REGULAR SENSITIVITY CALCULATION AND GRADIENT-BASED OPTIMIZATION OF CHAOTIC DYNAMICAL SYSTEMS

BY

SEUNG WHAN CHUNG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Theoretical and Applied Mechanics
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Jonathan B. Freund, Chair and Director of Research
Professor Daniel J. Bodony
Assistant Professor Andres Goza
Dr. Eric C. Cyr
Dr. Stephen D. Bond

Abstract

A gradient of a quantity-of-interest $\mathcal{J}$ with respect to problem parameters can augment the utility of a predictive simulation. By itself, the gradient provides sensitivity information to parameters, which can aid uncertainty quantification. Gradient-based optimization can be used in both scientific and engineering applications, including design optimization, data-assimilated modeling and nonmodal stability analysis.

However, obtaining useful gradients for chaotic systems is challenging. The extreme sensitivity to perturbations that defines chaos amplifies the gradient exponentially in time, which impedes both sensitivity analysis and gradient-based optimization. Fundamentally, any $\mathcal{J}$ defined in a chaotic system becomes highly non-convex in time. For such non-convex $\mathcal{J}$, Taylor expansions are useful only in small neighborhoods, which restricts the utility of the gradients, even if computed exactly. Thus they do not indicate a useful parametric sensitivity or guidance toward a useful optimum.

We examine this challenge and investigate routes to circumvent these challenges in two applications. The first is sensitivity computation in particle-in-cell (PIC) simulations involving plasma kinetics. PIC is attractive for representing non-equilibrium plasma distributions in the six-dimensional velocity–position phase-space. To do this, Lagrangian simulation particles represent the position and velocity distribution in a statistical sense. However, computing sensitivity for PIC methods is challenging due to the chaotic dynamics of these particles, and sensitivity techniques remain underdeveloped compared to those for Eulerian discretizations. This challenge is examined from a dual particle–continuum perspective that motivates a new sensitivity discretization. Two routes to sensitivity computation are presented and compared: a direct fully-Lagrangian particle-exact approach provides sensitivities of each particle trajectory, and a new particle-pdf discretization. The new formulation involves a continuum perspective but it is discretized by particles to take the advantages of the same type of Lagrangian particle description leveraged by PIC methods. Since the sensitivity particles in this approach are only indirectly linked to the plasma-PIC particles, they can be positioned and weighted independently for efficiency and accuracy. The corresponding numerical algorithms are presented in mathematical detail. The advantage of the particle-pdf approach in avoiding the spurious chaotic sensitivity of the particle-exact approach is demonstrated for Debye shielding and sheath

configurations. In essence, the continuum perspective makes implicit the distinctness of the particles, which is irrelevant to most prediction goals. In this way it circumvents the Lyapunov instability of the N-body PIC system. The cost of the particle-pdf approach is comparable to the baseline PIC simulation.

The other case considered is optimal control of turbulent flow. Evidence supports the possibility of control of turbulence in some applications, in that there seem to be useful, larger-scale components of the flow, which are less chaotic, in the midst of smaller-scale chaotic turbulence fluctuations. While there have been many attempts to extract model descriptions of such components from the full dynamics, such models are often limited in their applicability or accounting for nonlinearity of turbulence. Thus the full dynamics of turbulent flow is needed to be accurately predictive in simulations. However, in this case the sensitivity of the more chaotic turbulent fluctuations masks that of the useful component of the flow control. This challenge is illustrated with a model control problem of the Lorenz system and analyzed in two aspects: the growth of gradients and non-convexity of $\mathcal{J}$. The horseshoe mapping of chaotic dynamical systems is identified as the root-cause mechanism for both aspects of the challenge, and its impact is quantitatively evaluated in various chaotic flow systems, ranging from the Kuramoto–Sivashinsky Equation to a three-dimensional turbulent Kolmogorov flow. A new optimization framework is proposed based on a penalty-based method. In essence, the simulation time is split into multiple intervals and auxiliary states are introduced at intermediate time points, at which the governing equation is not strictly constrained, thus introducing discontinuities in time. These discontinuities allows $\mathcal{J}$ to be more convex, thus enlarging search scale in the optimization space. They are exploited in this sense then gradually suppressed with increasingly stronger penalty. This multi-step penalty-based optimization is first demonstrated with a one-dimensional logistic map and the Lorenz example. Then its effectiveness is further demonstrated for more complex chaotic systems and ultimately for turbulent Kolmogorov flow. The proposed method finds a solution that suppresses large-scale pressure fluctuations without laminarization, which suggests its ability to target useful components of the flow in the midst of chaotic turbulence, thereby showing its potential for practical turbulent flow controls. It far outperforms a simple gradient-based search.

Acknowledgments

I would like to show my greatest appreciation to Professor Jonathan B. Freund for providing constant and invaluable advising during the entire years of the Ph.D study, which have expanded my boundaries of critical thinking and academic research. Without the time spent working with Prof. Freund, this research would not be possible. I would like to thank Dr. Eric C. Cyr and Dr. Stephen D. Bond for their mentoring, academic discussion, and serving for the defense committee. I also appreciate the advice and efforts for other committee members, Professors Daniel J. Bodony, and Andres Goza.

The progress and achievements of my research were possible because of the pioneering works of Prof. Mingjun Wei, Prof. Jeonglae Kim, and Dr. Ramanathan Vishnampet. I would like to thank the past and present members of Prof. Freund's group for sharing their invaluable insight and knowledge: Prof. Jeonglae Kim, Prof. Jesse Capecelatro, Dr. Spencer Bryngelson, Dr. Jonathan Wang, Esteban Cisneros–Garibay, Wyatt Hagen, as well as Jaekwang Kim.

I would like to express my gratitude to my friend Sang Jin Nam, who fed me on the days of hunger.

This work would not be possible without the unwavering love and sacrifice of my parents, Gilbert and Sunshine, and my sister Karen. I also appreciate the saints from the church in Urbana, who nourished and cherished me in the second part of my life in US. Lastly, the cheering support of my wife, Shulamite, has sustained and encouraged me to complete this work in the last stretch.

This work is supported by the Department of Energy, National Nuclear Security Administration, under Award Number DE-NA0002374.

Isaiah 55:8-9

For My thoughts are not your thoughts,

And your ways are not My ways, declares Jehovah.

For as the heavens are higher than the earth,

So My ways are higher than your ways,

And My thoughts higher than your thoughts.

Table of Contents

List of Tables

List of Figures

Part I

Regular Sensitivity Computation Avoiding Chaotic Effects in Particle-in-cell Plasma Methods

Chapter 1

Introduction

In plasmas, where charged species interactions are important, velocity distributions often deviate from thermodynamic equilibrium, so thermal effects are incompletely described by, for example, a simple pressure [1]. Instead, such a non-equilibrium plasma is often well-described by a six-dimensional velocity–position phase-space distribution governed by the Boltzmann–Vlasov equation. However, representing six-dimensional phase-space with mesh-based discretizations is expensive, both in terms of memory and operation count [2]. Lagrangian particle discretizations, now commonly called particle-in-cell (PIC) methods since mesh-cell-based methods are typically used to accelerate their evaluation, facilitate statistical representation of distributions [3]. Although this introduces statistical fluctuations such that many quantities of interest converge only as $\mathcal{O}(N^{-1/2})$ for N particles [4], PIC methods can often represent phase-space distributions efficiently, while retaining essential nonlinear and non-equilibrium plasma mechanisms [5].

Such PIC methods are well-established; our goal is to augment them with computation of sensitivity. A common need is the calculation of the sensitivity of some Quantity of Interest (QoI) to parameters of the model, which is potentially informative in many circumstances, though we envision two particular uses. For uncertainty quantification, sensitivity quantifies how uncertain model parameters affect the predicted QoI. These can be physical model parameters, boundary condition parameters, or other parameters of the overall PIC models. An example we consider in Chapter 8 is how sensitive the sheath potential drop near the electrode is to the electron–ion charge ratio. This sensitivity could then be used to estimate the impact of uncertainty in this ratio on uncertainty of a corresponding prediction of the charge ratio. Similarly, if a parameter of the PIC model is uncertain, such as the width of the ionization source in the specific PIC model we use in this same example, the uncertainty of the result could likewise be quantified. In such cases, establishing insensitivity to some parameters can focus subsequent effort on the most consequential parametric uncertainties. Similarly, design optimization can be accelerated by sensitivity information when it is interpreted as a gradient. For example, if we wish to optimize the ion current based on an electrode shape, sensitivity to the geometric parameters of the shape can guide this optimization. These are the types of problems that motivate the proposed sensitivity formulation.

The challenge of computing sensitivity in conjunction with a PIC discretization stems from the chaotic dynamics of particles. Even for short simulation times, which are sufficient to converge a QoI prediction, deterministic chaos can produce effectively non-differentiable noise-like fluctuations in the QoI with respect to the parameters [6]. For ergodic dynamical systems, $t \to \infty$ averages of the system converge to the equilibrium distribution, which is differentiable based on linear response theory [7, 8]. However, due to exponential instability to small perturbations, differentiation with respect to parameters does not commute with the $t \to \infty$ limit operation [9, 10], so a naive time-average of sensitivity often fails [6, 11]. Algorithms that have been proposed to circumvent this non-commutability carry certain limitations [6, 9–13]. For example, their computational cost can be exponential in the dimension of the state variables [9], or a large-system optimization is required [10, 13], which renders them impractical for many PIC simulations, where the number of particles often exceeds millions. A more fundamental limitation is that the base assumption of ergodicity requires a well-defined and time-stable equilibrium, which is not always available. Indeed, PIC can be particularly attractive for computing transient responses of non-equilibrium plasmas. Our examples will be cases where a naive finite-difference estimate using PIC simulations with nominally acceptable accuracy leads to spuriously inaccurate sensitivity.

In this paper, the continuum limit (with $N \to \infty$) is presented as an alternative stable limit for sensitivity calculation. Even though specific particle trajectories are highly sensitive to perturbation, the collective distribution can remain Lyapunov stable [14]. In essence, since particles are nominally indistinguishable, an unstably deviated trajectory of one particle can be exchanged with another nearby particle, mollifying their collective deviation. Such a description is common. This phenomenologically is well-recognized in physics, such as Brownian motion [15], Lagrangian turbulence [16], and chaotic mixing [17, p. 185][18].

Of course, as in the time limit $t \to \infty$, a naive averaging over sensitivities of particles to approximate $N \to \infty$ leads a similar challenge of the non-commutability of the limit and differentiation operations. Starting in Chapter 2 we develop a method to avoid this by considering the particle description as a discretization operator applied to the continuum Vlasov–Poisson equation. From this continuum starting point, the sensitivity calculation entails differentiation. There is a choice regarding which is applied first: particle discretization or differentiation. This is a common question in the formulation of sensitivity or adjoint methods [19], though in the context of chaotic sensitivity challenge it introduces a way to obviate the commutation challenge. In essence, we solve the differentiated Vlasov equation for a continuum sensitivity rather than pursue an unobtainably accurate sensitivity of particles. However, direct discretization of the formulation, say on a fixed mesh, would be prohibitive due to the high phase-space dimensionality. Instead, following the original motivation for the PIC discretization, we introduce a particle discretization that is a consistent

approximation to continuum sensitivity with an important new flexibility: particle locations and associated weights can be selected independently of the plasma particle trajectories. This is the key to efficiency and accuracy. In this paper we develop the mathematical details for a specific PIC formulation, though it should be clear that the approach is more general.

In Chapter 2, the basic PIC formulation and sensitivity computation are introduced in the necessary context for formulating numerical schemes. In Chapter 3, we introduce two numerical formulations for computing sensitivities. The first is the direct *particle-exact* approach, which provides the exact (machine precision) sensitivity of Lagrangian particle trajectories in the PIC discretization and serves as a benchmark for illustrating the chaos challenge. The alternative *particle-pdf* approach discretizes the differentiated Vlasov equation with *sensitivity particles*. A Debye-shielding model configuration is introduced in Chapter 4 and is used to demonstrate the method. The advantage of the new approach is assessed in Chapter 5. The sensitivity in the continuum limit and its limit–differentiation non-commutability is analyzed and quantified in Chapter 6. In Chapter 7, the computational cost of the particle-pdf approach is shown to be superior to a reference finite-volume mesh-based sensitivity method and comparable to a typical PIC plasma method. In Chapter 8, sensitivity analysis of a sheath-edge configuration is demonstrated as a more challenging application of particle-pdf approach, illustrating adaptivity opportunities. The flexibility and additional advantage of the new particle-pdf approach are discussed in Chapter 9.

