by the PDE to a trajectory along the interval $[0,1] \subset \mathbb{R}$.

The problem (2.13) has been well studied [8, 17, 26], and the solution develops a sharp front (*shock wave*) moving toward the right boundary as time increases. This challenging nonlinear wave behavior is a desirable trait on which to test the effectiveness of new numerical methods (as a first pass) before moving on to higher dimensional fluid flow PDE such as the full Navier–Stokes or Euler vorticity equations (2.3). We apply our proposed optimal control approach to this model system (2.13) in the following chapters.

Chapter 3

Full Order Model

The transient nonlinear dynamical systems of fluid dynamics and other fields of science display an incredible range of spatial and temporal scales. To resolve these fine details via numerical simulation would undoubtedly be cost prohibitive for all but the most state-of-the-art high performance computing centers. However, in studying toy model problems such as the 1D Burgers' equation, full order direct numerical simulation (DNS) can be conducted with negligible downsides. High resolution data from DNS is useful for performing modal decompositions from which to extract additional information about the underlying flow fields or to serve as benchmark cases for later comparisons to new models or methods. We undertake the latter in this work by using a second-order accurate finite difference scheme to solve the Burgers' system (2.13).

3.1 Numerical Methods

Today, the theory of finite difference schemes for solving partial differential equations is well developed. There exist methods that are accurate up to arbitrary order and more so that exhibit desirable properties such as suppression of numerical dissipation or dispersion. One of the most celebrated family of high order methods is the compact finite difference scheme [33], used extensively in simulations requiring enormous detail of the underlying physics. In this work, we opt to take a simpler approach by implementing the (1,2) accurate forward-time central-space (FTCS) finite difference scheme to solve the 1D viscous Burgers' equation in divergence form,

$$u_t + uu_x - \nu u_{xx} = f(x,t).$$

This equation models the unidirectional propagation of a single wave and does not require high order convergence to resolve fine features in the flow.

The initial boundary value problem (2.13) for the 1D Burgers' system is repeated here in Leibniz notation for clarity:

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} - \nu\dfrac{\partial^2 u}{\partial x^2} = v(t)\delta(x-a), \quad (x,t) \in (0,1)\times(0,T] \\ \dfrac{\partial u}{\partial x}(0,t) = u(1,t) = 0, \quad t \in (0,T] \\ u(x,0) = \psi_1(x) = \sqrt{\tfrac{2}{3}}(1+\cos(\pi x)), \quad x \in (0,1) \end{cases}$$

We begin by discretizing the space-time domain $[0,1]\times[0,T]$ (with boundaries included) into a grid of equally spaced points (x_j, t^n), $0 = x_0 < x_1 < \ldots < x_{N_x-1} < x_{N_x} = 1$ and $0 = t^0 < t^1 < \ldots < t^{N_t-1} < t^{N_t} = T$, where N_x is the number of points in space, N_t is the number of time steps, j is the spatial index, and n is the temporal index (time level). This is the grid-centered approach.

It is well known from the *von Neumann stability analysis* that the time step for FTCS applied to parabolic problems in d-dimensions is limited by the CFL-type stability condition [35]

$$\nu\frac{k}{h^2} \leq \frac{1}{2d},$$

where $k = t_{i+1} - t_i$ is the time step and $h = x_{i+1} - x_i$ is the mesh size of the discretization. We choose the time step

$$k = 0.8\left(\frac{h^2}{2\nu}\right) \tag{3.1}$$

with a buffer factor of 0.8 to avoid numerical blow up of the computed approximation to the solution. In the FTCS scheme, the time derivative is discretized in the Forward–Euler sense,

$$\frac{\partial u}{\partial t}(x_j,t^n) \approx \frac{\partial V}{\partial t}(x_j,t^n) = \frac{V_j^{n+1} - V_j^n}{k}, \tag{3.2}$$

where $V(x_j,t^n) = V_j^n \approx u(x_j,t^n)$ is the numerical solution to the discretized difference equations at the grid point (x_j,t^n). Forward–Euler converges as $\mathcal{O}(k)$ (first-order in time). Similarly,

$$\frac{\partial u}{\partial x}(x_j,t^n) \approx \frac{\partial V}{\partial x}(x_j,t^n) = \frac{V_{j+1}^n - V_{j-1}^n}{2h} \tag{3.3}$$

and

$$\frac{\partial^2 u}{\partial x^2}(x_j,t^n) \approx \frac{\partial^2 V}{\partial x^2}(x_j,t^n) = \frac{V_{j+1}^n - 2V_j^n + V_{j-1}^n}{h^2} \tag{3.4}$$

are the central difference approximations to the spatial derivatives and are both second-order accurate

($\mathcal{O}(h^2)$) schemes [35, 62]. Plugging these differences into the PDE, we obtain

$$\frac{V_j^{n+1}-V_j^n}{k}+(V_j^n)\frac{V_{j+1}^n-V_{j-1}^n}{2h}-\nu\frac{V_{j+1}^n-2V_j^n+V_{j-1}^n}{h^2}=f_j^n,$$

where $f_j^n=f(x_j,t^n)$. We define the numbers $\lambda=\frac{k}{h}$ and $\mu=\frac{k}{h^2}$ and rearrange to obtain the explicit form

$$V_j^{n+1}=V_j^n-\frac{\lambda}{2}(V_j^n)(V_{j+1}^n-V_{j-1}^n)+\nu\mu(V_{j+1}^n-2V_j^n+V_{j-1}^n)+kf_j^n. \tag{3.5}$$

To treat the two boundary conditions, a similar procedure is applied. The Dirichlet boundary is simple: $V(1,t^n)=V_{N_x}^n=0$ for all $n=0,1,2,\dots,T/k$. The Neumann condition $u_x(0,t)=0$ implies $\frac{V_1^n-V_{-1}^n}{2h}=0$ or $V_1^n=V_{-1}^n$. Substituting this relation into the FTCS scheme (3.5) yields

$$V_0^{n+1}=V_0^n+2\nu\mu(V_1^n-V_0^n) \tag{3.6}$$

for the left boundary. In summary, the full order DNS data can be obtained by solving (via explicit time stepping) the linear system of difference equations (DE)

$$\begin{cases} \text{DE:} & V_j^{n+1}=V_j^n-\frac{\lambda}{2}(V_j^n)(V_{j+1}^n-V_{j-1}^n)+\nu\mu(V_{j+1}^n-2V_j^n+V_{j-1}^n)+kf_j^n \\ \text{BC:} & V_0^{n+1}=V_0^n+2\nu\mu(V_1^n-V_0^n) \\ & V_{N_x}^{n+1}=0 \\ \text{IC:} & V_j^0=\psi_1(x_j)=\sqrt{\frac{2}{3}}(1+\cos(\pi x_j)). \end{cases} \tag{3.7}$$

The last step is to discretize the delta distribution in the control term $f(x,t)=v(t)\delta(x-a)$. Using the previously developed notation, we set $f_j^n=v(t^n)\delta(x_j-a)$. Computation of accurate numerical delta functions has been explored in [70]. We implement the smoothed 4-point cosine discrete delta from [70] which satisfies the discrete version of Definition 2.14:

$$\delta_D(r)=\begin{cases} \frac{1}{4\pi h}\left(\pi+2\sin(\frac{\pi}{4}(2r+1))-2\sin(\frac{\pi}{4}(2r-1))\right), & |r|\leq 1.5 \\ -\frac{1}{8\pi h}\left(-5\pi+2\pi\left|r\right|+4\sin(\frac{\pi}{4}(2\left|r\right|-1))\right), & 1.5<|r|\leq 2.5 \\ 0, & 2.5<|r| \end{cases} \tag{3.8}$$

where $r=\frac{x-a}{h}$. This smooth version has higher regularity and spreads the input forcing more effectively than does a more standard sharp delta approximation.

3.2 Reference Solution

The reference solution to Eqn. (2.13) with no control input (i.e., $v(t) = 0$) displays the formation of a steep shock front that develops in time and approaches the right boundary (Fig. 3.1). This nonlinear behavior can be interpreted from the transport equation. The inviscid Burgers' equation (2.1) is in fact a transport equation with nonlinear flux. The wave speed of the solution is the value of the solution itself, $u(x, t)$. Hence, higher amplitudes of the wave propagate to the right faster (if u is positive) which leads to the observed steepening effect. We emphasize that the solution in Fig. 3.1 corresponds to the uncontrolled flow, that is, $v(t) = 0$. This is in stark contrast to the forced systems in the following sections with nonzero pointwise control inputs added to the PDE, in which the flow is sharply disrupted at the actuator location (yet mathematically is still differentiable due to the viscous term $\nu\Delta u$).

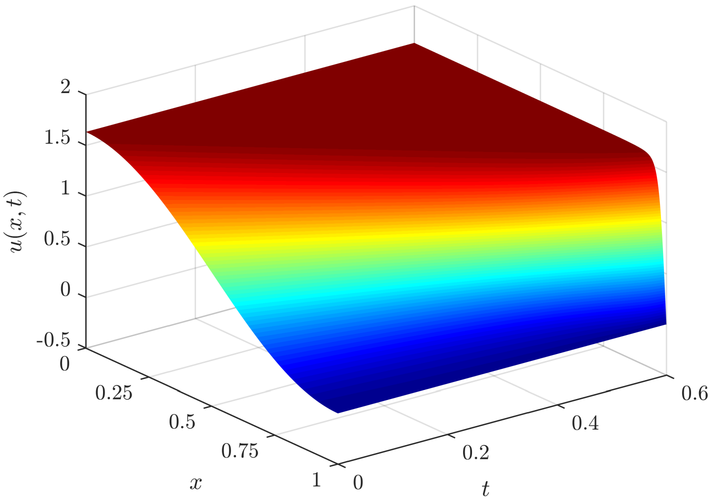


Figure 3.1: Full order reference solution without control, $v(t) = 0$.

3.3 Control Algorithm

The main objective in this work is to drive the nonlinear dynamical system governed by Burgers' equation (2.13) as close as possible into a preselected target spatial profile $u_T(x)$ at the final time $T > 0$. The system has a natural state that must be overwritten by the pointwise forcing $v(t)\delta(x - a)$ in order to control the state variable u such that the state at time $t = T$, $u(x, T)$, closely tracks the target function $u_T(x)$.

For example, the final state reached by the natural evolution of the Burgers' system (2.13) is a shock-like front seen in Fig. 3.2. This spatial profile is naturally reached by the state variable in the absence of any control input (i.e., the homogeneous PDE). The goal now is to prescribe some *other* target state for the solution to approach via a nonzero control input $v(t)$. However, this problem is infeasible without reducing the dimensionality of the control term itself, because the function v resides in some suitable *infinite dimensional* function space, which of course is computationally impractical to search within. Our approach follows closely that of [26] in that we parametrize the control function in terms of a *finite set* of searchable parameters. The explicit control input parametrization is a truncated Fourier series of the form

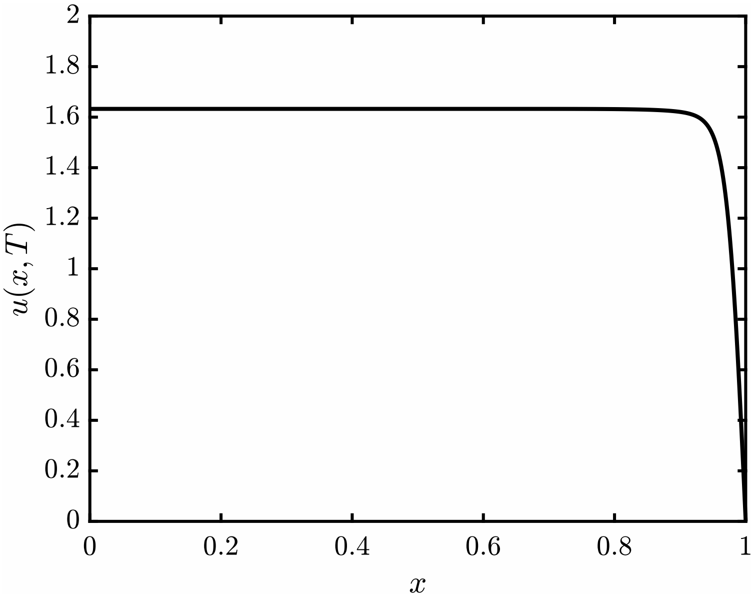


Figure 3.2: Natural final state $u(x,T)$ reached by u with $v(t) = 0$.

$$v_M(t) = \sum_{k=1}^{M} (\alpha_k \cos(\mu_k t) + \beta_k \sin(\mu t)) \approx v(t) \tag{3.9}$$

where $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$, $\beta = (\beta_1, \beta_2, \dots, \beta_k)$, $\mu = (\mu_1, \mu_2, \dots, \mu_k)$ are column vectors of unknown parameters and $M \in \mathbb{N}$. We write $v_M(t) = v_M(\alpha, \beta, \mu, t)$ to emphasize the dependence of v_M on the expansion coefficients. Therefore, the original infinite dimensional optimization problem has now been reduced to searching the finite dimensional vector space $\mathbb{R}^{3M}$ (which has dimension $3M$).

We choose the Fourier expansion because it is well known that sine or cosine are mutually orthogonal and form a basis of the space $L^2(\Omega)$, where $\Omega \subseteq \mathbb{R}$, which consists of the functions that are *square integrable* and is thus a suitable space to contain the control function v:

$$f \in L^2(\Omega) \quad \text{if} \quad \int_\Omega |f|^2 \, dx < \infty. \tag{3.10}$$

The space $L^2(0,1)$ is a Banach space and in particular a *Hilbert* space, equipped with the inner product

$$(f,g) = \int_0^1 f(x)g(x)dx \tag{3.11}$$

and norm

$$\|f\|_{L^2(0,1)} = (f,f)^{\frac{1}{2}} = \left(\int_0^1 |f(x)|^2 \, dx\right)^{\frac{1}{2}} \tag{3.12}$$

for any $f, g \in L^2(0,1)$. With these definitions, we can now make rigorous the measure of distance between the final state and the target state in the form of a **discrete cost functional**

$$J_N = \frac{1}{2} \|u_N(x,T) - u_T(x)\|^2_{L^2(0,1)} + \frac{\gamma}{2} \|v_M(\alpha,\beta,\mu,t)\|^2_{L^2(0,T)}, \tag{3.13}$$

where $\gamma > 0$ is the control weight, $u_T(x)$ is the desired target state function at time T, and N denotes the number of modes utilized if a reduced order model is used to compute the numerical solution $u_N(x,t)$. The first term in Eqn. (3.13) represents the distance between actual and target final states. A secondary consideration that is more practicable is embedded in the second term of Eqn. (3.13). This term is a measure of the "energy" of the control input; by weighting the energy or control effort with the small constant γ, we ensure that the control input does not grow too large.

By minimizing J_N with an optimal control v_M^{opt}, we will have obtained the best possible numerical solution at the final state $u_N(x,T)$ that best matches the target state $u_T(x)$. This, in essence, is an exercise in nonlinear mathematical programming. Hence, the formal control problem reads:

$$\begin{aligned} &\min_{(\alpha,\beta,\mu)\in\mathbb{R}^{3M}} \quad J_N(v_M(\alpha,\beta,\mu,t)) \\ &\text{subject to} \quad \begin{cases} u_t + uu_x - \nu u_{xx} = v_M(t)\delta(x-a), & (x,t) \in (0,1)\times(0,T] \\ u_x(0,t) = u(1,t) = 0, \quad t \in (0,T] \\ u(x,0) = \psi_1(x) = \sqrt{\frac{2}{3}}(1+\cos(\pi x)), \quad x \in (0,1) \end{cases} \end{aligned} \tag{3.14}$$

We propose the control algorithm (Alg. 1) to solve the open loop PDE-constrained optimal control problem (3.14). This algorithm is used for the full order model and both ROMs that are introduced in Chapter 4. The Nelder-Mead simplex method [see 32] in the Matlab function `fminsearch` is used to minimize the cost functional J in each loop of Alg. (1).

Algorithm 1 Optimal control loop: minimize the cost functional J

```
% INITIALIZE:
Define initial guess $X_0$, $M$ Fourier series terms, stopping criteria $\epsilon$, max loops $L$
Define $u_T(x)$
if ROM then
    Define $N$ modes
end if
$C = 1$
$J_{old} = J_N(X_0)$
$X_{old} = \texttt{fminsearch}(J_N(v_M(X_0)))$
$J_{new} = J_N(X_{old})$
% MAIN LOOP:
while $\left|\frac{J_{new}}{J_{old}} - 1\right| > \epsilon$ do
    if $C \geq L$ then
        break
    else
        $J_{old} = J_{new}$
        $X_{temp} = \texttt{fminsearch}(J_N(v_M(X_{old})))$
        $J_{new} = J_N(X_{temp})$
        $X_{old} = X_{temp}$
        $C = C + 1$
    end if
end while
```

3.4 Full Order Model Results

We first examine the sensitivity of our Burgers' system (2.13) to changes in model input parameters. Specifically, the effect of actuator location $x = a$ is examined in comparison to a previous study. Dean and Gubernatis [17] show that even with a full order model (FOM), it is hard to control *upstream* flow with a pointwise (local) control input; this is exactly the behavior we observe in full order finite difference simulations. Relatedly, Dean and Gubernatis perform an approach with two control points. Using the same initial condition, boundary condition, and forcing term as in their work [17], we observe the effect of the two control points spreading in time [Fig. 3.3]. If the control point is close enough to the boundary (in Fig. 3.4a it is $a = 0.1$), then the Neumann condition does indeed allow for the solution to shift up or down at the left boundary. This behavior was not observed when using the initial condition Ψ_1 from [26] in our main system Eqn. (2.13) and supports the claim that pointwise control has a strong local influence on the solution to PDE.

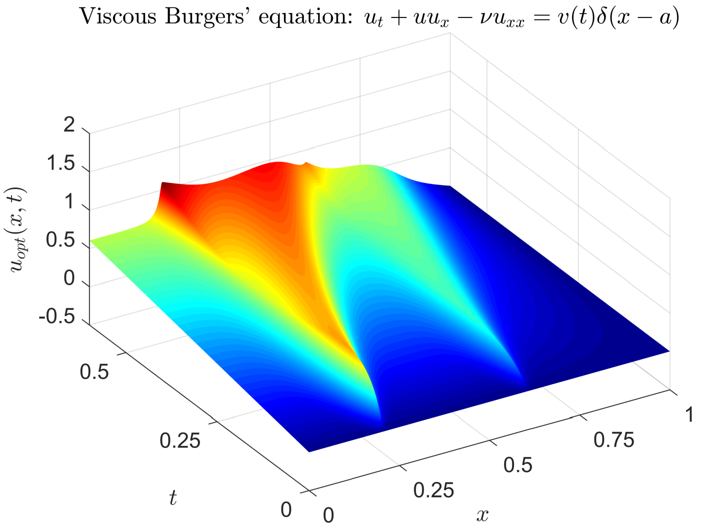
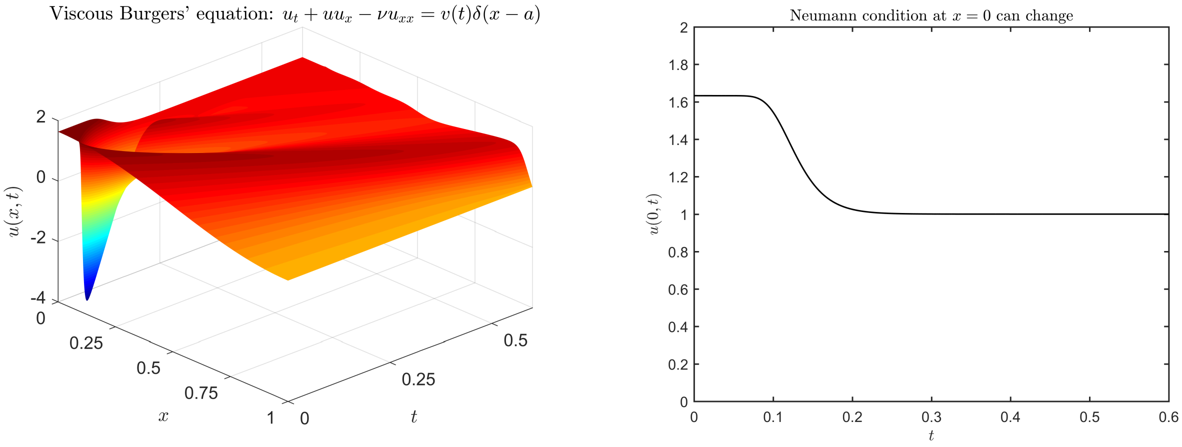


Figure 3.3: Example of multiple control inputs ($a_1 = 0.2$ and $a_2 = 0.6$).

We now present the results for six different target state functions (test cases) in the FOM control simulation. Of course for this FOM, it is meaningless to specify the number of modes N; the solution is hence denoted simply as $u = u(x,t)$. We use $h = \Delta x = \frac{1}{500}, T = 0.6, a = 0.6, \gamma = 10^{-3}, \epsilon = 10^{-3}, \nu = 10^{-2}, M = 2$ terms, and an initial guess vector of $\mathbf{X_0} = \mathbf{0}$. All of our optimization routinesin this thesis use a stopping criteria

$$\left| \frac{J^{n+1} - J^n}{J^n} \right| < \epsilon = 10^{-3}$$



(a) Full flow field with changing boundary at $x = 0$ (b) Time evolution of the Neumann boundary

Figure 3.4: The Neumann left boundary adjusts $u(x,t)$ when the control input location is nearby.

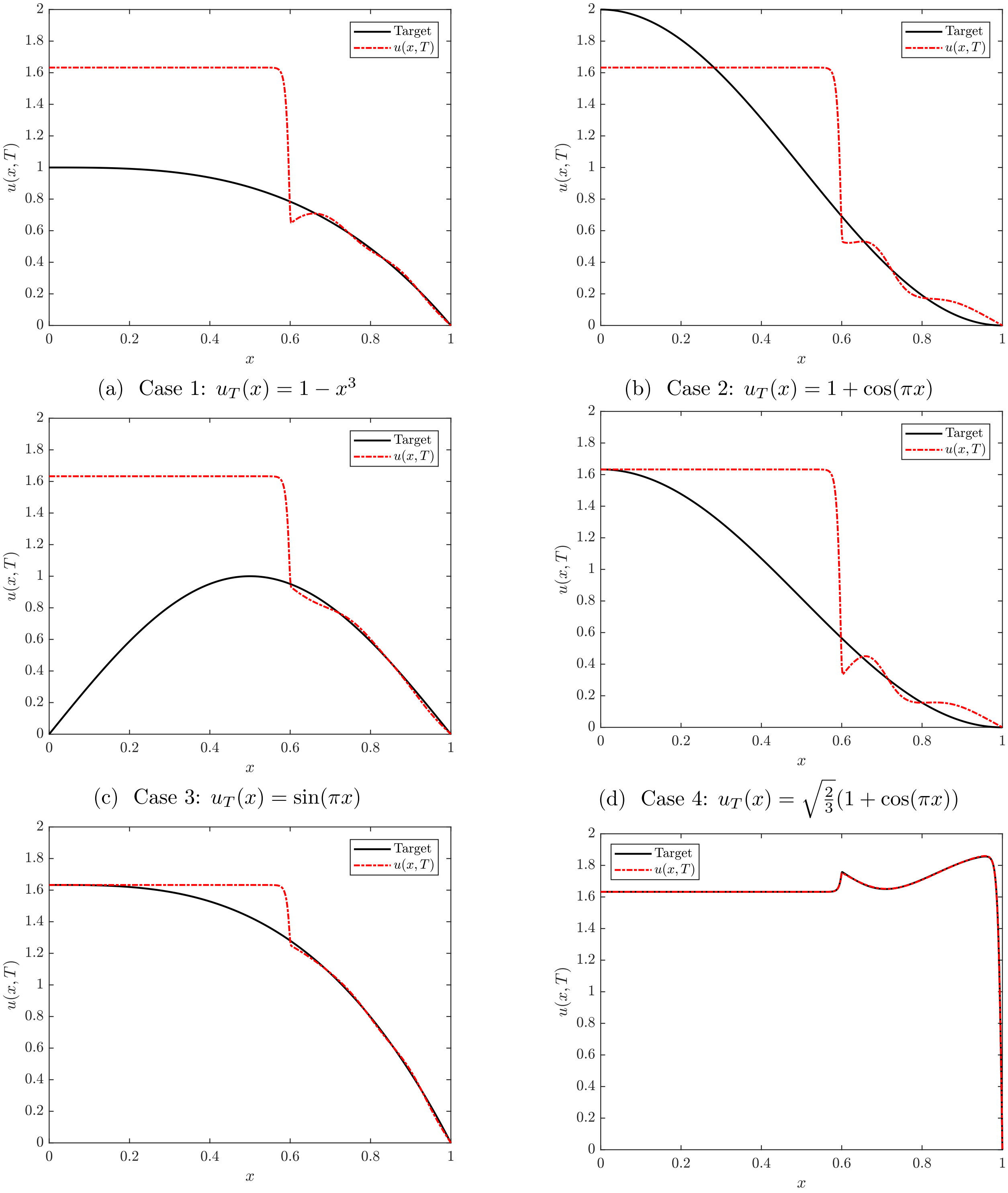
to approximate the continuous convergence condition $\left|\frac{J^{n+1}-J^n}{J^n}\right|$ as $n \to \infty$, where J^{n+1} denotes the current value of the cost functional and J^n the previous value.

FOM Experiment	$J(X^{opt})$	Number of Loops
Case 1: $u_T = 1 - x^3$	0.138854	3
Case 2: $u_T = 1 + \cos(\pi x)$	0.050898	2
Case 3: $u_T = \sin(\pi x)$	0.290653	2
Case 4: $u_T = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$	0.076629	3
Case 5: $u_T = 2\sqrt{\frac{2}{3}}(1 - x^3)$	0.004855	3
Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$	2.216697E-5	6

Table 3.1: FOM values of the cost functional at the optimal states for six different u_T.

Table 3.1 shows the final decayed value of the cost function $J = J_N$ evaluated at the optimal control input vector of parameters X^{opt} and the number of control loops required to achieve the stopping criteria for each of these six test cases. In general, smaller values of J suggest better performance of the numerical solution approaching the target state u_T.

The optimally controlled FOM with actuator located at $x = a = 0.6$ performs as one would expect. That is, the final state of the solution $u(x,T)$ quickly snaps to the prescribed target function u_T at $x = 0.6$ and closely matches u_T up to the right boundary $x = 1$. This behavior has been observed in other sources [see 17]. All six of the cases satisfy the boundary conditions of Eqn. (2.13) except Case 3, $u_T(x) = \sin(\pi x)$. However, the FOM in this case still follows the target profile well for values of x after $x = 0.6$, even though the value of J is the highest of the six cases. Such a desired



(a) Case 1: $u_T(x) = 1 - x^3$

(b) Case 2: $u_T(x) = 1 + \cos(\pi x)$

(c) Case 3: $u_T(x) = \sin(\pi x)$

(d) Case 4: $u_T(x) = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$

(e) Case 5: $u_T(x) = 2\sqrt{\frac{2}{3}}(1 - x^3)$

(f) Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$

Figure 3.5: FOM comparison of final state $u(x,T)$ to target state $u_T(x)$.

target function may be more realistic in applications, so it is encouraging to see this successful behavior. Full convergence histories and flow fields may be found in Appendix A, Figs. A.2–A.3.

In Case 4, the target function is equal to the initial condition, $u_T = \psi_1$. The optimal control was not able to drive the dynamical system back into its initial state for our Burgers' problem. Further, the cubic target function in Case 5 matches the value of the initial condition at the point $x = 0$, and performance is improved from the similar target state in Case 1 (Fig. 3.5a).

The Case 6 target function (Fig. 3.5f) requires some additional explanation. Instead of prescribing an explicit target that may or may not even be **reachable** by the PDE in Eqn. (2.13), we instead create an achievable target u_T^{reach} by explicitly inputting a *known* control function, here $v_p(t) = 0.25(1 + \sin(4\pi t/T))$. The reachable target is computed numerically by solving the FOM for the Burgers' system Eqn. (2.13) with $v = v_p$ known and setting $u_T^{reach} := u(x, T)$. Then the control algorithm is initiated as before in the other five cases to solve for the optimal $v^{opt}(t)$ that drives u to the reachable state u_T^{reach} (see Fig. A.1 for a comparison of the prescribed control v_p and the optimal control v^{opt} for Case 6). This process was conducted before in [40] and serves as a more effective means of comparing FOM results with ROM results, as we shall see in Section 4.3. As expected from the theory and existing literature, the FOM can exactly achieve the Case 6 target state as evidenced in Fig. 3.5f; the plots are indistinguishable.

Chapter 4

Reduced Order Models

With the full order model (FOM) at our disposal, we may now construct reduced order models (ROMs) to solve the same optimal control problem. We define two ROMs, the Fourier–ROM and the POD–G–ROM based on the Galerkin projection. The Fourier-type basis functions begin our discussion.