Chapter 2

Preliminaries

2.1 PIC as a discretization of the Vlasov equation

Particle-in-cell (PIC) methods are well-documented [5, 20–23]. A common approach is to start from the particle equations of motion and a regularized electromagnetic or electrostatic force interaction [5, 20–22]. For our purposes, PIC is introduced as a discretization of the Vlasov equation, following most directly the formulation of Lapenta [23].

The Vlasov–Poisson equation,

$$\vec{\mathcal{V}}[\boldsymbol{\Xi}] = 0 \begin{cases} \mathcal{V}_1[\boldsymbol{\Xi}] = \dfrac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \dfrac{q}{m} \mathbf{E} \cdot \nabla_{\mathbf{v}} f = 0 & \text{(2.1a)} \\ \mathcal{V}_2[\boldsymbol{\Xi}] = \nabla^2 \phi + \dfrac{\rho}{\varepsilon_0} = 0 & \text{(2.1b)} \\ \mathcal{V}_3[\boldsymbol{\Xi}] = \mathbf{E} + \nabla \phi = 0, & \text{(2.1c)} \end{cases}$$

with

$$\rho = q \int f \, d^3\mathbf{v} + \rho_{\text{ext}}, \tag{2.2}$$

governs the time-dependent position–velocity ($\mathbf{x}$–$\mathbf{v}$) phase-space population distribution $f(\mathbf{x}, \mathbf{v}, t)$ for a collisionless electrostatic plasma. Here, q and m are the particle charge and mass, ϕ is the electrostatic potential, $\mathbf{E}$ is the electric field, ρ_{ext} is a configuration-specific nominally external charge density, and ε_0 is permittivity. For compactness, we collect the dependent variables as $\boldsymbol{\Xi} = \{f, \phi, \mathbf{E}\}$. Generalization to multiple species with different q and m is straightforward.

The continuous distribution f in (2.1) is approximated by N computational particles [23],

$$f \approx f_N = \sum_{p=1}^{N} f_p(\mathbf{x}, \mathbf{v}, t) = \sum_{p=1}^{N} W_p S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p) S_{\mathbf{v}}(\mathbf{v} - \mathbf{v}_p), \tag{2.3}$$

where each f_p is a fixed nominal particle shape, making up the full f distribution. Each f_p has a weight factor W_p and distributed support $S_{\mathbf{x}} S_{\mathbf{v}}$ in the phase-space, with the shape functions $S_{\mathbf{x}}$ and $S_{\mathbf{v}}$ designed so that f can be approximated on a mesh with smooth forces between corresponding f_p. A tensor-product of

B-splines of order l $S_{\mathbf{x}} = b_l(\mathbf{x})$ is a common choice for space, and a Dirac function $S_{\mathbf{v}} = \delta(\mathbf{v})$ for velocity [23]. Appendix A summarizes details of the current shape functions, none of which are fundamental to the goals of this paper.

The governing equation for any particle f_p, with position $\mathbf{x}_p$ and velocity $\mathbf{v}_p$, is obtained from moments of (2.1a),

$$\int \mathcal{V}_1[\boldsymbol{\Xi}]\, d\mathbf{x}\, d\mathbf{v} = \sum_{p=1}^{N} \dot{W}_p = 0 \tag{2.4a}$$

$$\int \mathbf{x}\mathcal{V}_1[\boldsymbol{\Xi}]\, d\mathbf{x}\, d\mathbf{v} = \sum_{p=1}^{N} W_p \left[\dot{\mathbf{x}}_p - \mathbf{v}_p\right] + \sum_{p=1}^{N} \dot{W}_p \mathbf{x}_p = 0 \tag{2.4b}$$

$$\int \mathbf{v}\mathcal{V}_1[\boldsymbol{\Xi}]\, d\mathbf{x}\, d\mathbf{v} = \sum_{p=1}^{N} W_p \left[\dot{\mathbf{v}}_p - \frac{q}{m}\mathbf{E}_p\right] + \sum_{p=1}^{N} \dot{W}_p \mathbf{v}_p = 0, \tag{2.4c}$$

which yield

$$\dot{W}_p = 0 \tag{2.5a}$$

$$\dot{\mathbf{x}}_p = \mathbf{v}_p \tag{2.5b}$$

$$\dot{\mathbf{v}}_p = \frac{q}{m}\mathbf{E}_p = \frac{q}{m}\int \mathbf{E} S_{\mathbf{x}}(\mathbf{x} - \mathbf{x}_p)\, d\mathbf{x}. \tag{2.5c}$$

With this description, f can be considered to be discretized by particles $\{\mathbf{x}_p, \mathbf{v}_p, W_p\}_{p=1}^{N}$.

The field variables in (2.1) are approximated as mesh-cell averages:

$$\phi = \sum_{\mathbf{i}\in\text{mesh}} \phi_{\mathbf{i}} b_0(\boldsymbol{\xi}_{\mathbf{i}}) \tag{2.6a}$$

$$\mathbf{E} = \sum_{\mathbf{i}\in\text{mesh}} \mathbf{E}_{\mathbf{i}} b_0(\boldsymbol{\xi}_{\mathbf{i}}) \tag{2.6b}$$

$$\rho = \sum_{\mathbf{i}\in\text{mesh}} \rho_{\mathbf{i}} b_0(\boldsymbol{\xi}_{\mathbf{i}})\,, \tag{2.6c}$$

where $\boldsymbol{\xi}_{\mathbf{i}} = \left(\frac{x-x_{\mathbf{i}}}{\Delta x}, \frac{y-y_{\mathbf{i}}}{\Delta y}, \frac{z-z_{\mathbf{i}}}{\Delta z}\right)$ with mesh multi-index $\mathbf{i} = (i, j, k)$. With this description (2.1b) and (2.1c) are discretized as

$$\nabla^2_{\mathbf{i}}\phi_{\mathbf{i}} = -\frac{\rho_{\mathbf{i}}}{\varepsilon_0} = -\frac{1}{\varepsilon_0}\left[\frac{q}{\Delta\mathbf{x}}\int_{\mathbb{R}^3}\int_{\mathbf{x}_{\mathbf{i}}-\frac{\Delta\mathbf{x}}{2}}^{\mathbf{x}_{\mathbf{i}}+\frac{\Delta\mathbf{x}}{2}} f_N\, d\mathbf{x}d\mathbf{v} + \rho_{\text{ext},\mathbf{i}}\right] \tag{2.7a}$$

$$\mathbf{E}_{\mathbf{i}} = -\nabla_{\mathbf{i}}\phi_{\mathbf{i}}. \tag{2.7b}$$

In our particular implementation, the discretized $\mathbf{x}$-gradient $\nabla_{\mathbf{i}}$ and Laplace operator $\nabla^2_{\mathbf{i}}$ are based on

standard second-order centered finite differences. For a uniform grid and B-spline shape functions, the integrals in (2.7a) are evaluated as [23]

$$\begin{aligned}\rho_{\mathbf{i}} &= \frac{q}{\Delta\mathbf{x}} \int_{\mathbb{R}^3} \int_{\mathbf{x_i}-\frac{\Delta\mathbf{x}}{2}}^{\mathbf{x_i}+\frac{\Delta\mathbf{x}}{2}} f_N \, d\mathbf{x}d\mathbf{v} + \rho_{\text{ext},\mathbf{i}} \\ &= \frac{q}{\Delta\mathbf{x}} \int_{\mathbf{x_i}-\frac{\Delta\mathbf{x}}{2}}^{\mathbf{x_i}+\frac{\Delta\mathbf{x}}{2}} \sum_{p=1}^{N} W_p b_l\left(\frac{\mathbf{x}-\mathbf{x}_p}{\Delta\mathbf{x}}\right) \, d\mathbf{x} + \rho_{\text{ext},\mathbf{i}} \\ &= \frac{q}{\Delta\mathbf{x}} \sum_{p=1}^{N} W_p b_{l+1}\left(\boldsymbol{\xi}_{p,\mathbf{i}}\right) + \rho_{\text{ext},\mathbf{i}},\end{aligned} \tag{2.8}$$

where

$$\boldsymbol{\xi}_{p,\mathbf{i}} = \left(\frac{x_p - x_{\mathbf{i}}}{\Delta x}, \frac{y_p - y_{\mathbf{i}}}{\Delta y}, \frac{z_p - z_{\mathbf{i}}}{\Delta z}\right). \tag{2.9}$$

A reversible time integration scheme, which is often used for (2.5b) and (2.5c), is also used here [20]:

$$\frac{\mathbf{x}_p^k - \mathbf{x}_p^{k-1}}{\Delta t} = \mathbf{v}_p^{k-\frac{1}{2}} \tag{2.10a}$$

$$\frac{\mathbf{v}_p^{k+\frac{1}{2}} - \mathbf{v}_p^{k-\frac{1}{2}}}{\Delta t} = \frac{q}{m}\mathbf{E}_p^k, \tag{2.10b}$$

where Δt is the numerical time step. In summary, for discrete state variables $\boldsymbol{\Xi}_D = \{\mathbf{x}_p, \mathbf{v}_p, W_p, \phi_g, \mathbf{E}_g\}$, the following are evaluated sequentially for each particle p and each time step n,

$$\vec{\mathcal{V}}^D[\boldsymbol{\Xi}_D] = 0 \begin{cases} \text{Particle Motion}: & \mathcal{V}_1^D[\boldsymbol{\Xi}_D] = \dfrac{\mathbf{x}_p^k - \mathbf{x}_p^{k-1}}{\Delta t} - \mathbf{v}_p^{k-\frac{1}{2}} = 0 & \text{(2.11a)} \\ \text{Charge Assignment}: & \mathcal{V}_2^D[\boldsymbol{\Xi}_D] = \rho_{\mathbf{i}}^k - \dfrac{q}{\Delta\mathbf{x}} \displaystyle\sum_{p=1}^{N} W_p b_{l+1}\left(\boldsymbol{\xi}_{p,\mathbf{i}}\right) - \rho_{\text{ext},\mathbf{i}} = 0 & \text{(2.11b)} \\ \text{Electrostatic Potential}: & \mathcal{V}_3^D[\boldsymbol{\Xi}_D] = \nabla_{\mathbf{i}}^2\phi_{\mathbf{i}}^k + \dfrac{\rho_{\mathbf{i}}^k}{\varepsilon_0} = 0 & \text{(2.11c)} \\ \text{E-field Evaluation}: & \mathcal{V}_4^D[\boldsymbol{\Xi}_D] = \mathbf{E}_{\mathbf{i}}^k + \nabla_{\mathbf{i}}\phi_{\mathbf{i}}^k = 0 & \text{(2.11d)} \\ \text{Force Assignment}: & \mathcal{V}_5^D[\boldsymbol{\Xi}_D] = \mathbf{E}_p^k - \displaystyle\sum_{\mathbf{i}\in\text{mesh}} \mathbf{E}_{\mathbf{i}}^k b_{l+1}\left(\boldsymbol{\xi}_{p,\mathbf{i}}\right) = 0 & \text{(2.11e)} \\ \text{Particle Acceleration}: & \mathcal{V}_6^D[\boldsymbol{\Xi}_D] = \dfrac{\mathbf{v}_p^{k+\frac{1}{2}} - \mathbf{v}_p^{k-\frac{1}{2}}}{\Delta t} - \dfrac{q}{m}\mathbf{E}_p^k = 0. & \text{(2.11f)} \end{cases}$$

The formulation from (2.1) to (2.5) defines a discretization of f by particles, and (2.11) is a corresponding discretization of the particle dynamics in time and the electrostatic field in space. For an electromagnetic plasma, (2.1b) would be replaced with Maxwell's equations, and it could likewise be augmented with collisional models [5, 23, 24].