4.1 Fourier Basis

The Fourier basis functions are a sequence of sinusoids that only depend on boundary conditions; their construction did not involve any prior information about the dynamical system (Eqn. 2.13) itself. We shall see how this contrasts with the Proper Orthogonal Decomposition in later sections.

We begin with the sequence of functions that arise from a modified Fourier series:

$$\{\Psi_n(x)\}_{n=1}^{\infty} = \{\cos(n\pi x) - (-1)^n\}_{n=1}^{\infty}. \tag{4.1}$$

Clearly, the Ψ_n satisfies the Neumann boundary condition $(\frac{\partial}{\partial x}\Psi_n)(0) = 0$ on the left and the Dirichlet boundary condition $\Psi_n(1) = 0$ on the right for the domain $\Omega = [0, 1]$ in Eqn. (2.13).

However, the functions Ψ_n are not orthonormal nor even orthogonal when equipped with the standard L^2 inner product

$$(f, g) = \int_0^1 f(x)g(x)dx \tag{4.2}$$

and norm

$$\|f\|_{L^2(0,1)} = \left(\int_0^1 |f(x)|^2 \, dx\right)^{\frac{1}{2}} \tag{4.3}$$

for any $f, g \in L^2(0, 1)$. A sequence of elements φ_j in an inner product space are orthonormal if and only if they satisfy

$$(\varphi_j, \varphi_k) = \begin{cases} 1, & \text{if} \quad k = j \\ 0, & \text{if} \quad k \neq j \end{cases} \quad = \quad \delta_{jk} \tag{4.4}$$

for any j, k and where δ_{jk} is the Kronecker delta. Hence, we apply the well-known Gram-Schmidt orthogonalization process from linear algebra to the functions $\{\Psi_n\}_{n=1}^{\infty}$ to form an orthonormal basis for the infinite dimensional inner product space $L^2(0,1)$. The **Fourier basis** is then

$$\psi_n(x) = \sqrt{\frac{4n-2}{2n+1}} \left(\frac{(-1)^{n+1}}{2n-1} + \cos(n\pi x) + \frac{2}{2n-1} \sum_{k=1}^{n-1} (-1)^{n+k+1} \cos(k\pi x) \right), \quad n = 1, 2, \ldots \tag{4.5}$$

For practical computation, we truncate this sequence of modes to a finite number $N \in \mathbb{N}$. We will apply this Fourier basis in a reduced order model using the Galerkin projection, detailed in the forthcoming Section 4.2.1.

The first four Fourier basis functions for the problem (2.13) are plotted in Fig. 4.1. We indeed observe that the boundary conditions are satisfied and that higher indices j in ψ_j correspond to higher frequencies. It is clear that while the Fourier–ROM is easy to implement, it can be shown that this model requires a high number of modes for reasonable accuracy in approximating the true solution to the PDE. This contrasts to the POD approach, detailed next.

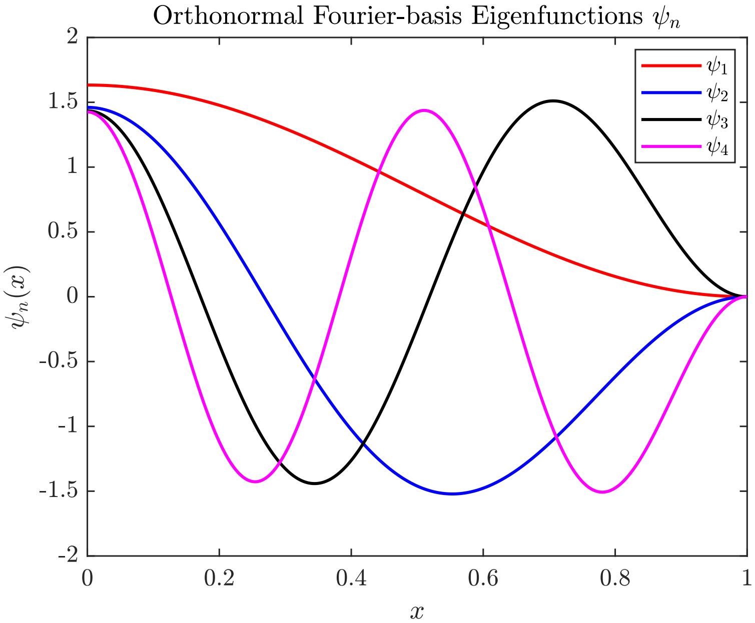


Figure 4.1: The first four orthonormal basis functions used to construct the Fourier–ROM.

4.2 Proper Orthogonal Decomposition Basis

Proper Orthogonal Decomposition (POD) is a popular data analysis technique based on the *singular value decomposition* (SVD) from numerical linear algebra that has recently arisen to prominence in a variety of fields. In this section, we detail the POD approach for reduced order modeling to capture unsteady, convective, and non-stationary dynamics of governing nonlinear PDE in fluid dynamics. San and Iliescu in [54] provide a far more detailed discussion of this topic. For clarity, POD is also known as the principle component analysis, Karhunen-Loéve decomposition, or the empirical orthogonal functions [55].

The POD basis functions are pre-computed from DNS snapshots or experimental measurements and are dependent on the parameters of the problem. This is the so-called *method of snapshots* [48] and explains why POD is known as a *data-driven* method. In this way, the basis inherently captures the dominant flow characteristics (e.g., *coherent structures*) that compose the flow field. We outline the POD in the one-dimensional case.

Suppose we have a data set $u(x,t)$. The spatial variable $x \in \Omega$ lives in a bounded subset $\Omega \subset \mathbb{R}$ and the time variable t is nonnegative and bounded, $0 \leq t \leq T$. The POD procedure begins by decomposing the high-dimensional flow field u into a mean part $\bar{u}$ and a fluctuating part $\tilde{u}$,

$$u(x,t) = \bar{u}(x) + \tilde{u}(x,t), \tag{4.6}$$

where the mean is constant in time and is given by

$$\bar{u}(x) = \frac{1}{N}\sum_{i=1}^{N} u(x,t_i), \tag{4.7}$$

and N corresponds to a finite number of distinct snapshots at time t_i used for time averaging. This decomposition often arises when there is a need to model Reynold's stresses in an incompressible fluid flow.

The goal now is to then derive a set of N bases $\phi_k(x),\ k = 1, 2, \ldots, N$, which can reproduce the fluctuating component of the above decomposition (4.6) by the modal expansion

$$\tilde{u}(x,t) = \sum_{k=1}^{N} a_k(t)\phi_k(x), \tag{4.8}$$

where $a_k(t)$ are the time-dependent POD coefficients. A mapping to the subspace spanned by the truncated set of basis functions $\{\phi_k\}_{k=1}^{R}$ with $R \leq N$ will lead to a reduced system of ODE that aim to capture the spatial and temporal evolution of the original PDE.

In this study, snapshots are sampled uniformly from the solution time interval $[0, T]$. In general, the number and location of the snapshots $u(x, t_i)$ are dependent on the dynamics of the system and may vary depending on the situation. We note that the decomposition (4.6) unnecessarily complicates our ROM (specifically, the coefficients in the ODE system) by inducing an affine projection instead of a linear projection. While our code has the capability for this particular velocity decomposition, we run all simulations without this feature active by artificially setting $\bar{u} = 0$ (which implies $u = \tilde{u}$).

4.2.1 Construction of POD basis

A POD can be constructed from the scalar field $u(x, t)$ at different time points, or snapshots. These snapshots are either obtained by solving the governing equations through a DNS, computing an exact solution, or processing experimental data. In the following, we will utilize the superscript i to indicate a particular snapshot in time. For the POD approach, we utilize a total of N snapshots for the field variable, i.e., $u(x, t_i) = u^i(x)$ for $i = 1, 2, \ldots, N$. A correlation (or autocorrelation) matrix $C \in \mathbb{C}^{N \times N}$ is constructed using the fluctuating components of the snapshots, $\tilde{u}(x, t_i) = \tilde{u}^i(x)$, and is defined as

$$C_{ij} = \int_\Omega \tilde{u}^i(x) \tilde{u}^j(x) dx, \tag{4.9}$$

where $i, j \in \{1, 2, \ldots, N\}$ refer to the i^{th} and j^{th} snapshots. The correlation matrix C is a non-negative symmetric square matrix of size N and each entry may be expressed as the L^2 inner product $C_{ij} = (\tilde{u}^i, \tilde{u}^j)$. Clearly, this computation of N^2 integrals is one indication that the offline stage of ROM formation is computationally expensive.

In this study, we use a trapezoidal integration rule for the numerical computation of all required L^2 inner products. The optimal POD basis functions are obtained by performing an eigendecomposition for the C matrix. This has been shown in detail in POD literature (see, e.g., [14]) and is closely related to the singular value decomposition [31]. The eigenvalue problem can be expressed as

$$CW = W\Lambda\,, \tag{4.10}$$

where $\Lambda = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_N]$ is a diagonal matrix containing the eigenvalues (*singular values*) of this decomposition and $W = [w^1, w^2, \ldots, w^N]$ is an orthogonal matrix consisting of the corresponding eigenvectors. The eigenvalues are stored from greatest to least magnitude, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$, and this, in a sense, ranks the "energy" of each POD mode. The orthogonal POD basis functions can then be obtained from the matrix of eigenvectors as the summation

$$\phi_k(x) = \sum_{i=1}^{N} w_i^k \tilde{u}(x, t_i), \quad k = 1, 2, \ldots, N\,, \tag{4.11}$$

where w_i^k is the i^{th} component of eigenvector w^k. The eigenvectors must also be normalized in order to satisfy the condition of orthonormality between bases given by Eqn. (4.4). For this relationship hold for the POD basis, the eigenvector w^j must satisfy

$$\sum_{i=1}^{N} w_i^j w_i^j = \frac{1}{\lambda_j}. \tag{4.12}$$

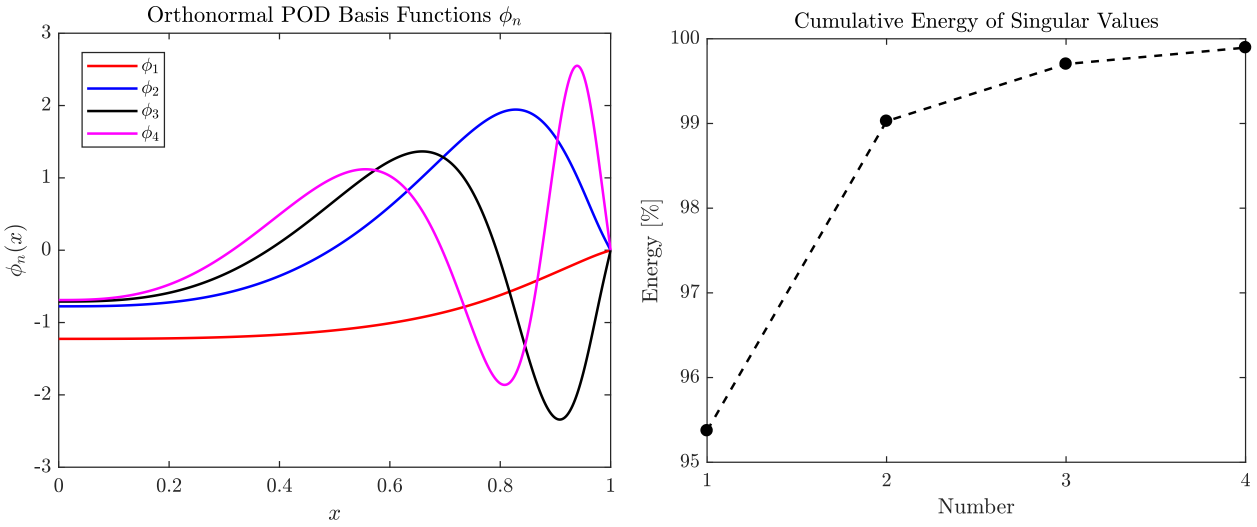
In practice, most modern packages for solving the eigensystem given by Eqn. (4.10) return the eigenvector matrix W with each eigenvector column already normalized to unity (e.g., MATLAB). In this case, the ***orthonormal* POD basis** is given by

$$\phi_j(x) = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{N} w_i^j \tilde{u}(x, t_i), \quad j = 1, 2, \ldots, N\,, \tag{4.13}$$

where $\phi_j(x)$ is the j^{th} POD basis function.

The main motivation behind the construction of a ROM using the optimal POD basis is due to the fact that modes with high magnitude eigenvalues retain a greater proportion of the "modal energy" of the system. To reduce the computational expense of our system in the transformed space, we truncate the number of modes in our system to a value $R < N$ with the assumption that these R modes capture a large fraction of the total energy of the system (ranked by the singular values). Hence, it is possible to construct a lower degree of freedom approximation of our PDE in the subspace spanned by the POD modes that include the most dominant characteristic dynamics.

We apply this procedure to Eqn. (2.13) to obtain the POD basis (4.13). In Fig. 4.2a, these functions are plotted on $\Omega = [0, 1]$. We observe how they are heavily weighted near the right boundary $x = 1$, which is near the location of the high gradient shock front (recall Fig. 3.1). For our posed problem, the first two singular values corresponding to the first two POD basis functions comprise nearly 99% of the total modal energy in the system (Fig. 4.2b) . However, in contrast to a linear system, the quadratic nonlinearity in Burgers' equation ensures that a two-mode approximation will most likely not be sufficient with respect to numerical and physical accuracy. The high index (e.g., $R \gg 50$) dissipative truncated modes have significant influence on the accuracy of the ROM, and this is the motivation for a variety of new closure models [see, e.g., 54].



(a) First four POD basis functions used to construct the POD–G–ROM

(b) Cumulative modal energy of the first four dominant singular values

Figure 4.2: The first four orthonormal POD basis functions and corresponding singular value energies.

4.3 Galerkin Projection

The Galerkin method uses a generalized Fourier series approach to recast a given PDE into a system of ODE. Given an orthogonal basis $\{\varphi_n\}_{n=1}^{\infty}$ of a function space, say the square integrable functions $L^2(\Omega)$, each element $v \in L^2(\Omega)$ can be uniquely expanded in terms of this basis as

$$v = \sum_{n=1}^{\infty} a_n \varphi_n,$$

where the coefficients a_n can be obtained by the projection

$$a_n = \frac{(v, \varphi_n)}{(\varphi_n, \varphi_n)} = \frac{\displaystyle\int_{\Omega} v \varphi_n dx}{\displaystyle\int_{\Omega} |\varphi_n|^2 \, dx}, \quad n = 1, 2, \ldots$$

using the standard L^2 inner product defined in Eqn. (4.2). If $v = v(x,t)$ is a function of space and time as in the case of fluid flow data, then the functions $a_n = a_n(t)$ evolve in time (assuming $\varphi_n = \varphi_n(x)$ are spatial modes), analogous to a separation of variables.

With the Galerkin projection, we truncate the state variable expansion to a finite number of modes $R < \infty$. Hence, we obtain a sequence of functions

$$v_R(x,t) = \sum_{n=1}^{R} a_n(t)\varphi_n(x)$$

that reside in the new finite dimensional vector space U_R that approximates the original infinite dimensional function space $L^2(\Omega)$; that is, $U_R \to L^2(\Omega)$ and $v_R \to v$ as $R \to \infty$.

Completion of the Galerkin procedure requires an elimination of the spatial dependence in the underlying PDE. To do this, the PDE is cast into its weak (*variational*) form by multiplying both sides by a test function $\omega \in L^2(\Omega)$ and integrating over the spatial domain Ω. This is equivalent to taking the inner product on both sides of the equation with the test function ω. In particular, since $U_R \subset L^2(\Omega)$, we choose $\omega := \varphi_k \in U_R$ as the specific test function to take advantage of the orthogonality of the functions φ_k. If the basis functions φ_k are finite elements (e.g., linear piecewise functions), then an integration by parts is usually performed to shift derivatives to these modes.

For the Burgers' equation (2.13), the Galerkin projection in one space dimension can be carried out in the following manner. We first recall the Burgers' system in its conservation form

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{1}{2}u^2\right) - \nu\dfrac{\partial^2 u}{\partial x^2} = v(t)\delta(x-a), \quad (x,t) \in (0,1)\times(0,T] \\ \dfrac{\partial u}{\partial x}(0,t) = u(1,t) = 0, \quad t \in (0,T] \\ u(x,0) = \psi_1(x) = \sqrt{\tfrac{2}{3}}(1+\cos(\pi x)), \quad x \in (0,1). \end{cases} \tag{4.14}$$

To derive the Galerkin projection as detailed above, we rewrite Burgers' equation (4.14) as

$$\frac{\partial u}{\partial t} = F + L[u] + N[u;u]\,, \tag{4.15}$$

where $F = F(x,t) = v(t)\delta(x-a)$ is the control input, L is the linear diffusion operator

$$L[f] = \nu\left(\frac{\partial^2}{\partial x^2}f\right)\,, \tag{4.16}$$

and N is the bi-nonlinear operator

$$N[f;g] = -\frac{\partial}{\partial x}\left(\frac{1}{2}fg\right) \tag{4.17}$$

that represents the advection term in conservative form. By applying the Galerkin projection to

Eqn. (4.15), we obtain the system of first-order ODE for each temporal coefficient $a_k(t)$ as our ROM:

$$\left(\frac{\partial u}{\partial t}, \phi_k\right) = (F, \phi_k) + (L[u], \phi_k) + (N[u;u], \phi_k), \quad \text{for} \quad k = 1, 2, ..., R. \tag{4.18}$$

Given the flow decomposition in Eqn. (4.6), we expand the fluctuating component of the field variable u in terms of an *orthonormal* basis ϕ_n truncated to R modes,

$$u(x,t) \approx u_R(x,t) = \bar{u}(x) + \sum_{n=1}^{R} a_n(t)\phi_n(x)\,, \tag{4.19}$$

where a_n are time dependent *modal coefficients*. By using the properties of the orthonormal basis (Eqn. 4.4), we simplify Eqn. (4.18) to obtain the final form of the ROM:

$$\frac{da_k}{dt} = v\phi_k(a) + B_k + \sum_{i=1}^{R} \mathfrak{L}_{ik} a_i + \sum_{i=1}^{R}\sum_{j=1}^{R} \mathcal{N}_{ijk} a_i a_j, \quad \text{for} \quad k = 1, 2, ..., R\,, \tag{4.20}$$

where

$$B_k = (L[\bar{u}], \phi_k) + (N[\bar{u};\bar{u}], \phi_k)\,, \tag{4.21}$$

$$\mathfrak{L}_{ik} = (L[\phi_i], \phi_k) + (N[\bar{u};\phi_i] + N[\phi_i;\bar{u}], \phi_k)\,, \tag{4.22}$$

$$\mathcal{N}_{ijk} = (N[\phi_i;\phi_j], \phi_k) \tag{4.23}$$

are constants that may be precomputed offline before evolving the ROM in time. The initial condition for the k-th ODE is again given by projection:

$$a_k(0) = (u(x,0) - \bar{u}(x), \phi_k)\ , \tag{4.24}$$

where $u(x,0) = u_0(x)$ is the initial condition of the problem (4.14). This system of R coupled nonlinear ODE can be solved using any standard numerical method, such as Runge-Kutta (RK) finite difference integrators. In this study, we use the fourth-order RK scheme `ode45` in MATLAB.

We denote Eqns. (4.20) and (4.24) as **Fourier–ROM** when using the Fourier basis $\{\psi_n\}_{n=1}^{N}$ from Section 4.1 and **POD–G–ROM** when using the POD modes $\{\phi_j\}_{j=1}^{R}$ from Section 4.2. We emphasize that the number of degrees of freedom in the Burgers' system is now significantly lower than the dimension of the original FOM. The vectors, matrices, and tensors in Eqns. (4.21)–(4.23) are also precomputed quantities, which results in a dynamical system that can be solved very efficiently.

4.4 Reduced Order Model Results

4.4.1 Fourier–ROM

We now proceed to show how the Fourier reduced order model (Fourier–ROM) performed for the same six test cases of the target state $u_T(x)$ as in Section 3.3. We use $h = \Delta x = \frac{1}{1000}$, $N = 4$ Fourier modes, $T = 0.6, a = 0.6, \epsilon = 10^{-3}, \nu = 10^{-2}, M = 2$ terms for the control parametrization, and our initial state vector (a guess) of $\mathbf{X_0} = \mathbf{0}$.

Time integration is performed using the `ode45` solver in MATLAB and derivatives of the Fourier basis functions are computed analytically in Eqns. (4.25)–(4.26) since the exact expression for $\psi_n(x)$ is known (Eqn. 4.5).

$$\frac{\partial \psi_n}{\partial x}(x) = \sqrt{\frac{4n-2}{2n+1}}\left(-n\pi \sin(n\pi x) + \frac{2}{2n-1}\sum_{k=1}^{n-1}(-1)^{n+k+2} k\pi \sin(k\pi x)\right), \tag{4.25}$$

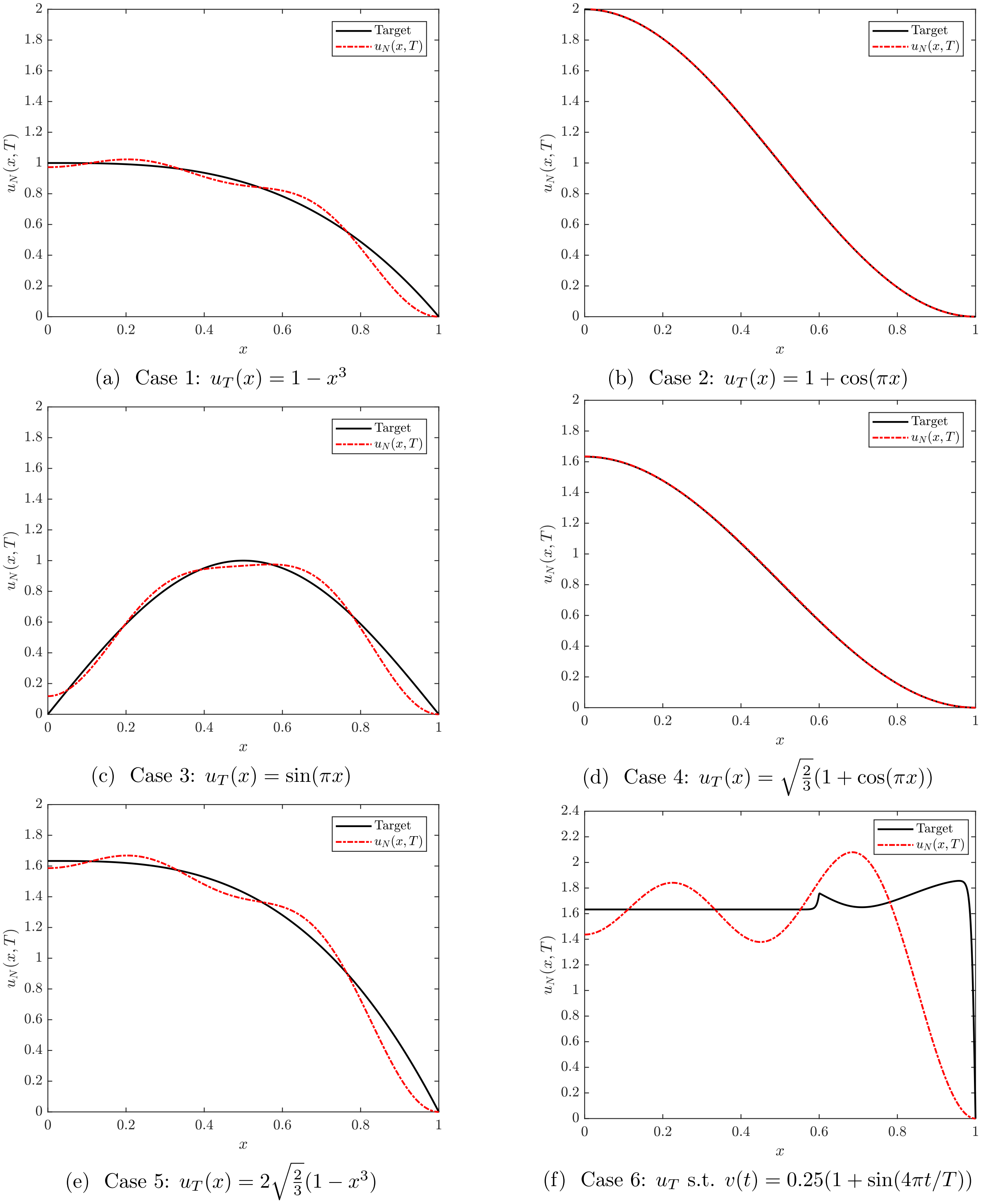
$$\frac{\partial^2 \psi_n}{\partial x^2}(x) = \sqrt{\frac{4n-2}{2n+1}}\left(-(n\pi)^2 \cos(n\pi x) + \frac{2}{2n-1}\sum_{k=1}^{n-1}(-1)^{n+k+2} (k\pi)^2 \cos(k\pi x)\right). \tag{4.26}$$

Fourier–ROM Experiment: $N = 4$ modes	$J_N(X^{opt})$	Number of Loops
Case 1: $u_T = 1 - x^3$	0.002229	4
Case 2: $u_T = 1 + \cos(\pi x)$	0.000424	3
Case 3: $u_T = \sin(\pi x)$	0.002709	3
Case 4: $u_T = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$	0.000722	5
Case 5: $u_T = 2\sqrt{\frac{2}{3}}(1 - x^3)$	0.003681	3
Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$	0.171887	3

Table 4.1: Fourier–ROM values of the cost functional at the optimal states for six different u_T.

Table 4.1 displays the minimum value of the discrete cost functional J_N for each of the six target functions. The rate of convergence to the optimal state is comparable to that of the FOM (Table 3.1), at least in terms of the number of algorithm loops required. A full convergence history may be found in Appendix A in Fig. A.5. In fact, the values of the cost function are indeed lower than the corresponding results yielded by the FOM. This is not surprising when examining the plots of the Fourier–ROM final states versus the target states in Fig. 4.3. In contrast to the FOM finals states observed in Fig. 3.5, the final states of the Fourier–ROM are *not constrained* to the profile of the natural state (Fig. 3.2) of the Burgers' system (2.13) prior to control actuation at $x = a = 0.6$. This is the reason why we observe lower values of J_N for the Fourier-ROM. In these four-mode reduced order models, the solution evolves dynamically on the *whole domain* Ω due to the control input and sinusoidal basis. This allows the final state to track the target function even for values of $x < a$. The reader is referred to Fig. A.4 in Appendix A for a full picture of the ROM flow fields in the x–t plane.

Cases 1–3 were originally presented in the work of Kucuk and Sadek in [26], and our implementation of the Fourier–ROM framework reproduces the same results. To explore the flexibility of the method, we additionally test the Fourier–ROM on Cases 4–6. Our results suggest that while the model can accurately track the target state for more arbitrary, sinusoidal target functions that may or may not satisfy the boundary conditions of the PDE, *reachable* target states such as Case 6 are not resolvable (at least with the four Fourier mode approximation), see Fig. 4.3f. Even an increase to ten modes did not improve the controllability to the level seen in Cases 1–5. If this can be definitively confirmed in future studies, then this deficiency will certainly call into question the effectiveness of the Fourier–ROM since any model should have the natural ability to optimize to a reachable target function. In the next section, we will see that the POD–G–ROM is superior in this respect.



(a) Case 1: $u_T(x) = 1 - x^3$

(b) Case 2: $u_T(x) = 1 + \cos(\pi x)$

(c) Case 3: $u_T(x) = \sin(\pi x)$

(d) Case 4: $u_T(x) = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$

(e) Case 5: $u_T(x) = 2\sqrt{\frac{2}{3}}(1 - x^3)$

(f) Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$

Figure 4.3: Fourier–ROM comparison of final state $u(x,T)$ to target state $u_T(x)$.

4.4.2 POD–G–ROM

We now proceed to show how the Proper Orthogonal Decomposition reduced order model with Galerkin projection (POD–G–ROM) performed for the same six test cases of the target state $u_T(x)$ and will draw comparisons to the other two control approaches (FOM and Fourier–ROM). Our numerical simulations use $h = \Delta x = \frac{1}{1000}$ to compute the DNS snapshot data for POD basis construction, $N = 1000$ snapshots, $R = 4$ POD modes, $T = 0.6, a = 0.6, \gamma = 10^{-3}, \epsilon = 10^{-3}, \nu = 10^{-2}, M = 2$ series terms for the control parametrization, and an initial guess vector $\mathbf{X_0} = \mathbf{0}$.