2.2 Sensitivity formulation

The goal is to quantify how a quantity-of-interest (QoI) $\mathcal{J}(\boldsymbol{\Theta})$ is sensitive to parameter-of-interest $\boldsymbol{\Theta} = \{\theta_1, \theta_2, \cdots\}$. For plasma kinetics, a common QoI would be the electric field energy or emission rate of electrons on an electrode. We assume that it can be expressed as an integral,

$$\mathcal{J}(\boldsymbol{\Theta}) = \iiint J(\boldsymbol{\Xi}; \boldsymbol{\Theta})\, d\mathbf{x}\, d\mathbf{v}\, dt. \tag{2.12}$$

For any $\theta \in \boldsymbol{\Theta}$, the sensitivity is then

$$\frac{\partial \mathcal{J}}{\partial \theta} = \iiint \left[\nabla_{\boldsymbol{\Xi}} J \cdot {}_{\partial}\lfloor \boldsymbol{\Xi} \rfloor + \frac{\partial J}{\partial \theta} \right] d\mathbf{x}\, d\mathbf{v}\, dt, \tag{2.13}$$

where derivative with respect to θ is denoted as ${}_{\partial}\lfloor \cdot \rfloor \equiv \frac{\partial}{\partial \theta}(\cdot)$. For the two perspectives we develop, this ${}_{\partial}\lfloor \cdot \rfloor$ notation will represent either a sensitivity variable to be discretized or a derivative of a discretized variable. In general, (2.13) shows that the QoI sensitivity depends on the time-dependent plasma-state sensitivity, ${}_{\partial}\lfloor \boldsymbol{\Xi} \rfloor \equiv \frac{\partial \boldsymbol{\Xi}}{\partial \theta}$. A governing equation for ${}_{\partial}\lfloor \boldsymbol{\Xi} \rfloor$ can be developed by differentiating (2.1), the Vlasov–Poisson equation $\vec{\mathcal{V}}[\boldsymbol{\Xi}; \boldsymbol{\Theta}] = 0$, where we now express $\theta \in \boldsymbol{\Theta}$ explicitly:

$$\frac{d}{d\theta}\left(\vec{\mathcal{V}}[\boldsymbol{\Xi}; \boldsymbol{\Theta}] \right) = \frac{\partial \vec{\mathcal{V}}}{\partial \boldsymbol{\Xi}} \cdot {}_{\partial}\lfloor \boldsymbol{\Xi} \rfloor + \frac{\partial \vec{\mathcal{V}}}{\partial \theta} = 0. \tag{2.14}$$

The second term $\partial_\theta \vec{\mathcal{V}}$ represents the explicit dependence of $\vec{\mathcal{V}}$ on θ and is a problem-specific source term. The first factor $\partial_{\boldsymbol{\Xi}} \vec{\mathcal{V}}$ is the linearized dependence about the solution $\boldsymbol{\Xi}$, which would be the same in any such sensitivity analysis.

2.3 Commutability of particle discretization and sensitivity differentiation

With the notation $(\cdot)^D$ indicating a PIC-like discretization as used in (2.11) and $\frac{d}{d\theta}$ the usual derivative (sensitivity) operator, either can be applied first: *discretize-then-differentiate* $\frac{d}{d\theta}(\vec{\mathcal{V}}^D)$ versus *differentiate-then-discretize* $(\frac{d}{d\theta}\vec{\mathcal{V}})^D$. These two routes are illustrated in figure 2.1. The first approach is *particle-exact*: it provides the discrete-exact sensitivity for the computational particles. Its accuracy for the discrete model is only limited by arithmetic precision. No further approximation is made after differentiation. The second approach is a *particle-pdf* discretization: the sensitivity distribution is subsequently discretized.

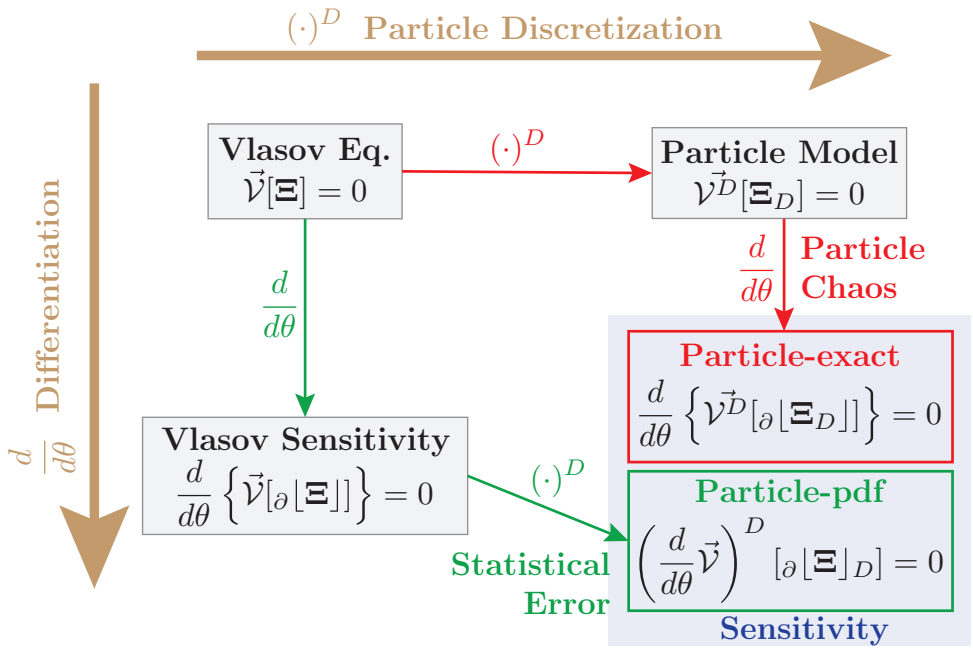


Figure 2.1: Sensitivity governing equation formulation schematic.

As a particle discretization, the PIC method provides a consistent approximation to the $N \to \infty$ (continuum) limit, with errors due to the use of finite N particles. The consequences of finite $\Delta\mathbf{x}$ and Δt are standard and are addressed subsequently and in the appendix, after our key concern: the sensitivity challenge of the particle discretization itself. When differentiated after discretization, the sensitivity is not necessarily as accurate as the original particle discretization of the plasma model. This notion is understood as dual-consistency for adjoint-based methods [19]. The *particle-exact* approach does not provide a sensitivity estimate that is consistent with a continuum formulation. Rather it is exponentially unstable due to deterministic chaos, which will be quantified in Chapter 5. This instability is not artificial: the particles used in the discretization mimic physical particles, which interact chaotically. However, as discussed below, the *particle-pdf* discretization provides a route to avoiding this, so long as sensitivity in the continuum limit is stable, which is a standard case for PIC. To do this, a PIC discretization is formulated for the differentiated continuum plasma model.

Chapter 3

Sensitivity Formulations

We first develop the particle-exact formulation for illustrating the spurious effects of particle chaos in specific cases, before introducing the particle-pdf method to avoid it.

3.1 Particle-exact sensitivity formulation

A discrete analog of the general QoI (2.12) is

$$\mathcal{J}_D = \sum_{p,\mathbf{i},k} J_D\big(\mathbf{x}_p^k, \mathbf{v}_p^{k+\frac{1}{2}}, \phi_\mathbf{i}^k, \mathbf{E}_\mathbf{i}^k\big)\Delta\mathbf{x}\Delta t. \tag{3.1}$$

The corresponding sensitivity from (2.13) is

$$\frac{\partial \mathcal{J}_D}{\partial \theta} = \sum_{p,\mathbf{i},k}\left(\frac{\partial J_D}{\partial \mathbf{x}_p^k}\cdot{}_\partial\lfloor\mathbf{x}_p^k\rfloor + \frac{\partial J_D}{\partial \mathbf{v}_p^{k+\frac{1}{2}}}\cdot{}_\partial\lfloor\mathbf{v}_p^{k+\frac{1}{2}}\rfloor + \frac{\partial J_D}{\partial \phi_\mathbf{i}^k}\,{}_\partial\lfloor\phi_\mathbf{i}^k\rfloor + \frac{\partial J_D}{\partial \mathbf{E}_\mathbf{i}^k}\cdot{}_\partial\lfloor\mathbf{E}_\mathbf{i}^k\rfloor\right), \tag{3.2}$$

and the sensitivities for the discrete variables

$${}_\partial\lfloor\boldsymbol{\Xi}_D\rfloor = \left\{{}_\partial\lfloor\mathbf{x}_p^k\rfloor, {}_\partial\lfloor\mathbf{v}_p^{k+\frac{1}{2}}\rfloor, {}_\partial\lfloor\phi_\mathbf{i}^k\rfloor, {}_\partial\lfloor\mathbf{E}_\mathbf{i}^k\rfloor\right\} \tag{3.3}$$

in (3.2) are obtained by differentiation of (2.11),

$$\frac{d}{d\theta}\vec{\mathcal{V}}^D\big({}_\partial\lfloor\boldsymbol{\Xi}_D\rfloor\big) = \frac{\partial\vec{\mathcal{V}}^D}{\partial\boldsymbol{\Xi}_D}\cdot{}_\partial\lfloor\boldsymbol{\Xi}_D\rfloor + \frac{\partial\vec{\mathcal{V}}^D}{\partial\theta} = 0. \tag{3.4}$$

Including full details of $\frac{\partial\vec{\mathcal{V}}^D}{\partial\boldsymbol{\Xi}_D}\cdot{}_\partial\lfloor\boldsymbol{\Xi}_D\rfloor$ in (3.4), the particle-exact sensitivity is

$$\frac{d}{d\theta}\Big(\vec{\mathcal{V}}^D[\boldsymbol{\Xi}_D]\Big)=0\begin{cases}\text{Particle Motion}: & \dfrac{d}{d\theta}\mathcal{V}_1^D=\dfrac{{}_\partial\lfloor\mathbf{x}_p^k\rfloor-{}_\partial\lfloor\mathbf{x}_p^{k-1}\rfloor}{\Delta t}-{}_\partial\lfloor\mathbf{v}_p^{k-\frac{1}{2}}\rfloor+\dfrac{\partial\mathcal{V}_1^D}{\partial\theta}=0 & (3.5a)\\ \text{Charge Assignment}: & \dfrac{d}{d\theta}\mathcal{V}_2^D={}_\partial\lfloor\rho_{\mathbf{i}}\rfloor-\displaystyle\sum_p qW_p\nabla_{\mathbf{x}}b_{l+1}(\boldsymbol{\xi}_{p,\mathbf{i}})\cdot{}_\partial\lfloor\mathbf{x}_p^k\rfloor & (3.5b)\\ & \qquad+\dfrac{\partial\mathcal{V}_2^D}{\partial\theta}=0 & (3.5c)\\ \text{Electrostatic Potential}: & \dfrac{d}{d\theta}\mathcal{V}_3^D=\nabla_{\mathbf{i}}^2\,{}_\partial\lfloor\phi_{\mathbf{i}}\rfloor+\dfrac{{}_\partial\lfloor\rho_{\mathbf{i}}\rfloor}{\varepsilon_0}+\dfrac{\partial\mathcal{V}_3^D}{\partial\theta}=0 & (3.5d)\\ \text{E-field Evaluation}: & \dfrac{d}{d\theta}\mathcal{V}_4^D={}_\partial\lfloor\mathbf{E}_{\mathbf{i}}\rfloor+\nabla_{\mathbf{i}}\,{}_\partial\lfloor\phi_{\mathbf{i}}\rfloor+\dfrac{\partial\mathcal{V}_4^D}{\partial\theta}=0 & (3.5e)\\ \text{Force Assignment}: & \dfrac{d}{d\theta}\mathcal{V}_5^D={}_\partial\lfloor\mathbf{E}_p\rfloor-\displaystyle\sum_{\mathbf{i}\in\text{mesh}}{}_\partial\lfloor\mathbf{E}_{\mathbf{i}}\rfloor b_{l+1}(\boldsymbol{\xi}_{p,\mathbf{i}}) & (3.5f)\\ & \qquad-\displaystyle\sum_{\mathbf{i}}\mathbf{E}_{\mathbf{i}}\Big(\nabla_{\mathbf{x}}b_{l+1}(\boldsymbol{\xi}_{p,\mathbf{i}})\cdot{}_\partial\lfloor\mathbf{x}_p^k\rfloor\Big)+\dfrac{\partial\mathcal{V}_5^D}{\partial\theta}=0 & (3.5g)\\ \text{Particle Acceleration}: & \dfrac{d}{d\theta}\mathcal{V}_6^D=\dfrac{{}_\partial\lfloor\mathbf{v}_p^{k+\frac{1}{2}}\rfloor-{}_\partial\lfloor\mathbf{v}_p^{k-\frac{1}{2}}\rfloor}{\Delta t}-\dfrac{q}{m}\,{}_\partial\lfloor\mathbf{E}_p\rfloor+\dfrac{\partial\mathcal{V}_6^D}{\partial\theta}=0. & (3.5h)\end{cases}$$

3.2 Particle-pdf formulation

3.2.1 Overview

The particle-pdf formulation reverses the order of operations, first differentiating (2.12) with respect to θ,

$$\frac{\partial\mathcal{J}}{\partial\theta}=\iiint\left(\frac{\partial J}{\partial f}\,{}_\partial\lfloor f\rfloor+\frac{\partial J}{\partial\phi}\,{}_\partial\lfloor\phi\rfloor+\frac{\partial J}{\partial\mathbf{E}}\cdot{}_\partial\lfloor\mathbf{E}\rfloor\right)\,d\mathbf{x}\,d\mathbf{v}\,dt. \tag{3.6}$$

Extension to an array of parameters $\boldsymbol{\Theta}=\{\theta_1,\theta_2,\cdots\}$ is straightforward. The sensitivity distribution ${}_\partial\lfloor f\rfloor$ is then approximated discretely by M sensitivity particles

$${}_\partial\lfloor f\rfloor\approx{}_\partial\lfloor f\rfloor_M=\sum_{s=1}^{M}\hat{W}_sS_{\mathbf{x}}(\mathbf{x}-\hat{\mathbf{x}}_s)S_{\mathbf{v}}(\mathbf{v}-\hat{\mathbf{v}}_s), \tag{3.7}$$

where, similar to W_p in (2.3), $\hat{W}_s$ is the weight of sensitivity particle s, which contributes to the distribution ${}_\partial\lfloor f\rfloor$. A key point is that the sensitivity particles $\{\hat{\mathbf{x}}_s,\hat{\mathbf{v}}_s,\hat{W}_s\}$ can be distinct from the plasma PIC particles, which will be further discussed in Section 3.2.4. The particle-discretized sensitivity state variable

$${}_\partial\lfloor\boldsymbol{\Xi}\rfloor_D=\{\hat{\mathbf{x}}_s,\hat{\mathbf{v}}_s,\hat{W}_s,{}_\partial\lfloor\phi\rfloor_{\mathbf{i}},{}_\partial\lfloor\mathbf{E}\rfloor_{\mathbf{i}}\} \tag{3.8}$$

includes all $s=1,\ldots,M$ sensitivity particles and all mesh points multi-indexed by $\mathbf{i}$.