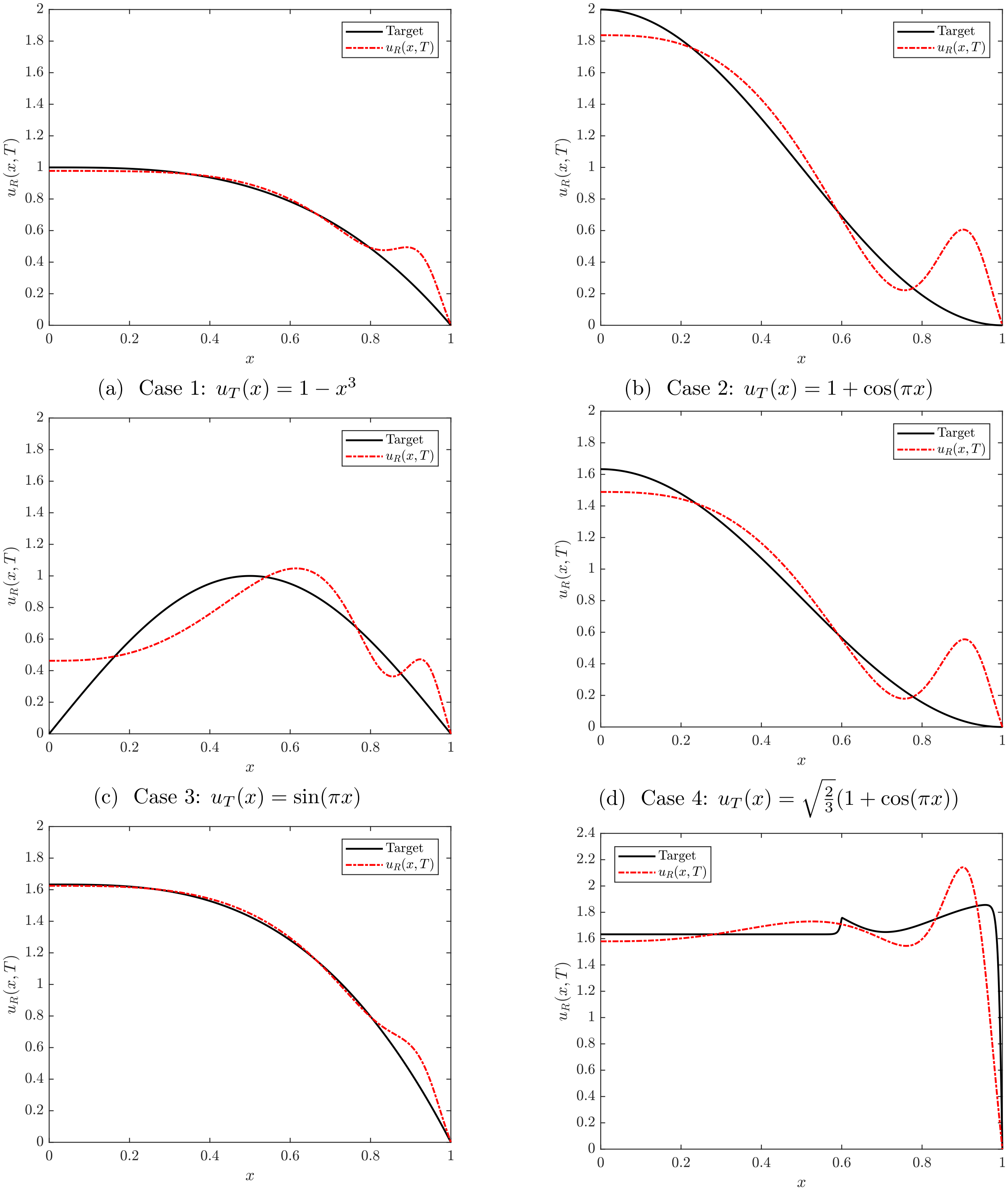
Time integration is again performed using the RK `ode45` integrator in Matlab, and derivatives of the POD basis functions ϕ_k are computed using second-order accurate built-in Matlab central difference schemes `gradient` and `del2` for the gradient (first derivative in 1D) and Laplacian (second derivative in 1D), respectively.

POD–G–ROM Experiment: $R = 4$ modes	$J_R(X^{opt})$	Number of Loops
Case 1: $u_T = 1 - x^3$	0.007793	5
Case 2: $u_T = 1 + \cos(\pi x)$	0.024874	6
Case 3: $u_T = \sin(\pi x)$	0.016572	4
Case 4: $u_T = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$	0.022411	6
Case 5: $u_T = 2\sqrt{\frac{2}{3}}(1 - x^3)$	0.005019	6
Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$	0.022308	3

Table 4.2: POD–G–ROM values of the cost functional at the optimal states for six different u_T.

The POD–G–ROM values of the discrete cost functional J_R (Table 4.2) are comparable to that of the FOM (Table 3.1) and are higher than the values produced by the Fourier–ROM (Table 4.1) except for the cubic polynomial targets in Case 1 and 5. One explanation for why the cubics performed better than the more general sinusoids of Case 2–4 is that the derivatives of the targets in Case 1 and 5 are non-positive, $\frac{\partial u_T}{\partial x} \leq 0$. This contrasts with the targets in Cases 2–4 which exhibit inflection points. The natural final state of the PDE satisfies $\left(\frac{\partial u}{\partial x}\right)(x,T) \leq 0$ (see Fig. 3.2) and this similarity may explain why the data-driven POD approach performs better for target functions that share this trait.

Table 4.2 also indicates the higher cost associated with the POD–G–ROM; most test cases required one to three more loops than corresponding cases for the FOM or Fourier–ROM to meet the same stopping criteria in Alg. (1), even though expensive set up calculations were precomputed offline before running the simulations. A full convergence history for each of the six test cases may be found in Appendix A, Fig. A.7. Perhaps future studies incorporating higher order differentiation schemes or more modes would better address the issue of computational efficiency of the POD–G–ROM.



(a) Case 1: $u_T(x) = 1 - x^3$

(b) Case 2: $u_T(x) = 1 + \cos(\pi x)$

(c) Case 3: $u_T(x) = \sin(\pi x)$

(d) Case 4: $u_T(x) = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$

(e) Case 5: $u_T(x) = 2\sqrt{\frac{2}{3}}(1 - x^3)$

(f) Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$

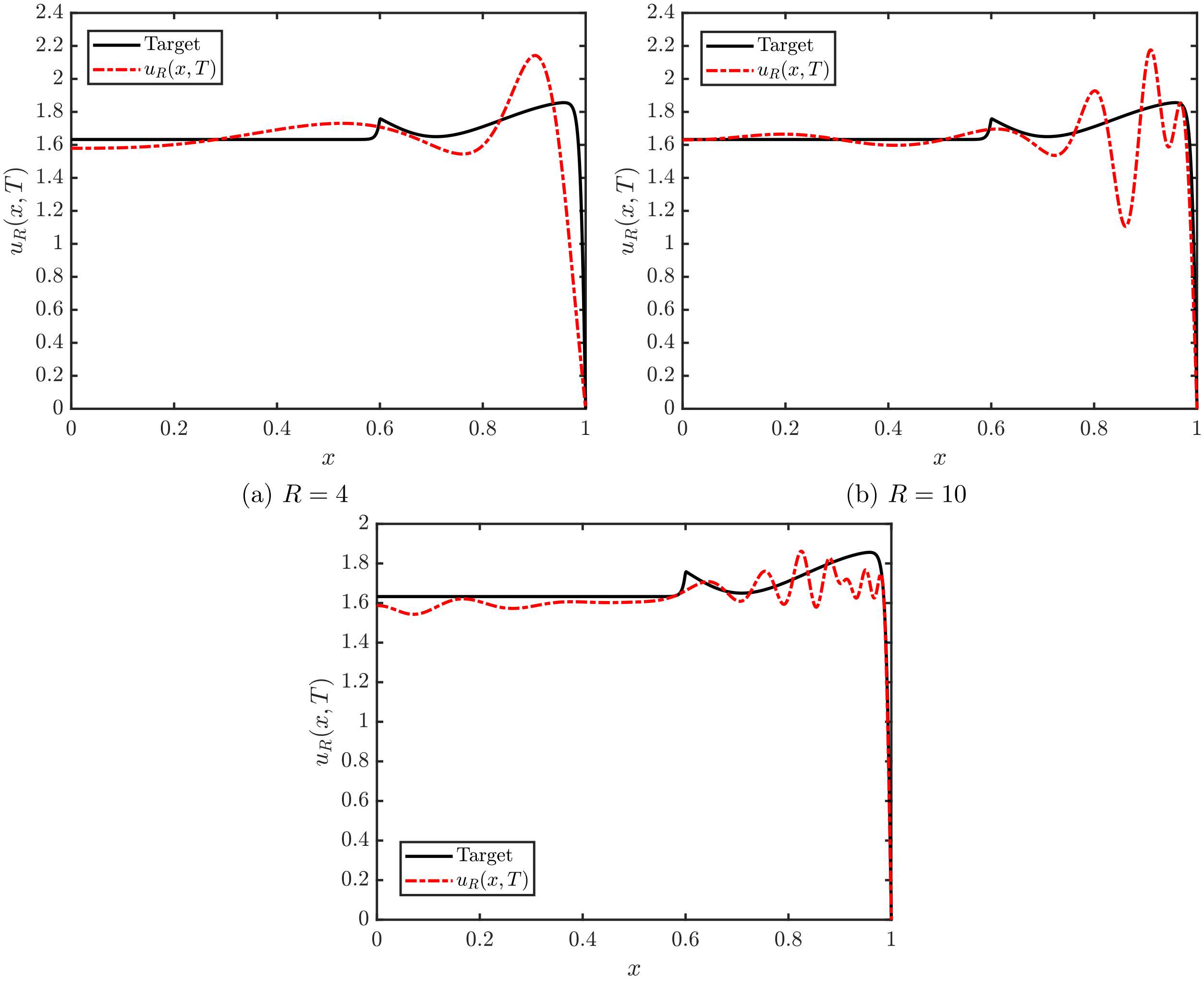
Figure 4.4: POD–G–ROM comparison of final state $u(x,T)$ to target state $u_T(x)$.

In terms of ability to track the target state, the controlled solutions $u_R(x,T)$ in Fig. 4.4 display a general tendency to approach u_T and in fact do so excellently in Case 1 and 5, as mentioned previously. The final state diverges slightly from the target in nearly all of the cases around the point $x = 0.9$. This location is actually the location of the steep shock front that develops in the natural final state of our initial boundary value problem Eqn. (2.13). Since the POD basis functions are constructed from the DNS data of the natural uncontrolled solution, this high frequency structure is likely built into the basis functions by design. Only by increasing the number of modes do we expect this localized behavior to damp out.

In the reachable Case 6 target, the POD method better approximates the target state than did the Fourier–ROM method (Fig. 4.4f). The data-driven construction of the POD basis again appears to be the influence generating this result; that is, the natural physics of the Burgers' system is inherently encoded into the POD–G–ROM model and would likely be amenable to small perturbations like the prescribed control input in Case 6.

While overall the POD–G–ROM is not as successful at controlling the evolution of u as is the Fourier–ROM, the ability to match reachable target states is promising. However, we do concede that the full approximations $u_R(x,t)$ are more globally erratic than the counterparts $u_N(x,t)$ for the Fourier–ROM; see Fig. A.6 for the full flow fields. Yet this seemingly poor accuracy is a well known behavior in very low order POD ROMs with few modes, as in the present study.

We emphasize that all of the numerical experiments for the two ROMs presented thus far have only incorporated four modes. To briefly address this point, we focus on the Case 6 target function. By now increasing the number of modes for the POD–G–ROM method, we observe interesting behavior in the controlled solution (Fig. 4.5). It appears that the approximation u_R becomes better as R increases; however, higher frequency oscillations before the steep shock at $x \approx 0.95$ become apparent with more modes. But in Fig. 4.5c, we observe that the amplitude of this undershoot and overshoot artifacts have damped lower when compared to that of Fig. 4.5a and Fig. 4.5b. We would expect this trend of improvement to continue as R is increased, but future investigations on higher performance computing clusters with $R = 40$ modes or greater would provide more conclusive evidence into this matter.



(a) $R = 4$

(b) $R = 10$

(c) $R = 20$

Figure 4.5: POD–G–ROM approximation of target state Case 6 as number of modes R increases.

Chapter 5

Conclusion

In summary, we have designed a numerical framework for cost effective simulation and optimal control of nonlinear partial differential equations (PDE) that govern complex flow phenomena. An open loop control problem for the one-dimensional viscous Burgers' equation is solved by the optimization of a parametrized control input function forcing the PDE. The control objective is to minimize the distance between the final state of the PDE solution, $u(x,T)$, and a prescribed target state, $u_T(x)$. Three approaches utilize our proposed control loop algorithm: a full order model (FOM) for forward baseline simulation, and two reduced order models (ROMs), the Fourier–ROM and the POD–G–ROM. Both ROMs are obtained from a Galerkin projection onto the corresponding Fourier modes (from a Fourier series) or POD modes (from a Proper Orthogonal Decomposition obtained via high fidelity direct numerical simulation snapshots).

We have demonstrated that this low order computational framework succeeds in controlling the evolution of Burgers' equation for a variety of target function test cases. However, controllability of the ROMs is sensitive to the choice of target function. The POD–ROM–G performs more accurately for target functions similar in spatial profile to the final state of the uncontrolled Burgers' PDE, that is, *reachable* target states. In contrast, the Fourier–ROM is generally more effective for arbitrary target functions composed of linear combinations of sinusoids. Our code can replicate the results of many canonical numerical experiments from the reduced order modeling and optimal control literature [see 26, 40].

One limitation in this study is the slow optimization scheme used to decay the cost functional. In forthcoming studies, we aspire to improve the optimization routine through gradient or adjoint-based methods. Adjoint methods have been applied to chaotic dynamical systems effectively [39, 66], and, in particular, we would expect gradient-based methods to perform more efficiently with our reduced order numerical framework than does the current Nelder–Mead simplex optimization approach. Information about the gradient of the cost function (and hence the direction of steepest descent) will inherently reduce the number of expensive function evaluations during the iteration procedure (Alg. 1), which each evaluation corresponding to one forward time evolution of all R modal coefficients.

Further, a more comprehensive study of actuator location would be of use in determining the effectiveness of the ROMs presented in this thesis. While we have shown evidence that the control of Burgers' equation is sensitive to the locality of the pointwise control, additional insight from 2D and 3D studies or variations of the posed 1D initial boundary value problem would improve understanding of this phenomenon. Of relevance in real-world practice, the precise placement location of sensors or actuators is a major factor contributing to the success or failure of flow control applications [49].