Differentiation of the Vlasov–Poisson equation (2.1) yields

$$\frac{\partial \vec{\mathcal{V}}}{\partial \boldsymbol{\Xi}} \cdot {}_{\partial}\lfloor \boldsymbol{\Xi} \rfloor + \frac{\partial \vec{\mathcal{V}}}{\partial \theta} = 0 \begin{cases} \dfrac{d\mathcal{V}_1}{d\theta} = \dfrac{\partial}{\partial t}\, {}_{\partial}\lfloor f \rfloor + \mathbf{v} \cdot \nabla_{\mathbf{x}}\, {}_{\partial}\lfloor f \rfloor + \dfrac{q}{m} \mathbf{E} \cdot \nabla_{\mathbf{v}}\, {}_{\partial}\lfloor f \rfloor \\ \qquad\qquad + \dfrac{q}{m}\, {}_{\partial}\lfloor \mathbf{E} \rfloor \cdot \nabla_{\mathbf{v}} f + \dfrac{\partial \mathcal{V}_1}{\partial \theta} = 0 & \text{(3.9a)} \\ \dfrac{d\mathcal{V}_2}{d\theta} = \nabla^2\, {}_{\partial}\lfloor \phi \rfloor + \dfrac{{}_{\partial}\lfloor \rho \rfloor}{\varepsilon_0} + \dfrac{\partial \mathcal{V}_2}{\partial \theta} = 0 & \text{(3.9b)} \\ \dfrac{d\mathcal{V}_3}{d\theta} = {}_{\partial}\lfloor \mathbf{E} \rfloor + \nabla\, {}_{\partial}\lfloor \phi \rfloor + \dfrac{\partial \mathcal{V}_3}{\partial \theta} = 0, & \text{(3.9c)} \end{cases}$$

with sensitivity density

$${}_{\partial}\lfloor \rho \rfloor = \int q\, {}_{\partial}\lfloor f \rfloor \, d^3\mathbf{v} + {}_{\partial}\lfloor \rho_{\text{ext}} \rfloor. \tag{3.10}$$

The sensitivity distribution ${}_{\partial}\lfloor f \rfloor$ evolves according to (3.9a), which is similar to the original Vlasov equation (2.1a), though ${}_{\partial}\lfloor f \rfloor$ is not constant along the characteristics of (3.9a) because of its final two terms, which we write compactly as a nominal source

$$\mathcal{H}(\mathbf{x}, \mathbf{v}, t) \equiv \frac{q}{m}\, {}_{\partial}\lfloor \mathbf{E} \rfloor \cdot \nabla_{\mathbf{v}} f + \frac{\partial \mathcal{V}_1}{\partial \theta}. \tag{3.11}$$

So, unlike the original PIC discretization where the moments of the Vlasov equation (2.1a) lead to constant weights of particles per (2.5), moments of (3.9a) with ${}_{\partial}\lfloor f \rfloor_M$ yield evolving sensitivity-particle weights due to $\mathcal{H}$,

$$\int \frac{d\mathcal{V}_1}{d\theta}\, d\mathbf{x}\, d\mathbf{v} = \int \left[\frac{\partial}{\partial t}\, {}_{\partial}\lfloor f \rfloor_M + \mathcal{H} \right] d\mathbf{x}\, d\mathbf{v} = 0 \tag{3.12a}$$

in addition to PIC-like expressions for particle position and velocity,

$$\int \mathbf{x} \frac{d\mathcal{V}_1}{d\theta}\, d\mathbf{x}\, d\mathbf{v} = \sum_{s=1}^{M} \hat{W}_s \left[\frac{d\hat{\mathbf{x}}_s}{dt} - \hat{\mathbf{v}}_s \right] = 0 \tag{3.12b}$$

$$\int \mathbf{v} \frac{d\mathcal{V}_1}{d\theta}\, d\mathbf{x}\, d\mathbf{v} = \sum_{s=1}^{M} \hat{W}_s \left[\frac{d\hat{\mathbf{v}}_s}{dt} - \frac{q}{m} \mathbf{E}_s \right] = 0. \tag{3.12c}$$

Advection of sensitivity particles thus is identical to PIC particles,

$$\frac{d\hat{\mathbf{x}}_s}{dt} = \hat{\mathbf{v}}_s \tag{3.13a}$$

$$\frac{d\hat{\mathbf{v}}_s}{dt} = \frac{q}{m} \mathbf{E}_s = \frac{q}{m} \sum_{\mathbf{i} \in \text{mesh}} \mathbf{E}_{\mathbf{i}}\, b_{l+1}\left(\hat{\boldsymbol{\xi}}_{s,\mathbf{i}} \right), \tag{3.13b}$$

with $\hat{\boldsymbol{\xi}}_{s,\mathbf{i}} = \left(\frac{\hat{\mathbf{x}}_s - \mathbf{x}_\mathbf{i}}{\Delta \mathbf{x}}\right)$.

To evolve sensitivity-particle weights per (3.12a), $\mathcal{H}$ is also discretized by sensitivity particles $\{\hat{\mathbf{x}}_s, \hat{\mathbf{v}}_s\}$:

$$\mathcal{H}(\mathbf{x}, \mathbf{v}, t) \equiv \frac{q}{m}\, {}_\partial\lfloor \mathbf{E} \rfloor \cdot \nabla_\mathbf{v} f + \frac{\partial \mathcal{V}_1}{\partial \theta} \approx \sum_{s=1}^{M} h_s S_\mathbf{x}(\mathbf{x} - \hat{\mathbf{x}}_s) S_\mathbf{v}(\mathbf{v} - \hat{\mathbf{v}}_s), \tag{3.14}$$

though determining h_s is complicated by the disparate discretizations used for ${}_\partial\lfloor \mathbf{E} \rfloor$ and f and potentially for $\partial_\theta \mathcal{V}_1$. The needs are outlined here and the specific methods are detailed subsequently. The E-field sensitivity ${}_\partial\lfloor \mathbf{E} \rfloor$ is evaluated from (3.9c) on the mesh, with discretization most naturally matching that for the E-field equation (2.7b). However, f is discretized by the plasma-PIC particles per (2.3), and $\partial_\theta \mathcal{V}_1$ depends on the specific sensitivity goal and possibly can involve either mesh or particle discretization. These are schematically represented in Figure 3.1. To evaluate $\mathcal{H}$ (and h_s) at the sensitivity particles, we first evaluate $f(\mathbf{x}, \mathbf{v}, t)$ on a $(\mathbf{x}, \mathbf{v})$-phase-space mesh by interpolating PIC particles to the phase-space mesh, as shown in Figure 3.1. With f represented on the mesh, $\mathcal{H}$ can also be evaluated and interpolated onto sensitivity particle positions, so particle-discretization of the sensitivity distribution ${}_\partial\lfloor f \rfloor_M$ in (3.7) can be integrated in time from (3.12a),

$$\frac{\partial}{\partial t}\, {}_\partial\lfloor f \rfloor_M = -\mathcal{H}. \tag{3.15}$$

In summary, the overall implementation of (3.9a) entails four stages each numerical time step:

1. Particle advection by (3.13),
2. Evaluation of each contribution to $\mathcal{H}$ (3.11) on the phase-space mesh,
3. Interpolation of $\mathcal{H}$ from the mesh to the sensitivity particles by (3.14), and
4. Time-integration of (3.15).

The advection (3.13) matches the original PIC simulation, and thus will only be included in the formulation summary of Section 3.2.5. The other procedures are formulated in detail in the following subsections.

3.2.2 Evaluation of $\mathcal{H}$ on the phase-space grid

The first task is to represent the source term $\mathcal{H}$ on the $(\mathbf{x}, \mathbf{v})$ phase-space mesh $(\mathbf{x}_\mathbf{i}, \mathbf{v}_{\mathbf{i}_v})$,

$$\mathcal{H}(\mathbf{x}, \mathbf{v}) = \sum_{\mathbf{i}, \mathbf{i}_v} \mathcal{H}_{\mathbf{i}, \mathbf{i}_v} b_0\left(\boldsymbol{\xi}_\mathbf{i}\right) b_0\left(\boldsymbol{\zeta}_{\mathbf{i}_v}\right), \tag{3.16}$$

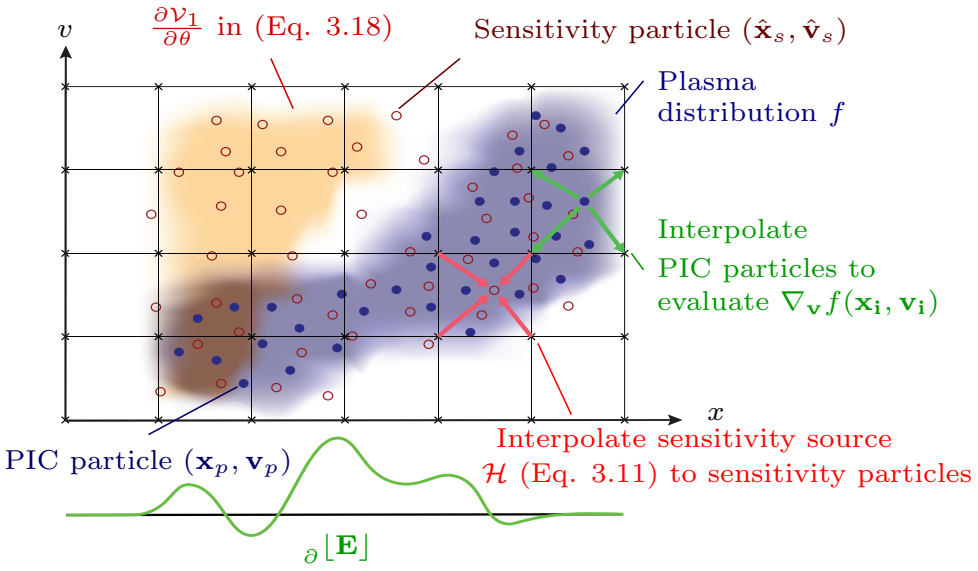


Figure 3.1: Schematic evaluation of $\mathcal{H}$ in (3.11) for an one-dimensional $\mathbf{x} - \mathbf{v}$ space-velocity model and its interpolation onto sensitivity particles by (3.14). As in PIC, key operations involve interpolation between the particles and mesh.

where $\boldsymbol{\xi}_\mathbf{i}$ matches (2.6) and for $\mathbf{v} = (u, v, w)$ the corresponding velocity cell coordinate is

$$\boldsymbol{\zeta}_{\mathbf{i}_v} = \left(\frac{u - u_{\mathbf{i}_v}}{\Delta u}, \frac{v - v_{\mathbf{i}_v}}{\Delta v}, \frac{w - w_{\mathbf{i}_v}}{\Delta w} \right), \tag{3.17}$$

with multi-index $\mathbf{i}_v = (i_v, j_v, k_v)$. Of course, a six-dimensional phase-space mesh is potentially prohibitively expensive for discretizing (2.1) as a PDE system. However, advection by (3.13) avoids the restrictive time step for integrating the corresponding PDE on a mesh. An additional efficiency comes from storing and reusing the interpolation coefficients $b_{l+1}(\boldsymbol{\xi}_{p,\mathbf{i}})$ in (2.11). Though it uses the construct of a six-dimensional mesh, operations are on a particle basis, and detailed algorithm analysis and profiling results in Section 7.1 show that despite using the full phase-space mesh in this way the procedures carry comparable expense to the original PIC algorithm.

The mesh values $\mathcal{H}_{\mathbf{i},\mathbf{i}_v}$ are

$$\mathcal{H}_{\mathbf{i},\mathbf{i}_v} = \left(\frac{q}{m} {}_\partial\lfloor \mathbf{E} \rfloor \cdot \nabla_{\mathbf{v}} f \right)_{\mathbf{i},\mathbf{i}_v} + \left(\frac{\partial \mathcal{V}_1}{\partial \theta} \right)_{\mathbf{i},\mathbf{i}_v}, \tag{3.18}$$

where $(\partial_\theta \mathcal{V}_1)_{\mathbf{i},\mathbf{i}_v}$ depends on the specific sensitivity objective. The E-field sensitivity ${}_\partial\lfloor \mathbf{E} \rfloor_\mathbf{i}$ from the electro-

static system (3.9b) and (3.9c) is discretized on the mesh as

$$\nabla^2_{\mathbf{i}}\, {}_{\partial}\lfloor\phi\rfloor_{\mathbf{i}} + \frac{{}_{\partial}\lfloor\rho\rfloor_{\mathbf{i}}}{\varepsilon_0} + \left(\frac{\partial \mathcal{V}_2}{\partial\theta}\right)_{\mathbf{i}} = 0 \tag{3.19a}$$

$${}_{\partial}\lfloor\mathbf{E}\rfloor_{\mathbf{i}} + \nabla_{\mathbf{i}}\, {}_{\partial}\lfloor\phi\rfloor_{\mathbf{i}} + \left(\frac{\partial \mathcal{V}_3}{\partial\theta}\right)_{\mathbf{i}} = 0, \tag{3.19b}$$

with problem-specific source terms $(\partial_\theta \mathcal{V}_2)_{\mathbf{i}}$ and $(\partial_\theta \mathcal{V}_3)_{\mathbf{i}}$, and from (3.10) the sensitivity charge density is

$$\begin{aligned} {}_{\partial}\lfloor\rho\rfloor_{\mathbf{i}} &= \frac{1}{\Delta\mathbf{x}} \int_{\mathbb{R}^3} \int_{\mathbf{x}_{\mathbf{i}}-\frac{\Delta\mathbf{x}}{2}}^{\mathbf{x}_{\mathbf{i}}+\frac{\Delta\mathbf{x}}{2}} q\, {}_{\partial}\lfloor f\rfloor_M \, d\mathbf{x}\, d\mathbf{v} + {}_{\partial}\lfloor\rho_{\text{ext}}\rfloor_{\mathbf{i}} \\ &= \frac{q}{\Delta\mathbf{x}} \sum_{s=1}^{M} \hat{W}_s b_{l+1}(\hat{\xi}_{s,\mathbf{i}}) + {}_{\partial}\lfloor\rho_{\text{ext}}\rfloor_{\mathbf{i}}. \end{aligned} \tag{3.20}$$

The simplest choice is that the quadrature in (3.20) matches that for the charge density in (2.8). To compute $({}_{\partial}\lfloor\mathbf{E}\rfloor \cdot \nabla_{\mathbf{v}} f)_{\mathbf{i},\mathbf{i}_v}$ in (3.18), the $\mathbf{v}$-gradient and the dot product with ${}_{\partial}\lfloor\mathbf{E}\rfloor$ are first applied to the $\mathbf{v}$-dependent shape function,

$$\begin{aligned} ({}_{\partial}\lfloor\mathbf{E}\rfloor \cdot \nabla_{\mathbf{v}} f)_{\mathbf{i},\mathbf{i}_v} &= \frac{1}{\Delta\mathbf{x}\Delta\mathbf{v}} \int_{\mathbf{v}_{\mathbf{i}_v}-\frac{\Delta\mathbf{v}}{2}}^{\mathbf{v}_{\mathbf{i}_v}+\frac{\Delta\mathbf{v}}{2}} \int_{\mathbf{x}_{\mathbf{i}}-\frac{\Delta\mathbf{x}}{2}}^{\mathbf{x}_{\mathbf{i}}+\frac{\Delta\mathbf{x}}{2}} {}_{\partial}\lfloor\mathbf{E}\rfloor \cdot \sum_{p=1}^{N} W_p S_{\mathbf{x}}(\mathbf{x}-\mathbf{x}_p) \nabla_{\mathbf{v}} S_{\mathbf{v}}(\mathbf{v}-\mathbf{v}_p)\, d\mathbf{x}\, d\mathbf{v} \\ &= \sum_{p=1}^{N} \frac{W_p}{\Delta\mathbf{x}\Delta\mathbf{v}} \int_{\mathbf{x}_{\mathbf{i}}-\frac{\Delta\mathbf{x}}{2}}^{\mathbf{x}_{\mathbf{i}}+\frac{\Delta\mathbf{x}}{2}} S_{\mathbf{x}}(\mathbf{x}-\mathbf{x}_p)\, d\mathbf{x} \int_{\mathbf{v}_{\mathbf{i}_v}-\frac{\Delta\mathbf{v}}{2}}^{\mathbf{v}_{\mathbf{i}_v}+\frac{\Delta\mathbf{v}}{2}} {}_{\partial}\lfloor\mathbf{E}\rfloor_{\mathbf{i}} \cdot \nabla_{\mathbf{v}} S_{\mathbf{v}}(\mathbf{v}-\mathbf{v}_p)\, d\mathbf{v}, \end{aligned} \tag{3.21}$$

where mesh-cell $\mathbf{i}$ values of ${}_{\partial}\lfloor\mathbf{E}\rfloor$ from (3.19) are used. To match the interpolation order in both $\mathbf{x}$ and $\mathbf{v}$, we take $S_{\mathbf{x}} = b_l$ and $S_{\mathbf{v}} = b_{l+1}$, though this is not a requirement so long as $l \geq 1$. The result is

$$({}_{\partial}\lfloor\mathbf{E}\rfloor \cdot \nabla_{\mathbf{v}} f)_{\mathbf{i},\mathbf{i}_v} = \sum_{p=1}^{N} \frac{W_p}{\Delta\mathbf{x}\Delta\mathbf{v}} b_{l+1}\big(\boldsymbol{\xi}_{p,\mathbf{i}}\big)\, {}_{\partial}\lfloor\mathbf{E}\rfloor_{\mathbf{i}} \cdot \nabla_{\mathbf{v}} b_{l+2}\big(\boldsymbol{\zeta}_{p,\mathbf{i}_v}\big)\,, \tag{3.22}$$

with

$$\boldsymbol{\zeta}_{p,\mathbf{i}_v} = \left(\frac{u_p - u_{\mathbf{i}_v}}{\Delta u}, \frac{v_p - v_{\mathbf{i}_v}}{\Delta v}, \frac{w_p - w_{\mathbf{i}_v}}{\Delta w}\right). \tag{3.23}$$

In our implementation, $S_{\mathbf{x}} = b_1$ and $S_{\mathbf{v}} = b_2$, and the formula for $\nabla_{\mathbf{v}} b_3(\boldsymbol{\zeta}_{p,\mathbf{i}_v})$ is (A.2) in Appendix A.

3.2.3 Interpolation of $\mathcal{H}$ from the mesh to the sensitivity particles

We start considering $\mathcal{H}(\mathbf{x},\mathbf{v},t)$ in a generic continuous form, which must be linked to the particle distributions. The discretization of $\mathcal{H}$ is considered subsequently. The irregular and evolving distribution of particles

over the $(\mathbf{x}, \mathbf{v})$ phase-space is quantified by their number density,

$$\hat{n} = \sum_{s=1}^{M} \hat{n}_s = \sum_{s=1}^{M} S_{\mathbf{x}}(\mathbf{x} - \hat{\mathbf{x}}_s) S_{\mathbf{v}}(\mathbf{v} - \hat{\mathbf{v}}_s). \tag{3.24}$$

This is equivalent to (3.7) with $\hat{W}_s = 1$, and allows us to define a particle-specific source strength $h = \mathcal{H}/\hat{n}$, such that $\mathcal{H}$ can be ascribed to the particles as

$$\mathcal{H}(\mathbf{x}, \mathbf{v}) = h(\mathbf{x}, \mathbf{v}) \hat{n}(\mathbf{x}, \mathbf{v}) = \sum_{s} \underbrace{h(\mathbf{x}, \mathbf{v})}_{\text{source strength}} \underbrace{S_{\mathbf{x}}(\mathbf{x} - \hat{\mathbf{x}}_s) S_{\mathbf{v}}(\mathbf{v} - \hat{\mathbf{v}}_s)}_{\text{particle configuration}}. \tag{3.25}$$

To do this, we attempt to find a representation value at particle s, denoting as h_s, considering its finite-sized shape. This can be evaluated by convolution, as for the electric field for particles $\mathbf{E}_p$ in (2.5c),

$$h_s = S * h\big|_{\hat{\mathbf{x}}_s, \hat{\mathbf{v}}_s} \equiv \iint h(\mathbf{x}, \mathbf{v}) S_{\mathbf{x}}(\mathbf{x} - \hat{\mathbf{x}}_s) S_{\mathbf{v}}(\mathbf{v} - \hat{\mathbf{v}}_s) \, d\mathbf{x} d\mathbf{v}. \tag{3.26}$$

Replacing $h(\mathbf{x}, \mathbf{v})$ in (3.25) with h_s provides the discretization of $\mathcal{H}$ with sensitivity particles given in (3.14).

In order to calculate the convolution in (3.26), $h = \mathcal{H}/\hat{n}$ is evaluated on sensitivity particles. First, the mesh values of $\mathcal{H}$ are evaluated via (3.18), and then $\hat{n}$ is interpolated from sensitivity particles to the mesh as in (3.22), using $S_{\mathbf{x}} = S_{\mathbf{v}} = b_l$

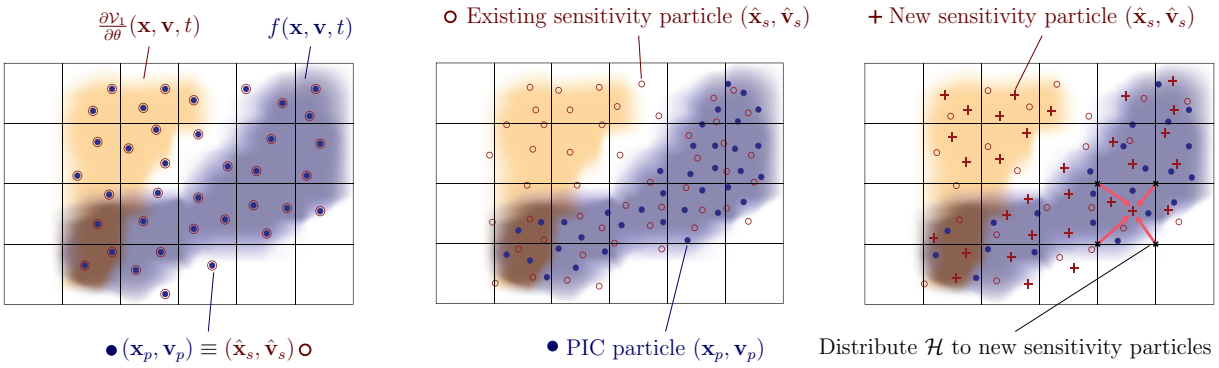
$$\begin{aligned} \hat{n}_{\mathbf{i},\mathbf{i}_v} &= \frac{1}{\Delta\mathbf{x}\Delta\mathbf{v}} \int_{\mathbf{x}_\mathbf{i} - \frac{\Delta\mathbf{x}}{2}}^{\mathbf{x}_\mathbf{i} + \frac{\Delta\mathbf{x}}{2}} \int_{\mathbf{v}_{\mathbf{i}_v} - \frac{\Delta\mathbf{v}}{2}}^{\mathbf{v}_{\mathbf{i}_v} + \frac{\Delta\mathbf{v}}{2}} \hat{n}(\mathbf{x}, \mathbf{v}) \, d\mathbf{x} \, d\mathbf{v} \\ &= \frac{1}{\Delta\mathbf{x}\Delta\mathbf{v}} \sum_{s=1}^{M} b_{l+1}\left(\hat{\boldsymbol{\xi}}_{s,\mathbf{i}}\right) b_{l+1}\left(\hat{\boldsymbol{\zeta}}_{s,\mathbf{i}_v}\right), \end{aligned} \tag{3.27}$$

with $\hat{\boldsymbol{\xi}}_{s,\mathbf{i}}$ and $\hat{\boldsymbol{\zeta}}_{s,\mathbf{i}_v}$ defined as $\boldsymbol{\xi}_{p,\mathbf{i}}$ in (2.9) and $\boldsymbol{\zeta}_{p,\mathbf{i}_v}$ in (3.23), now for $(\hat{\mathbf{x}}_s, \hat{\mathbf{v}}_s)$. Evaluating $h_{\mathbf{i},\mathbf{i}_v} = \frac{\mathcal{H}_{\mathbf{i},\mathbf{i}_v}}{\hat{n}_{\mathbf{i},\mathbf{i}_v}}$ with (3.18) and (3.27), we obtain (3.26) in fully-discrete form

$$h_s = \sum_{\mathbf{i},\mathbf{i}_v} h_{\mathbf{i},\mathbf{i}_v} b_{l+1}\left(\hat{\boldsymbol{\xi}}_{s,\mathbf{i}}\right) b_{l+1}\left(\hat{\boldsymbol{\zeta}}_{s,\mathbf{i}_v}\right). \tag{3.28}$$

Representation of $\mathcal{H}$ this way is predicated on there being sufficient particles in the necessary region to support it. Thus $\hat{n}_{\mathbf{i},\mathbf{i}_v}$ must be finite wherever $\mathcal{H} \neq 0$:

$$\hat{n}_{\mathbf{i},\mathbf{i}_v} > 0 \qquad \forall (\mathbf{i}, \mathbf{i}_v) \in \{(\mathbf{i}, \mathbf{i}_v) \mid \mathcal{H}_{\mathbf{i},\mathbf{i}_v} \neq 0\}. \tag{3.29}$$



(a) Collocated (b) Non-collocated (c) Particle addition

Figure 3.2: The (a) collocated weight-evolution, (b) non-collocated weight-evolution, and (c) particle addition schemes.