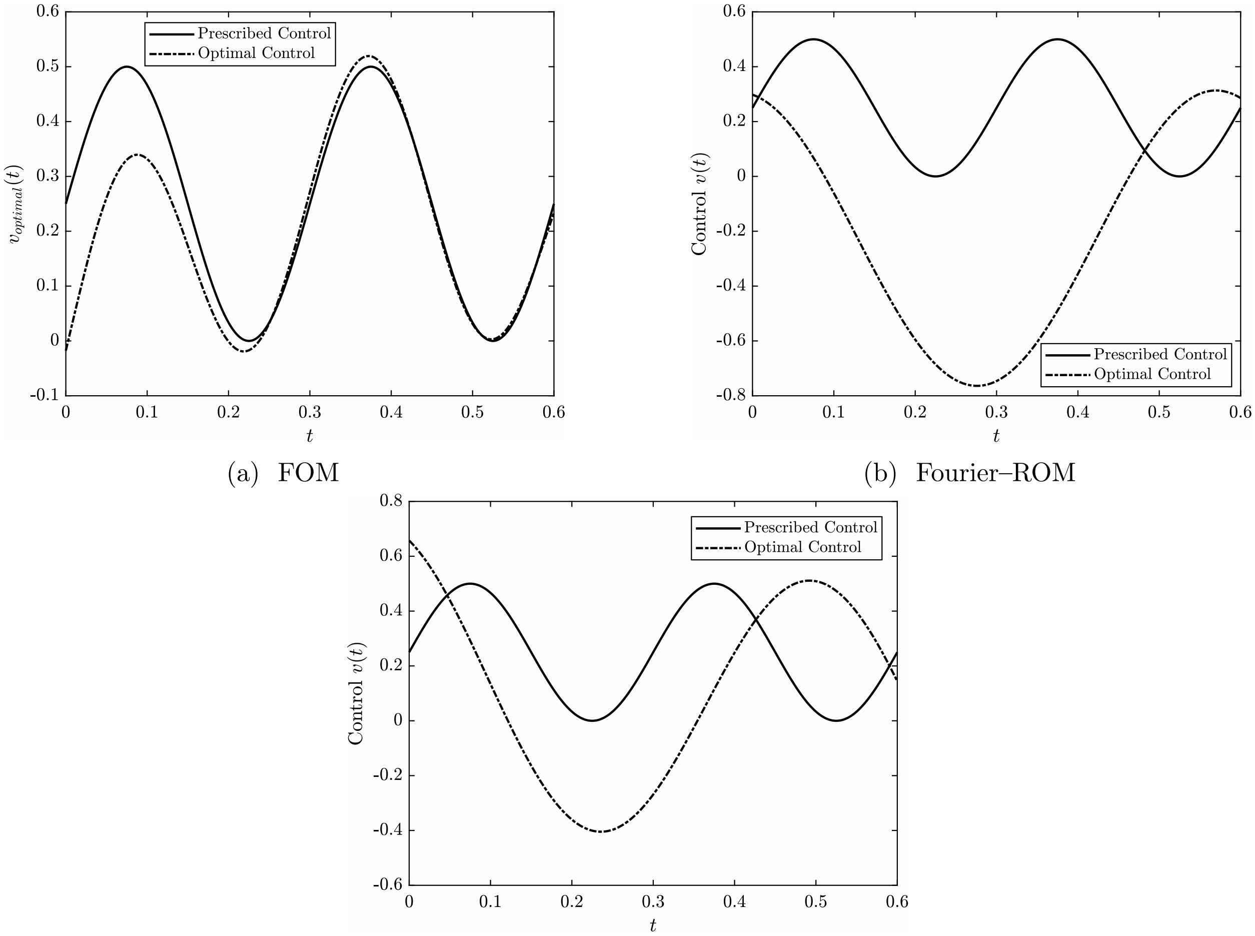
Many similar avenues of research could be explored, even for simple 1D toy problems. Boundary control for Burgers' equation has been studied in [27] with the same control parametrization approach. Kucuk and Sadek suggest that *boundary control* is more effective than the similar *pointwise control* method used in this work. Without the need to prescribe a discrete delta function in this type of problem, an implementation of a POD ROM would be of obvious interest in the boundary control case. Additionally, most modern state-of-the-art POD models incorporate some regularization or closure terms that stabilize the approximations u_N [4, 5]. For example, artificial viscosity has been added to POD ROMs before [see 10]; however, to the best of this author's knowledge, stabilization and regularization routines have not been tested in an optimal flow control setting.

Since the POD is loosely a form of *unsupervised learning*, more traditional *supervised learning* approaches are worthy of investigation and comparison. For example, Dynamic Mode Decomposition (DMD) [44, 59] or radial basis function ROMs using an artificial neural network (ANN) approach are two ideas with great potential. San and Maulik in [55, 56, 57, 58] have demonstrated the excellent performance of ROMs augmented with ANN and extreme learning machine (ELM) procedures. For the practical exploration of these ideas, the python library `TensorFlow` is an attractive package to begin testing these approaches specifically for optimal flow control applications. Lastly, semi-online or fully adaptive approaches for updating a ROM basis during the online stage could facilitate enhanced model prediction for PDE and improved accuracy in flow control scenarios.

Modeling high-dimensional, complex physical or artificial systems with low-dimensional approximations is extremely valuable, especially in the impending era of exascale computing. While computing hardware is evolving in anticipation of this paradigm shift, the software and algorithms must also adapt to meet the unparalleled needs of these architectures. It is clear that the numerical approach to PDE-constrained optimal flow control developed in this thesis serves as an early test bed for future studies involving more challenging problems including the control of full three-dimensional turbulent fluid flows. Our proposed reduced order framework will serve as one early, yet promising, step in the development of next-generation techniques for optimal control applications.

Appendix A

Additional Figures

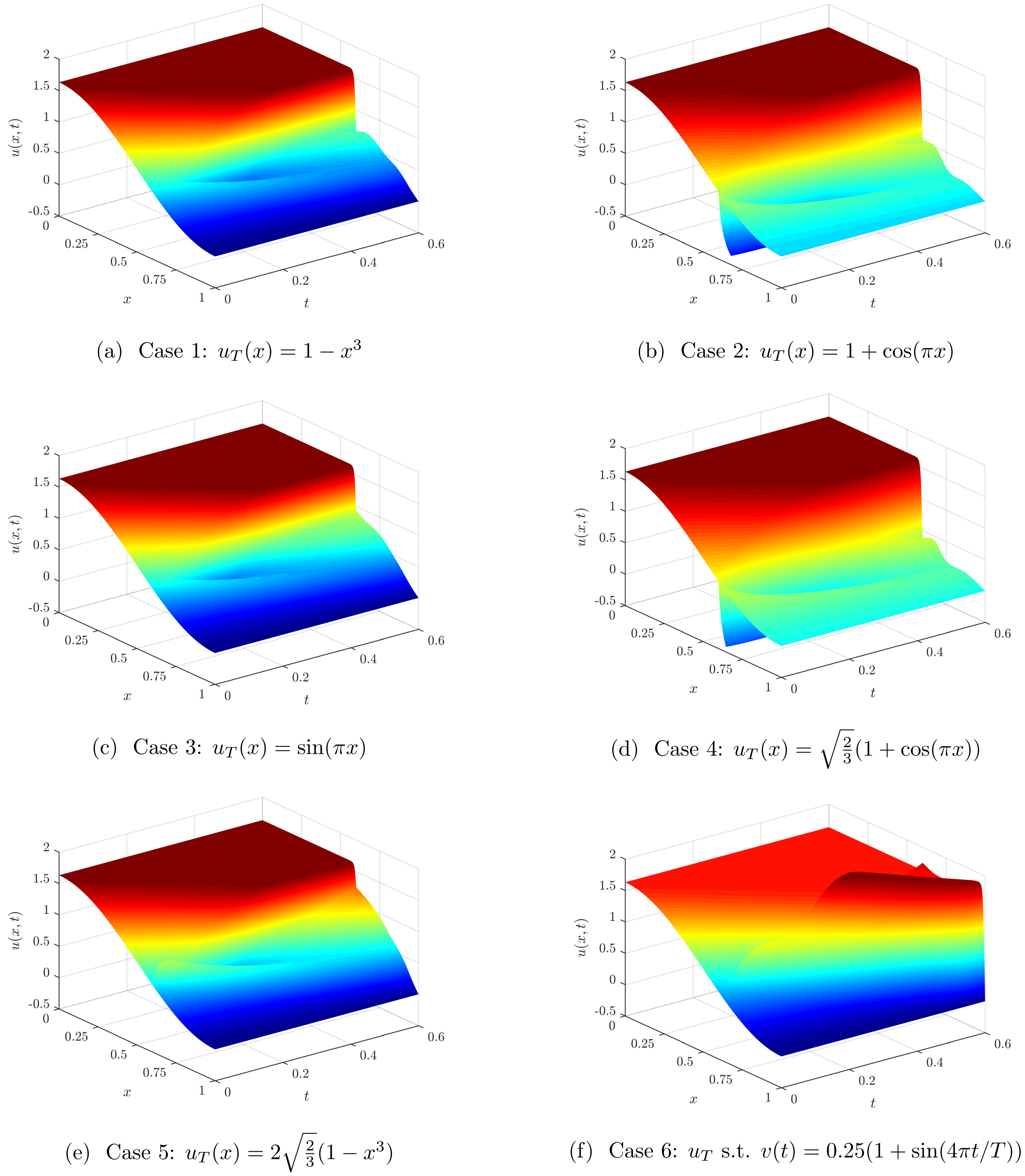


(a) FOM

(b) Fourier–ROM

(c) POD–G–ROM

Figure A.1: Case 6 computed optimal control compared to prescribed control.



(a) Case 1: $u_T(x) = 1 - x^3$

(b) Case 2: $u_T(x) = 1 + \cos(\pi x)$

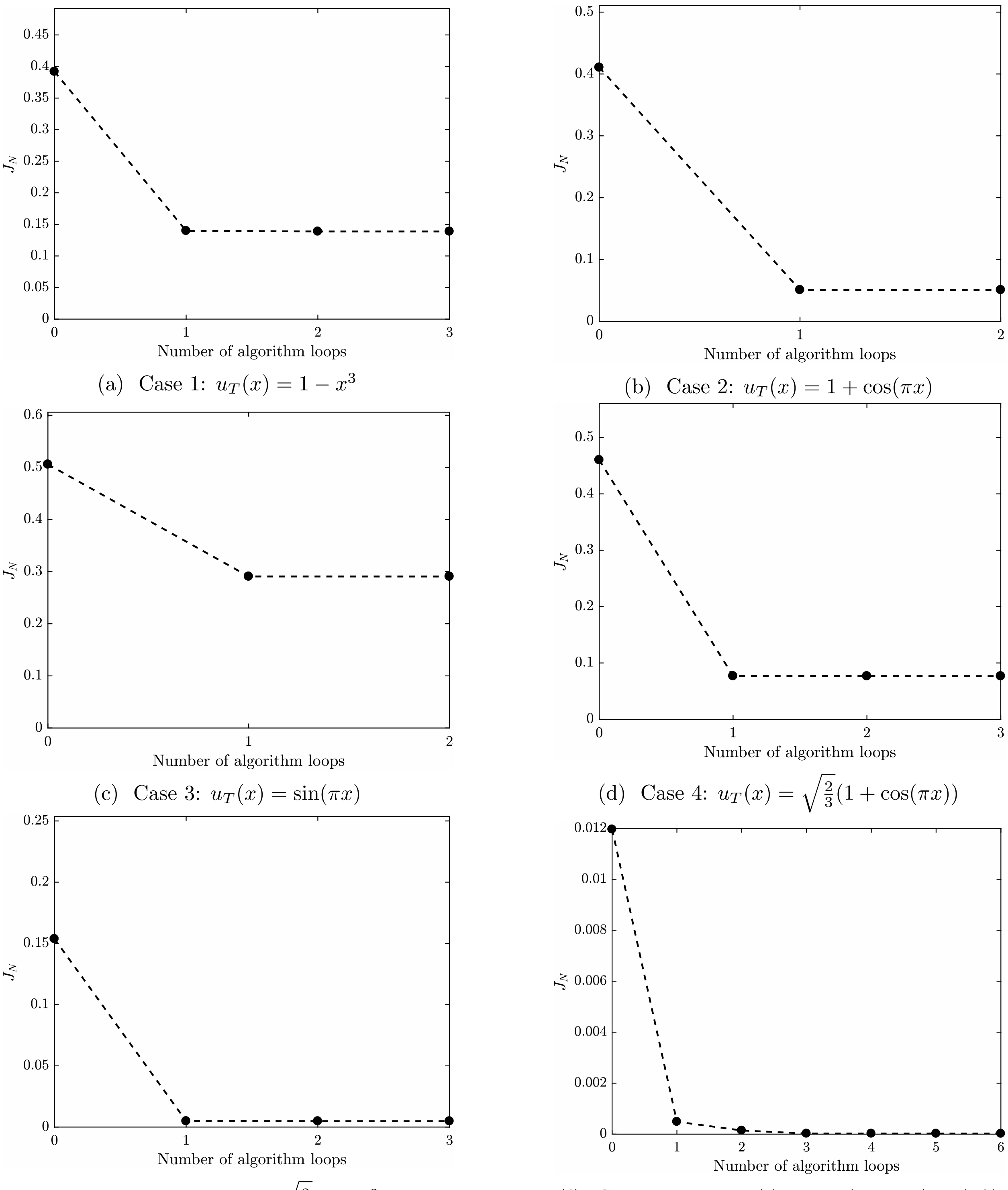
(c) Case 3: $u_T(x) = \sin(\pi x)$

(d) Case 4: $u_T(x) = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$

(e) Case 5: $u_T(x) = 2\sqrt{\frac{2}{3}}(1 - x^3)$

(f) Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$

Figure A.2: FOM full flow field $u(x,t)$ for each target state u_T.