If this is not the case, the flexibility of the formulation can be leveraged to create new particles to represent $\mathcal{H}$. Of course, accuracy goals might indicate the addition of more particles too, as considered in subsequent examples.

3.2.4 Time-integration of $\mathcal{H}$

Three basis methods for representing $\mathcal{H}$ are illustrated in Figure 3.2. They differ primarily in the choice of the particles with which $\mathcal{H}$ is discretized. In the first, sensitivity particles are collocated with PIC particles and move synchronously with them. This is relatively simple, and it benefits from some redundancy of calculation, but it does not expose all the available flexibility. The second method uses distinct sensitivity particles. This independence is further exploited in the third method, which allows the addition and redistribution of sensitivity particles each time step, opening opportunities for advantageous adaptation. In the first two methods, only the weights in (3.7) of existing particles evolve in time. In the third method, the weights are assigned to new particles to represent $\mathcal{H}$. We refer these methods as collocated, non-collocated, and particle addition. Collocated particles advect just as the PIC particles, so interpolation coefficients in (2.8) can be reused. However, to be effective, it is necessary that the original PIC particles be arranged so that the $\hat{n}$ support condition (3.29) be satisfied. To satisfy this minimum necessity or to improve resolution, non-collocated and particle addition methods provide additional flexibility. A combination of these two schemes provides the greatest flexibility to represent sparse sensitivity distributions, which is demonstrated in Chapter 8.

For both the collocated and non-collocated weight-evolution schemes, particle source strength h_s (3.28) is evaluated for existing sensitivity particles. Then with the discretization of $\mathcal{H}$ in (3.14), the zeroth moment

of the differentiated Vlasov equation (3.12a) becomes

$$\frac{d\hat{W}_s}{dt} = -h_s \qquad \text{for } s = 1, \ldots, M, \tag{3.30}$$

which allows sensitivity weights $\hat{W}_s$ to evolve independently from the PIC particle weights.

For the particle-addition scheme, N_{new} particles $(\hat{\mathbf{x}}_i, \hat{\mathbf{v}}_i)$ are created, potentially in each time step, which must be done such that

$$\hat{n}_{\text{new}} = \sum_{i=1}^{N_{\text{new}}} S_{\mathbf{x}}(\mathbf{x} - \hat{\mathbf{x}}_i) S_{\mathbf{v}}(\mathbf{v} - \hat{\mathbf{v}}_i) \tag{3.31}$$

satisfies (3.29). As presented, the new particles assume the full burden of representing the source $\mathcal{H}$, though this can be generalized. In this simplest form, the source term $\mathcal{H}$ is represented just by these new particles as in (3.25)

$$\mathcal{H}(\mathbf{x}, \mathbf{v}) \approx \sum_{i=1}^{N_{\text{new}}} h_i S_{\mathbf{x}}(\mathbf{x} - \hat{\mathbf{x}}_i) S_{\mathbf{v}}(\mathbf{v} - \hat{\mathbf{v}}_i), \tag{3.32}$$

where the weight factor h_i is evaluated via (3.28) using only the new particles, so mesh values of $\hat{n}_{\text{new}}$ are pre-computed and do not need to be updated each time step. The time-integration of (3.12a) is realized through the addition of these new particles,

$${}_{\partial}\lfloor f \rfloor_{M'} = {}_{\partial}\lfloor f \rfloor_{M} + \sum_{i=1}^{N_{\text{new}}} \hat{W}_i S_{\mathbf{x}}(\mathbf{x} - \hat{\mathbf{x}}_i) S_{\mathbf{v}}(\mathbf{v} - \hat{\mathbf{v}}_i), \tag{3.33}$$

with their weights $\hat{W}_i$ set to match integral of h_i during Δt,

$$\hat{W}_i = -\int_{t^*}^{t^*+\Delta t} h_i \, dt \qquad \text{for } i = 1, \ldots, N_{\text{new}}, \tag{3.34}$$

which replaces the weight evolution (3.30). The exact time-integration of the entire governing equation is achieved only when $\Delta t \to 0$, when particles are added continuously in time. For a finite Δt, (3.34) is evaluated with a numerical quadrature rule, that will be illustrated in section 3.2.5.

Flexibility can be augmented by periodically replacing particles, which requires redistribution of ${}_{\partial}\lfloor f \rfloor$ to a new set of M_{new} particles,

$${}_{\partial}\lfloor f \rfloor_{\text{redistributed}} = \sum_{r=1}^{M_{\text{new}}} \hat{W}_r S_{\mathbf{x}}(\mathbf{x} - \hat{\mathbf{x}}_r) S_{\mathbf{v}}(\mathbf{v} - \hat{\mathbf{v}}_r). \tag{3.35}$$

Similar procedures are used in vortex methods [25, 26], smooth-particle hydrodynamics [27], and PIC meth-

ods [28, 29]. The new-particle weights in (3.35) are

$$\hat{W}_r = \sum_{s=1}^{M} \frac{\hat{W}_s}{\Delta\mathbf{x}\Delta\mathbf{v}} \tilde{b}_l\left(\hat{\boldsymbol{\xi}}_{s,r}\right) \tilde{b}_l\left(\hat{\boldsymbol{\zeta}}_{s,r}\right), \tag{3.36}$$

with $(\hat{\boldsymbol{\xi}}_{s,r}, \hat{\boldsymbol{\zeta}}_{s,r}) = \left(\frac{\hat{\mathbf{x}}_s - \hat{\mathbf{x}}_r}{\Delta\mathbf{x}}, \frac{\hat{\mathbf{v}}_s - \hat{\mathbf{v}}_r}{\Delta\mathbf{v}}\right)$ and $\tilde{b}_l$ is the modified B-spline function of order l, which is more accurate than standard B-spline of same order in (2.8), and keeps the redistribution error below that of other procedures in the particle-pdf method [29, 30].

3.2.5 Summary

The particle-pdf approach has several steps common with PIC,

$$\text{Particle Motion}: \quad \frac{d\hat{\mathbf{x}}_s}{dt} = \hat{\mathbf{v}}_s \tag{3.37a}$$

$$\text{Sensitivity Charge}: \quad {}_{\partial}\lfloor\rho\rfloor_{\mathbf{i}} = \frac{q}{\Delta\mathbf{x}} \sum_{s}^{M^k} \hat{W}_s b_{l+1}(\hat{\xi}_{s,\mathbf{i}}) + {}_{\partial}\lfloor\rho_{ext}\rfloor_g \tag{3.37b}$$

$$\text{Sensitivity Potential}: \quad \nabla^2_{\mathbf{i}}\, {}_{\partial}\lfloor\phi\rfloor_{\mathbf{i}} = -\frac{{}_{\partial}\lfloor\rho\rfloor_{\mathbf{i}}}{\varepsilon_0} \tag{3.37c}$$

$$\text{Sensitivity E-field}: \quad {}_{\partial}\lfloor\mathbf{E}\rfloor_{\mathbf{i}} = -\nabla_{\mathbf{i}}\, {}_{\partial}\lfloor\phi\rfloor_{\mathbf{i}} \tag{3.37d}$$

$$\text{Force Assignment}: \quad \mathbf{E}_s = \sum_{\mathbf{i}\in\text{mesh}} \mathbf{E}_{\mathbf{i}} b_{l+1}(\hat{\xi}_{s,\mathbf{i}}) \tag{3.37e}$$

$$\text{Particle Acceleration}: \quad \frac{d\hat{\mathbf{v}}_s}{dt} = \frac{q}{m}\mathbf{E}_s. \tag{3.37f}$$

In addition, the sensitivity source term $\mathcal{H}$ is evaluated as

$$\text{Source Evaluation}: \quad ({}_{\partial}\lfloor\mathbf{E}\rfloor \cdot \nabla_{\mathbf{v}} f)_{\mathbf{i},\mathbf{i}_v} = \sum_{p=1}^{N} \frac{W_p}{\Delta\mathbf{x}\Delta\mathbf{v}} b_{l+1}\left(\xi_{p,\mathbf{i}}\right) {}_{\partial}\lfloor\mathbf{E}\rfloor_{\mathbf{i}} \cdot \nabla_{\mathbf{v}} b_{l+2}\left(\zeta_{p,\mathbf{i}_v}\right), \tag{3.38}$$

then $\mathcal{H}$ is discretized by sensitivity particles with weights h_s,

$$\hat{n}\text{ Evaluation}: \quad \hat{n}_{\mathbf{i},\mathbf{i}_v} = \frac{1}{\Delta\mathbf{x}\Delta\mathbf{v}} \sum_{s} b_{l+1}\left(\hat{\boldsymbol{\xi}}_{s,\mathbf{i}}\right) b_{l+1}\left(\hat{\boldsymbol{\zeta}}_{s,\mathbf{i}_v}\right) \tag{3.39a}$$

$$\text{Source weights}: \quad h_s = \sum_{\mathbf{i},\mathbf{i}_v} \frac{\mathcal{H}_{\mathbf{i},\mathbf{i}_v}}{\hat{n}_{\mathbf{i},\mathbf{i}_v}} b_{l+1}\left(\hat{\boldsymbol{\xi}}_{s,\mathbf{i}}\right) b_{l+1}\left(\hat{\boldsymbol{\zeta}}_{s,\mathbf{i}_v}\right), \tag{3.39b}$$

where the particles are either existing particles for collocated or non-collocated methods, or new particles for the particle-addition method. For collocated or non-collocated methods, the weight is integrated in time

for existing particles,

$$\text{Weight Evolution}: \qquad \frac{d\hat{W}_{\text{exist}}}{dt} = -h_{\text{exist}}. \tag{3.40a}$$

For the particle addition method, a new set of particles are added with their weights,

$$\text{Addition}: \qquad \hat{W}_{\text{new}} = -\int_{t^*}^{t^*+\Delta t} h_{\text{new}}\, dt, \tag{3.40b}$$

with replacement and redistribution of ${}_{\partial}\lfloor f \rfloor_M$ as deemed advantageous:

$$\text{Redistribution}: \qquad {}_{\partial}\lfloor f \rfloor_{\text{redistribution}}(\mathbf{x}, \mathbf{v}) = \sum_{r=1}^{M_{\text{new}}} \hat{W}_r S_x(\mathbf{x} - \hat{\mathbf{x}}_r) S_v(\mathbf{v} - \hat{\mathbf{v}}_r). \tag{3.41}$$

Redistributed weights are calculated as

$$\hat{W}_r = \sum_{s=1}^{M} \frac{\hat{W}_s}{\Delta \mathbf{x} \Delta \mathbf{v}} \tilde{b}_{l+1}\left(\hat{\boldsymbol{\xi}}_{s,r}\right) \tilde{b}_{l+1}\left(\hat{\boldsymbol{\zeta}}_{s,r}\right). \tag{3.42}$$

There is considerable flexibility for time-integrating (3.37a), (3.37f) and (3.40). A Liouville formulation [15], similar to standard PIC methods, is illustrated in Figure 3.3. It provides a second-order time-reversible integration. As presented, since sensitivity particle weights do not affect the electrostatic field on particle s ($\mathbf{E}_s^k$), it does not need to be re-evaluated each time step. For collocated particles, the advection stages (3.37a), (3.37e), and (3.37f) are unneeded, and interpolation coefficients $b_{l+1}(\xi_{p,\mathbf{i}})$ in (2.8) can be reused for $\hat{\xi}_{s,\mathbf{i}}$ in (3.37b) and (3.39b).