(a) Case 1: $u_T(x) = 1 - x^3$

(b) Case 2: $u_T(x) = 1 + \cos(\pi x)$

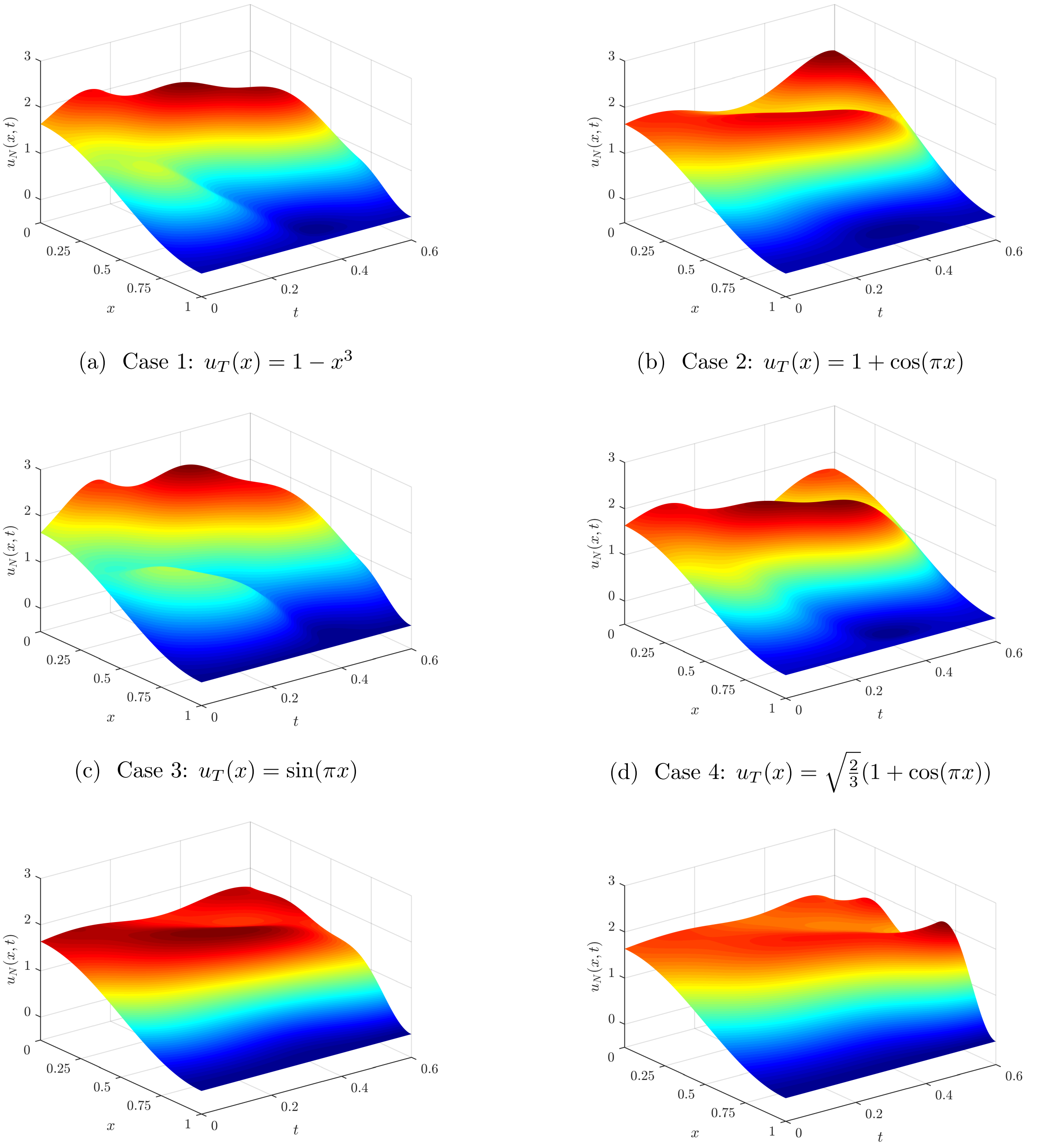
(c) Case 3: $u_T(x) = \sin(\pi x)$

(d) Case 4: $u_T(x) = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$

(e) Case 5: $u_T(x) = 2\sqrt{\frac{2}{3}}(1 - x^3)$

(f) Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$

Figure A.3: FOM convergence history of the discrete cost functional for each u_T.



(a) Case 1: $u_T(x) = 1 - x^3$

(b) Case 2: $u_T(x) = 1 + \cos(\pi x)$

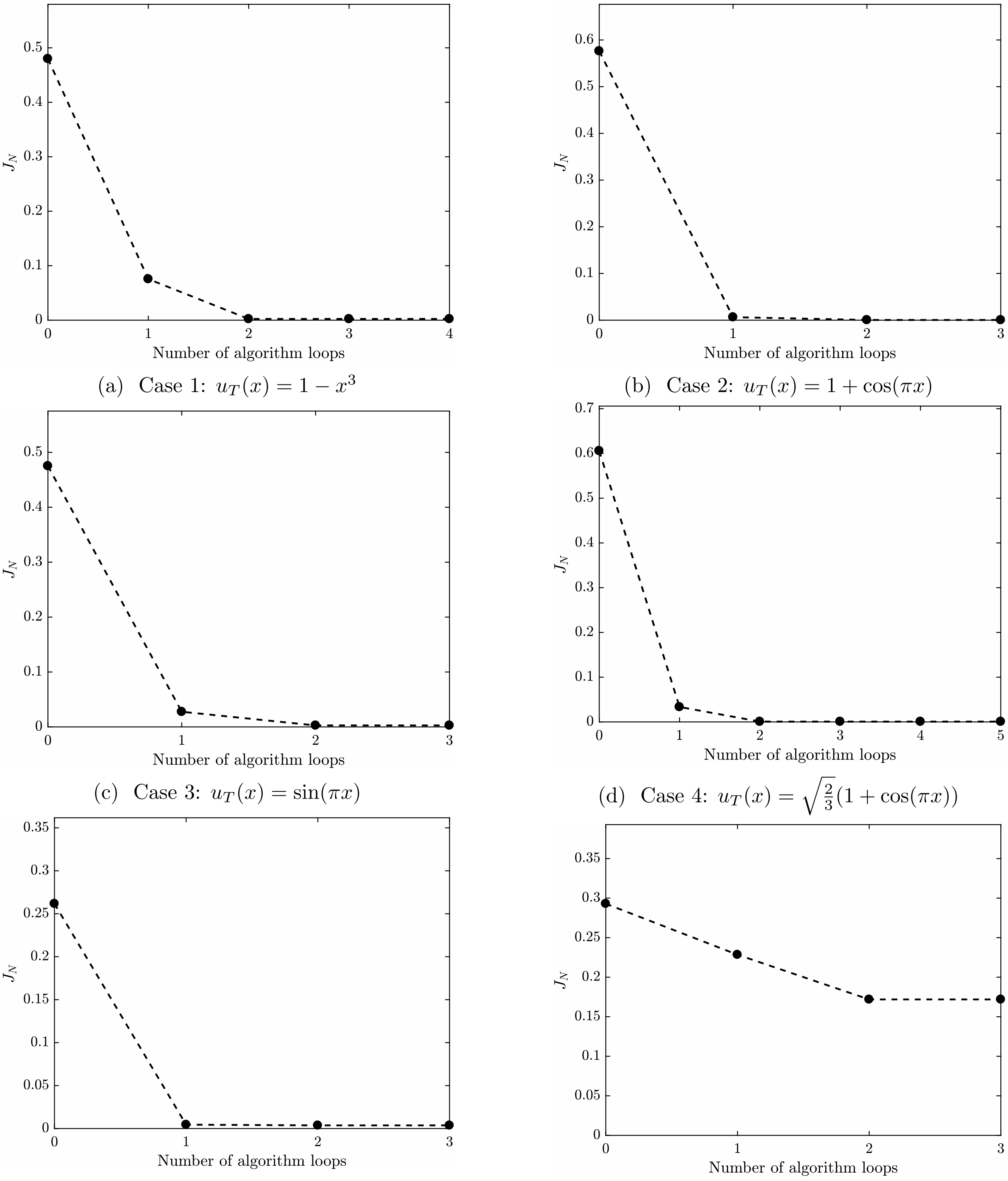
(c) Case 3: $u_T(x) = \sin(\pi x)$

(d) Case 4: $u_T(x) = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$

(e) Case 5: $u_T(x) = 2\sqrt{\frac{2}{3}}(1 - x^3)$

(f) Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$

Figure A.4: Fourier–ROM full flow field $u(x, t)$ for each target state u_T.



(a) Case 1: $u_T(x) = 1 - x^3$

(b) Case 2: $u_T(x) = 1 + \cos(\pi x)$

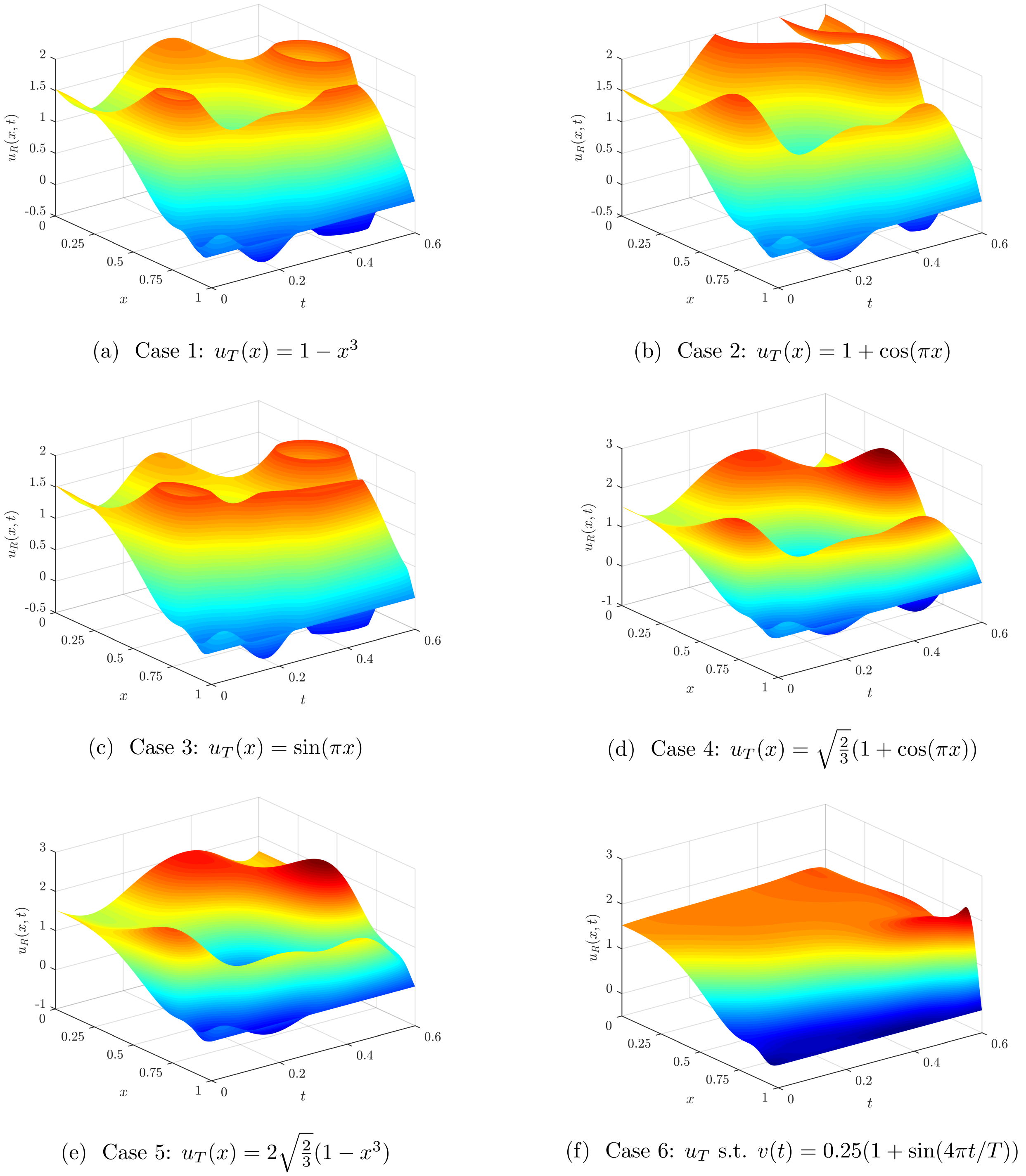
(c) Case 3: $u_T(x) = \sin(\pi x)$

(d) Case 4: $u_T(x) = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$

(e) Case 5: $u_T(x) = 2\sqrt{\frac{2}{3}}(1 - x^3)$

(f) Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$

Figure A.5: Fourier–ROM convergence history of the discrete cost functional for each u_T.



(a) Case 1: $u_T(x) = 1 - x^3$

(b) Case 2: $u_T(x) = 1 + \cos(\pi x)$

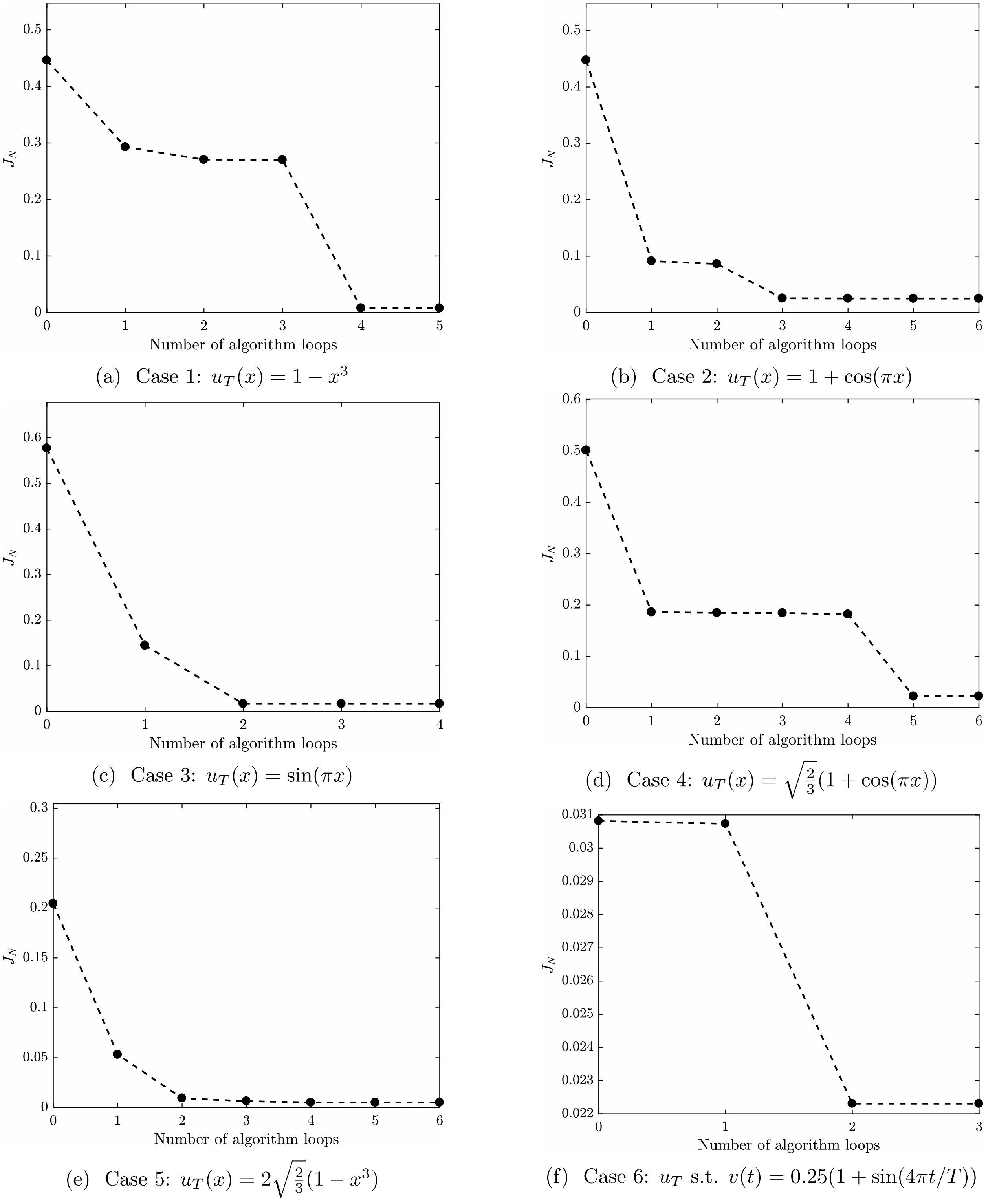
(c) Case 3: $u_T(x) = \sin(\pi x)$

(d) Case 4: $u_T(x) = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$

(e) Case 5: $u_T(x) = 2\sqrt{\frac{2}{3}}(1 - x^3)$

(f) Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$

Figure A.6: POD–G–ROM full flow field $u(x,t)$ for each target state u_T.