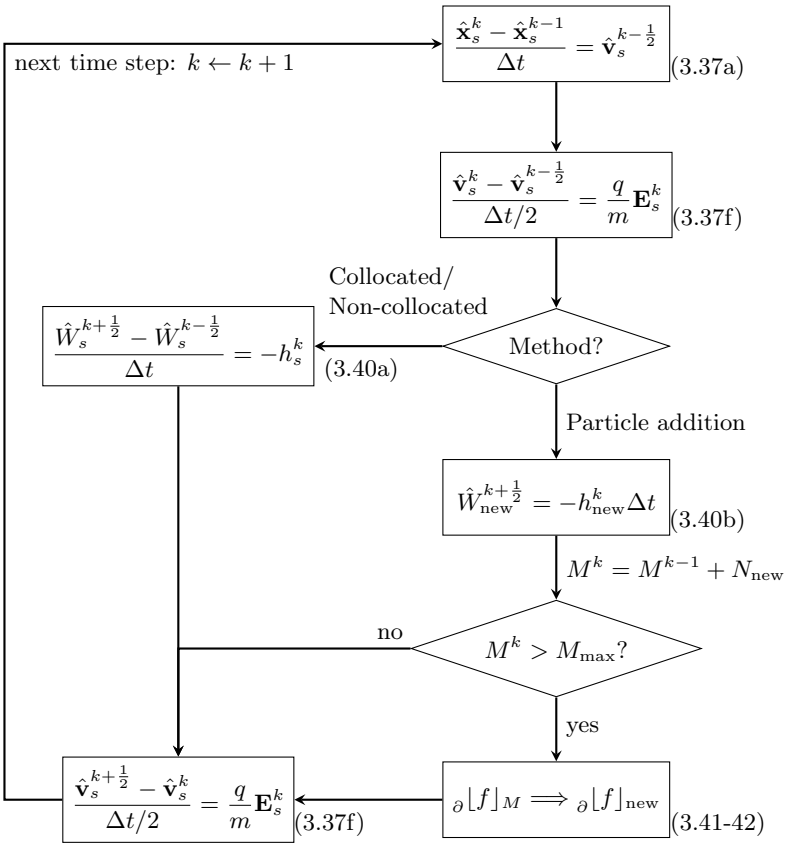


Figure 3.3: Flow diagram for second-order time-reversible integration, including references to equations in the formulation.

Chapter 4

Demonstration model: Debye shielding

Any local charge in a plasma is shielded by the plasma charges distributed on the Debye length scale [1]. We consider the sensitivity of the Debye length to temperature for a quasi-neutral plasma with singly-charged ions: $n_{i0} = n_{e0}$. Following a common approximation [1, 31], the ions are assumed to be uniformly distributed and stationary, so $n_i(x,t) = n_{e0}$ for this demonstration, and the electrons are initially in thermal equilibrium,

$$f(x, v, t = 0) = \frac{n_{e0}}{\sqrt{2\pi} v_T} \exp\left(-\frac{v^2}{2v_T^2}\right), \tag{4.1}$$

with $v_T = \sqrt{k_B T_e / m_e}$, where m_e and T_e are the electron mass and temperature, and k_B is the Boltzmann constant. For (4.1) particle velocities are initialized statistically,

$$v_p(t = 0) = v_T \gamma_p \qquad\qquad p = 1, \ldots, N, \tag{4.2}$$

where γ_p are pseudorandom numbers with standard normal distribution. The plasma is assumed to be homogeneous in y and z and in an x-periodic domain of length L, so

$$f(x + L) = f(L), \qquad \phi(x + L) = \phi(x), \quad \text{and} \quad \mathbf{E}(x + L) = \mathbf{E}(x). \tag{4.3}$$

The length is set $L = 20\lambda_D$, where $\lambda_D = \sqrt{\frac{\epsilon_0 k_B T_0}{n_{e0} q_e^2}}$ is the Debye length at reference temperature T_0, with q_e the electron charge. The nominal external charge distribution is

$$\rho_{\text{ext}}(x) = -\frac{2q_e}{L} + \frac{2q_e}{\sqrt{2\pi} w_q} \exp\left[-\frac{(x - x_c)^2}{2w_q^2}\right], \tag{4.4}$$

with $w_q = 0.1L$ and $x_c = L/2$, which will attract electrons to a Debye-length-scale neighborhood of x_c.

The PIC simulation uses $N = 10^5$ particles, $N_m = 64$ spatial mesh points, and time step $\Delta t = 0.05/\omega_p$, where $\omega_p = \sqrt{\frac{n_e q_e^2}{m_e \epsilon_0}}$ is the electron plasma frequency. Figures 4.1 (a) and (b) visualize the electron distribution and show the shielded potential for $v_T = 1.5 v_0$, where $v_0 = \lambda_D \omega_p$ is the electron thermal velocity at the

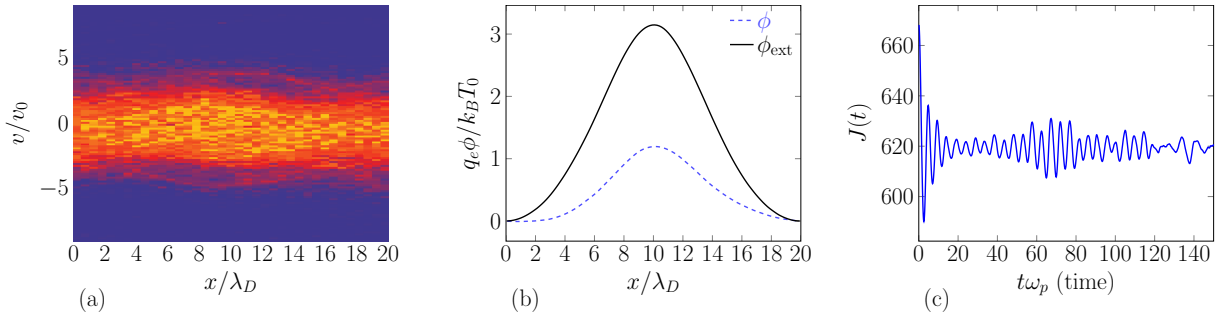


Figure 4.1: (a) Electron distribution histrogram for Debye-shielding configuration (see text), (b) the applied ϕ_{ext} and shielded ϕ potential, and (c) time-dependent integrand of the quantity-of-interest (4.5).

reference temperature.

The QoI is based on the mean electron distribution for time t_f,

$$\mathcal{J} = \frac{1}{t_f}\int_0^{t_f} J(t)\,dt, \tag{4.5a}$$

with

$$J(t) = \int_{-\infty}^{\infty}\int_0^L (x - x_c)^2 f(x, v, t)\,dxdv, \tag{4.5b}$$

which provides a measure of the shielding length scale. It is discretized as

$$\mathcal{J}_D = \frac{1}{N_t}\sum_{k=1}^{N_t}\sum_{p=1}^{N} W_p(x_p^k - x_c)^2. \tag{4.6}$$

The QoI depends on the temperature of the electrons. In the high temperature ($T \to \infty$) limit, the effect of fixed charge is neglected and the electron number density remains uniform in space, which yields maximum QoI $\frac{1}{12}L^3$. For $T \to 0$, the electron number density tracks (per our assumptions) the background fixed charge distribution, and the QoI is minimized $\frac{1}{12}L^3 - \frac{1}{6}L^2 + 2w_q^2$. The time history of $J(t)$ is shown in Figure 4.1 (c), where it oscillates about the equilibrium value as expected due to the statistical character of the PIC approximation. The parameter-of-interest is the initial thermal velocity of the electrons $\theta = v_T/v_0$. For the particle-pdf approach

$$\begin{aligned}\frac{\partial\mathcal{J}}{\partial\theta} &= \frac{1}{t_f}\int_0^{t_f}\int_{-\infty}^{\infty}\int_0^L (x - x_c)^2\,{}_\partial\lfloor f\rfloor\,dx\,dv\,d\tau \\ &\approx \left(\frac{\partial\mathcal{J}}{\partial\theta}\right)_D = \frac{1}{N_t}\sum_{k=1}^{N_t}\sum_{s=1}^{M^k}\hat{W}_s(\hat{x}_s^k - x_c)^2,\end{aligned} \tag{4.7}$$

and for the particle-exact approach

$$\frac{\partial \mathcal{J}_D}{\partial \theta} = \frac{1}{N_t} \sum_{k=1}^{N_t} \sum_{p=1}^{N} 2W_p (x_p^k - x_c)\, {}_\partial\lfloor x_p^k \rfloor. \tag{4.8}$$

The source (3.9) for the particle-pdf approach is

$$\frac{\partial \mathcal{V}_1}{\partial \theta} = \left(\frac{v^2}{v_T^2} - 1 \right) \frac{n_{e0}}{\sqrt{2\pi} v_T^2} \exp\left(-\frac{v^2}{2v_T^2} \right) \delta(t), \tag{4.9}$$

which provides a $t = 0$ initial condition for the sensitivity distribution ${}_\partial\lfloor f \rfloor$. The discrete source for particle-exact approach (3.4) is

$$\frac{\partial \mathcal{V}_1^D}{\partial \theta} = \begin{cases} \gamma_p & \text{at } k = 1 \\ 0 & \text{at } k > 1 \end{cases} \qquad \text{for } p = 1, \ldots, N, \tag{4.10}$$

which is the differentiation of (4.2) with $\theta = v_T$. Particle exactness depends on the exact initial condition so the pseudorandom numbers γ_p are kept for the sensitivity calculation.

Chapter 5

Demonstrations

5.1 Accuracy and regularity

The accuracy of the computed sensitivity (4.8) is assessed against a finite difference of (4.6)

$$\frac{\Delta \mathcal{J}_D}{\Delta \theta} = \frac{\mathcal{J}_D[\theta_0 + \Delta\theta] - \mathcal{J}_D[\theta_0]}{\Delta\theta}, \tag{5.1}$$

which yields a nominal error

$$\epsilon = \left| \frac{\frac{\Delta \mathcal{J}_D}{\Delta \theta} - \frac{\partial \mathcal{J}_D}{\partial \theta}}{\frac{\partial \mathcal{J}_D}{\partial \theta}} \right| = \mathcal{O}(\Delta\theta) + \mathcal{O}\left(\frac{\epsilon_r}{\Delta\theta}\right), \tag{5.2}$$

which is used to assess accuracy. It is deemed accurate when in agreement with the $\mathcal{O}(\Delta\theta)$ truncation error of the first-order finite difference. This will fail as $\mathcal{O}\left(\frac{\epsilon_r}{\Delta\theta}\right)$ due to accumulation of error from finite-precision arithmetic, where ϵ_r quantifies precision [32].

For applications it is also important to assess the size of the neighborhood of θ for which the approximation $\Delta\mathcal{J}_D = \frac{\partial \mathcal{J}_D}{\partial \theta}\Delta\theta + \mathcal{O}(\Delta\theta^2)$ is accurate, since this limits the utility of any $\frac{\partial \mathcal{J}_D}{\partial \theta}$ estimate, and can be reduced by chaos. For a relative error threshold ϵ_p, this size is quantified by a nominal prediction range

$$\Delta\theta_p = \max_{\Delta\theta > 0} \{\Delta\theta \mid \epsilon(\Delta\theta) \leq \epsilon_p\}. \tag{5.3}$$

5.2 Sensitivity calculations

Figure 5.1 shows the increasing error and decreasing prediction range with increasing simulation times and confirms that the accuracy is indeed precision limited. In Figure 5.1 (a), the truncation error curve $\mathcal{O}(\Delta\theta)$ shifts toward smaller $\Delta\theta$ with increasing simulation time, corresponding to the precipitous decrease in the prediction range in Figure 5.1 (b). In essence, the perturbation needs to be machine precision after about 100 plasma periods.