(a) Case 1: $u_T(x) = 1 - x^3$

(b) Case 2: $u_T(x) = 1 + \cos(\pi x)$

(c) Case 3: $u_T(x) = \sin(\pi x)$

(d) Case 4: $u_T(x) = \sqrt{\frac{2}{3}}(1 + \cos(\pi x))$

(e) Case 5: $u_T(x) = 2\sqrt{\frac{2}{3}}(1 - x^3)$

(f) Case 6: u_T s.t. $v(t) = 0.25(1 + \sin(4\pi t/T))$

Figure A.7: POD–G–ROM convergence history of the discrete cost functional for each u_T.

Bibliography

[1] Akhtar, I., Nayfeh, A. H., and Ribbens, C. J. “On the stability and extension of reduced-order Galerkin models in incompressible flows”. In: *Theoretical and Computational Fluid Dynamics* 23.3 (2009), pp. 213–237.

[2] Akhtar, I. et al. “Using functional gains for effective sensor location in flow control: A reduced-order modelling approach”. In: *Journal of Fluid Mechanics* 781 (2015), pp. 622–656.

[3] Amsallem, D. and Farhat, C. “Stabilization of projection-based reduced-order models”. In: *International Journal for Numerical Methods in Engineering* 91.4 (2012), pp. 358–377.

[4] Balajewicz, M. and Dowell, E. H. “Stabilization of projection-based reduced order models of the Navier–Stokes”. In: *Nonlinear Dynamics* 70.2 (2012), pp. 1619–1632.

[5] Barone, M. F. et al. “Stable Galerkin reduced order models for linearized compressible flow”. In: *Journal of Computational Physics* 228.6 (2009), pp. 1932–1946.

[6] Benner, P., Gugercin, S., and Willcox, K. “A survey of projection-based model reduction methods for parametric dynamical systems”. In: *SIAM Review* 57 (2015), pp. 483–531.

[7] Benton, E. R. and Platzman, G. W. “A table of solutions of the one-dimensional Burgers equation”. In: *Quarterly of Applied Mathematics* 30.2 (1972), pp. 195–212.

[8] Berggren, M., Glowinski, R., and Lions, J.-L. “A computational approach to controllability issues for flow-related models.(I): Pointwise control of the viscous Burgers equation”. In: *International Journal of Computational Fluid Dynamics* 7.3 (1996), pp. 237–252.

[9] Bewley, T. R. “Flow control: New challenges for a new renaissance”. In: *Progress in Aerospace Sciences* 37.1 (2001), pp. 21–58.

[10] Borggaard, J., Iliescu, T., and Wang, Z. “Artificial viscosity proper orthogonal decomposition”. In: *Mathematical and Computer Modelling* 53.1-2 (2011), pp. 269–279.

[11] Borggaard, J., Wang, Z., and Zietsman, L. “A goal-oriented reduced-order modeling approach for nonlinear systems”. In: *Computers & Mathematics with Applications* 71.11 (2016), pp. 2155–2169.

[12] Buckmaster, T. and Vicol, V. “Nonuniqueness of weak solutions to the Navier-Stokes equation”. In: *arXiv preprint arXiv:1709.10033* (2017).

[13] Bui-Thanh, T. et al. “Goal-oriented, model-constrained optimization for reduction of large-scale systems”. In: *Journal of Computational Physics* 224.2 (2007), pp. 880–896.

[14] Chatterjee, A. “An introduction to the proper orthogonal decomposition”. In: *Current science* (2000), pp. 808–817.

[15] Cheng, X., Zong, X., and Zhang, Y. “Computational method for the optimal control problem for the Korteweg-de Vries equation”. In: *Control Conference (CCC), 2012 31st Chinese*. IEEE. 2012, pp. 1223–1227.

[16] Cordier, L. et al. “Identification strategies for model-based control”. In: *Experiments in Fluids* 54.8 (2013), p. 1580.

[17] Dean, E. and Gubernatis, P. “Pointwise control of Burgers’ equation—a numerical approach”. In: *Computers & Mathematics with Applications* 22.7 (1991), pp. 93–100.

[18] Doering, C. R. and Gibbon, J. D. *Applied analysis of the Navier-Stokes equations*. Vol. 12. Cambridge University Press, 1995.

[19] Evans, L. C. *Partial Differential Equations*. Vol. 19. Graduate Studies in Mathematics. Providence, Rhode Island: AMS, 1998.

[20] Farlow, S. J. *Partial differential equations for scientists and engineers*. Courier Corporation, 1993.

[21] Fefferman, C. L. “Existence and smoothness of the Navier-Stokes equation”. In: *The millennium prize problems* 57 (2006), p. 67.

[22] Glowinski, R. and Lions, J.-L. “Exact and approximate controllability for distributed parameter systems”. In: *Acta numerica* 4 (1995), pp. 159–328.

[23] Gottlieb, S. and Shu, C. W. “Total variation diminishing Runge-Kutta schemes”. In: *Mathematics of Computation* 67.221 (1998), pp. 73–85.

[24] Gunzburger, M. D., Hou, L., and Svobodny, T. P. “Numerical aproximation of an optimal control problem associated with the Navier–Stokes equations”. In: *Applied Mathematics Letters* 2.1 (1989), pp. 29–31.

[25] Gunzburger, M., Jiang, N., and Schneier, M. “An Ensemble-Proper Orthogonal Decomposition Method for the Nonstationary Navier–Stokes Equations”. In: *SIAM Journal on Numerical Analysis* 55.1 (2017), pp. 286–304.

[26] Kucuk, I. and Sadek, I. “An efficient computational method for the optimal control problem for the Burgers equation”. In: *Mathematical and Computer Modelling* 44.11-12 (2006), pp. 973–982.

[27] Kucuk, I. and Sadek, I. “A numerical approach to an optimal boundary control of the viscous Burgers’ equation”. In: *Applied Mathematics and Computation* 210.1 (2009), pp. 126–135.

[28] Kunisch, K. and Volkwein, S. “Galerkin proper orthogonal decomposition methods for parabolic problems”. In: *Numerische Mathematik* 90.1 (2001), pp. 117–148.

[29] Kunisch, K. and Volkwein, S. “Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition”. In: *Journal of optimization theory and applications* 102.2 (1999), pp. 345–371.

[30] Kunisch, K. and Volkwein, S. “Optimal snapshot location for computing POD basis functions”. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 44.3 (2010), pp. 509–529.

[31] Kutz, J. N. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.

[32] Lagarias, J. C. et al. “Convergence properties of the Nelder–Mead simplex method in low dimensions”. In: *SIAM Journal on optimization* 9.1 (1998), pp. 112–147.

[33] Lele, S. K. “Compact finite difference schemes with spectral-like resolution”. In: *Journal of computational physics* 103.1 (1992), pp. 16–42.

[34] Majda, A. J. and Bertozzi, A. L. *Vorticity and incompressible flow*. Vol. 27. Cambridge University Press, 2002.

[35] Moin, P. *Fundamentals of engineering numerical analysis*. Cambridge University Press, 2010.

[36] Noack, B. R. et al. “A hierarchy of low-dimensional models for the transient and post-transient cylinder wake”. In: *Journal of Fluid Mechanics* 497 (2003), pp. 335–363.

[37] Noack, B. R. et al. “Galerkin method for nonlinear dynamics”. In: *Reduced-Order Modelling for Flow Control*. Springer, 2011, pp. 111–149.

[38] Ohkitani, K. and Dowker, M. “Numerical study on comparison of Navier-Stokes and Burgers equations”. In: *Physics of Fluids* 24.5 (2012), p. 055113.

[39] Ou, K. and Jameson, A. “Unsteady adjoint method for the optimal control of advection and Burger’s equations using high-order spectral difference method”. In: *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. 2011, p. 24.

[40] Park, H. and Jang, Y. “Control of Burgers equation by means of mode reduction”. In: *International Journal of Engineering Science* 38.7 (2000), pp. 785–805.

[41] Peherstorfer, B. and Willcox, K. “Data-driven operator inference for nonintrusive projection-based model reduction”. In: *Computer Methods in Applied Mechanics and Engineering* 306 (2016), pp. 196–215.

[42] Pinnau, R. “Model reduction via proper orthogonal decomposition”. In: *Model Order Reduction: Theory, Research Aspects and Applications* 13 (2008), pp. 95–109.

[43] Press, W. H. et al. *Numerical Recipes in FORTRAN*. Cambridge University Press, 1992.

[44] Proctor, J. L., Brunton, S. L., and Kutz, J. N. “Dynamic mode decomposition with control”. In: *SIAM Journal on Applied Dynamical Systems* 15.1 (2016), pp. 142–161.

[45] Raissi, M., Perdikaris, P., and Karniadakis, G. E. “Physics informed deep learning (Part I): Data-driven discovery of nonlinear partial differential equations”. In: *arXiv preprint arXiv:1711.10561* (2017).

[46] Ramos, A. M., Glowinski, R., and Periaux, J. “Pointwise control of the Burgers equation and related Nash equilibrium problems: computational approach”. In: *Journal of optimization theory and applications* 112.3 (2002), pp. 499–516.

[47] Rowley, C. W. “Model reduction for fluids, using balanced proper orthogonal decomposition”. In: *Modeling And Computations In Dynamical Systems: In Commemoration of the 100th Anniversary of the Birth of John von Neumann*. World Scientific, 2006, pp. 301–317.

[48] Rowley, C. W., Colonius, T., and Murray, R. M. “Model reduction for compressible flows using POD and Galerkin projection”. In: *Physica D: Nonlinear Phenomena* 189.1-2 (2004), pp. 115–129.

[49] Rowley, C. W. and Dawson, S. T. “Model reduction for flow analysis and control”. In: *Annual Review of Fluid Mechanics* 49 (2017), pp. 387–417.

[50] Sabeh, Z., Shamsi, M., and Dehghan, M. “Distributed optimal control of the viscous Burgers equation via a Legendre pseudo-spectral approach”. In: *Mathematical Methods in the Applied Sciences* 39.12 (2016), pp. 3350–3360.

[51] Sachs, E. W. and Volkwein, S. “POD-Galerkin approximations in PDE-constrained optimization”. In: *GAMM-Mitteilungen* 33.2 (2010), pp. 194–208.

[52] Sadek, I. and Kucuk, I. “A robust technique for solving optimal control of coupled Burgers’ equations”. In: *IMA Journal of Mathematical Control and Information* 28.3 (2011), pp. 239–250.

[53] San, O. and Borggaard, J. “Basis selection and closure for POD models of convection dominated Boussinesq flows”. In: *21st International Symposium on Mathematical Theory of Networks and Systems*. Vol. 5. 3. 2014.

[54] San, O. and Iliescu, T. “Proper orthogonal decomposition closure models for fluid flows: Burgers equation”. In: *arXiv preprint arXiv:1308.3276* (2013).

[55] San, O. and Maulik, R. “Neural network closures for nonlinear model order reduction”. In: *arXiv preprint arXiv:1705.08532* (2017).

[56] San, O. and Maulik, R. “Extreme learning machine for reduced order modeling of turbulent geophysical flows”. In: *arXiv preprint arXiv:1803.00222* (2018).

[57] San, O. and Maulik, R. “Machine learning closures for model order reduction of thermal fluids”. In: *Applied Mathematical Modelling* (2018).

[58] San, O., Maulik, R., and Ahmed, M. “An artificial neural network framework for reduced order modeling of transient flows”. In: *arXiv preprint arXiv:1802.09474* (2018).

[59] Schmid, P. J. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656 (2010), pp. 5–28.

[60] Serre, D. *Matrices: Theory and applications.* Springer-Verlag, New York, 2002.

[61] Stein, E. M. *Singular integrals and differentiability properties of functions (PMS-30).* Vol. 30. Princeton university press, 2016.

[62] Strikwerda, J. C. *Finite difference schemes and partial differential equations.* 2004.

[63] Taira, K. et al. “Modal analysis of fluid flows: An overview”. In: *arXiv preprint arXiv:1702.01453* (2017).

[64] Trefethen, L. N. *Spectral methods in MATLAB.* Vol. 10. Siam, 2000.

[65] Wall, M. E., Rechtsteiner, A., and Rocha, L. M. “Singular value decomposition and principal component analysis”. In: *A practical approach to microarray data analysis.* Springer, 2003, pp. 91–109.

[66] Wang, Q. “Forward and adjoint sensitivity computation of chaotic dynamical systems”. In: *Journal of Computational Physics* 235 (2013), pp. 1–13.

[67] Wang, Z., McBee, B., and Iliescu, T. “Approximate partitioned method of snapshots for POD”. In: *Journal of Computational and Applied Mathematics* 307 (2016), pp. 374–384.

[68] Wang, Z. et al. “Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison”. In: *Computer Methods in Applied Mechanics and Engineering* 237 (2012), pp. 10–26.

[69] Wu, J. “Seminar Lecture Notes for MATH 6010 Theory and Applications of Fluid Flows”. In: (2018).

[70] Yang, X. et al. “A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations”. In: *Journal of Computational Physics* 228.20 (2009), pp. 7821–7836.

[71] Zhang, L. “Overview of control approaches in Fluid Flow”. In: *Systems and Control in Aeronautics and Astronautics (ISSCAA), 2010 3rd International Symposium on.* IEEE. 2010, pp. 327–332.