To further evaluate methods, we first construct QoI $\mathcal{J}(\theta)$ in (4.5) by brute-force simulating 1000 cases

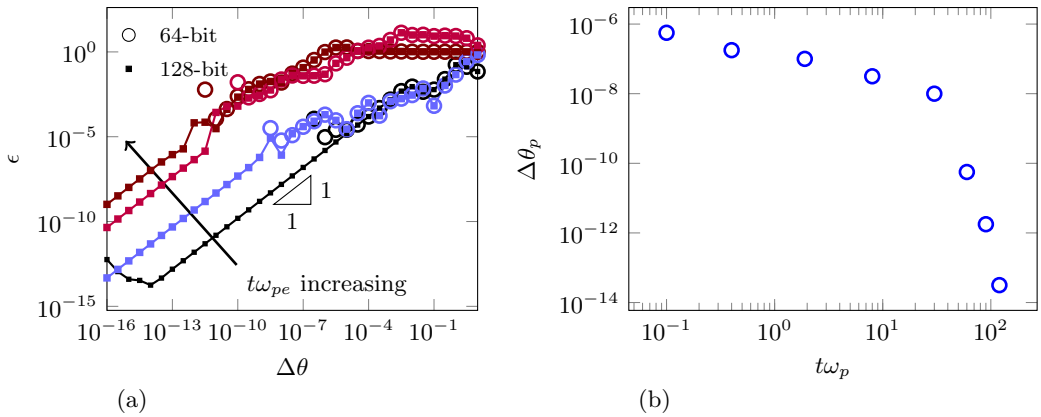


Figure 5.1: (a) Sensitivity error convergence (5.2) for $t\omega_p = 0.1$ to $t\omega_p = 150.0$, for 64- and 128-bit IEEE floating point. (b) Prediction range (5.3) for $\epsilon_p = 10^{-6}$.

for uniformly distributed $\theta_k \in [0.5, 3.5]$, all with the same initial particle distribution. This is done with both the PIC scheme of Section 2.1 and a second-order finite-volume discretization of (2.1) using flux limiters [33]. For the finite-volume discretization, the grid sizes are $\Delta x = 0.02\lambda_D$ and $\Delta v = 0.018v_0$, and the time step for a standard Strang-splitting scheme [33] is set such that

$$\min\left(\frac{\Delta x}{\max(v)\Delta t}, \frac{\Delta v}{\frac{q}{m}\max(E)\Delta t}\right) = 0.5. \tag{5.4}$$

The corresponding $\mathcal{J}(\theta)$ are plotted in Figure 5.2. As expected, PIC solutions approximate the brute-force $\mathcal{J}$ consistently with finite-volume solutions, though with variation due to the chaotic particle dynamics.

Sensitivity for $\theta = 1.5v_0$ is computed using the particle-exact approach of Section 3.1, particle-pdf approaches of Section 3.2, and the same finite-volume discretization for (3.9a). For the particle-pdf approach, the same PIC simulation is used to compute $\mathbf{E_i}$ and $\nabla_{\mathbf{v}} f_{\mathbf{i},\mathbf{i}_v}$, and for our initial comparisons, the non-collocated weight-updating scheme from Section 3.2.4 is used. To evaluate $\mathcal{H}$ in (3.16), $N_m = 64$ uniform mesh cells discretize the velocity domain $v \in [-9v_0, 9v_0]$. The Δt and Δx match the baseline PIC discretization, and $M = 2 \times 10^5$ sensitivity particles are used, initialized uniformly in x and v, for this demonstration to ensure the support condition (3.29) at all times. The same finite-volume discretization parameters are used for the corresponding sensitivity finite-volume discretization of (3.9a).

In Figure 5.2, the particle-exact approach provides accurate but local sensitivity, inconsistent with the sensitivity of the continuum limit. In contrast, the particle-pdf approach, still with the advantages of the particle discretization, matches the smooth sensitivity of the finite-volume results. The local sensitivity distribution in Figure 5.3 shows how the particle-pdf approach provides a coarse (PIC-like) representation

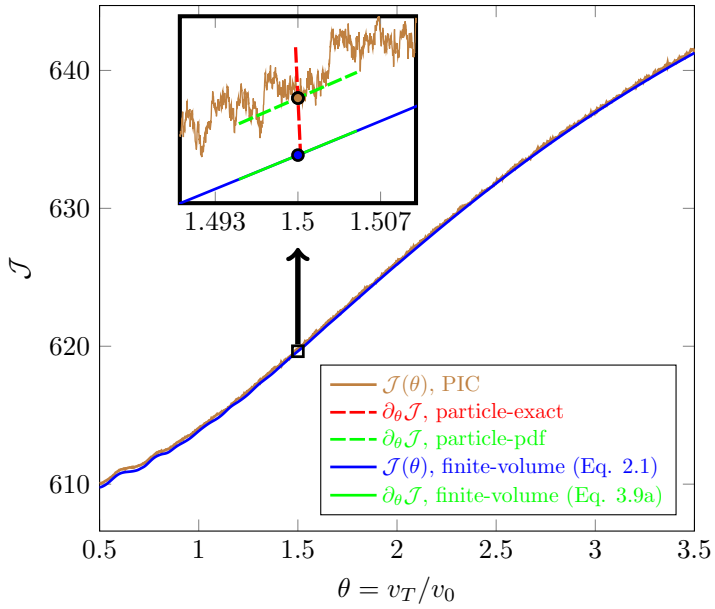
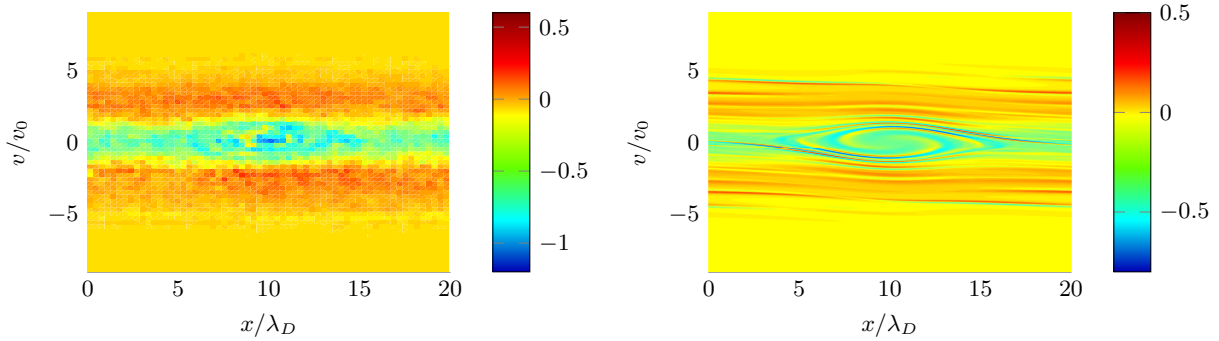


Figure 5.2: $\mathcal{J}(\theta)$ from (4.5) at $\omega_p t = 150.0$. The line segments visualize the computed sensitivities for $\theta = 1.5$.

consistent with the finite-volume solution.

5.3 Evaluation of statistical consistency

Consistency of particle methods is typically assessed in a statistical sense [34–36], with computational particles considered as a sample of a continuum distribution [20, 21, 37]. This approach is useful here to quantify the statistical behavior of the sensitivity, relative to that of the QoI itself. If the sensitivity has significantly more statistical variance than $\mathcal{J}_D$, it will not be useful. Standard deviations of $\mathcal{J}_D$ in (4.6), $\frac{\partial \mathcal{J}_D}{\partial \theta}$ in (4.8),



(a) Sensitivity-pdf (b) Finite volume

Figure 5.3: The sensitivity distribution $_\partial\lfloor f \rfloor$ using (a) the particle-pdf approach, and (b) finite-volume discretization of (3.9a).

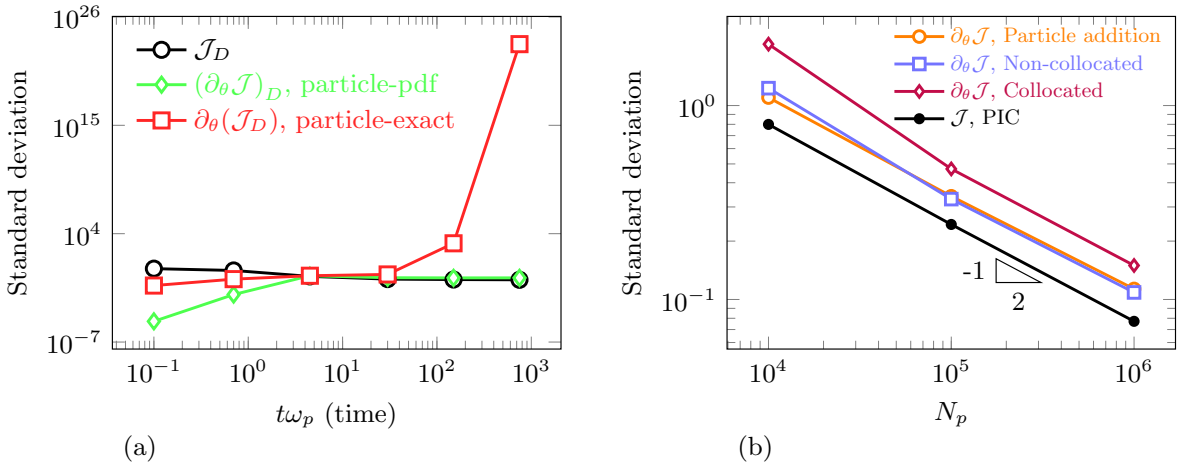


Figure 5.4: Statistical stability for methods applied to the Debye-shielding example: (a) quantity-of-interest and its sensitivity standard deviations, averaged over 10^4 solutions with different random seeds, and (b) sensitivity standard deviations for the particle-pdf variations of Section 3.2, averaged over 10^3 solutions. For the particle-addition scheme, $0.05N$ particles are injected per time step.

and $\left(\frac{\partial \mathcal{J}}{\partial \theta}\right)_D$ in (4.7) are evaluated from ensembles of many realizations and shown in Figure 5.4 (a). The particle-exact sensitivity, despite its accuracy for a single PIC realization, has an exponentially unstable variance. In contrast, the statistical error of the particle-pdf approach tracks the PIC variance for $\mathcal{J}_D$. Figure 5.4 (b) confirms that errors of all three particle-pdf variants decrease as $N^{-1/2}$, as they should, matching PIC behavior.

The chaotic sensitivity of particles is not due to the finite mesh size, time step, or insufficient (or excessive) number of particles, such as discrete-particle noise [20, 21, 38]. It is an intrinsic property of N-body particle dynamics [39]. The numerical parameters (e.g., number of particles and mesh shape functions $S_{\mathbf{x}}$ and $S_{\mathbf{v}}$) are confirmed not to fundamentally change the maximum Lyapunov exponent [40–42]. Appendix B shows how the present conclusions are insensitive to the numerical parameters.

Chapter 6

Assessment of the non-commutability

The non-commutability between the sensitivity derivative and the continuum limit motivated the particle-pdf method developed in Section 2.3, particularly Figure 2.1. This underlying challenge is now evaluated quantitatively for the Debye shielding configuration of Chapter 4.

6.1 Error metrics

The metrics we use are motivated by the assertion to be investigated, which is considered first. A particle distribution $n_N(\mathbf{x};\theta)$ is discretized by N particles,

$$n_N(\mathbf{x};\theta) = \sum_{p=1}^{N} \frac{1}{N}\delta(\mathbf{x} - \mathbf{x}_p), \tag{6.1}$$

where θ is the usual parameter-of-interest as introduced in (2.13) and $\mathbf{x}_p(t;\theta)$ are the particle positions governed by (2.5). The $n_N(\mathbf{x};\theta)$ distribution approximates a corresponding continuum number density $n(\mathbf{x};\theta)$ with an error ϵ_N due to finite N,

$$n_N(\mathbf{x};\theta) = n(\mathbf{x};\theta) + \epsilon_N, \tag{6.2}$$

where $\epsilon_N \to 0$ for $N \to \infty$ [43]. This limit is useful for analysis even though a continuum is in truth only a model for a huge but finite-N system, one which is far too large to simulate in most cases. In this section, intuitively, it is equivalent to considering the number of particles required for a stable finite-difference-approximation of sensitivity with a parameter perturbation $\Delta\theta$.

$\Delta\theta$ is introduced to quantify the effect of commutation between the $\Delta\theta \to 0$ limit of differentiation and the $N \to \infty$ limit that defines a continuum. The perturbation $\Delta\theta$ alters the original particle trajectories $\mathbf{x}_p(t;\theta)$ to new trajectories,

$$\mathbf{x}'_p(t;\theta + \Delta\theta) = \mathbf{x}_p(t;\theta) + \Delta\mathbf{x}_p(t), \tag{6.3}$$

which leads to a new particle distribution,

$$n'_N(\mathbf{x};\theta+\Delta\theta)=\sum_{p=1}^{N}\frac{1}{N}\delta(\mathbf{x}-\mathbf{x}'_p), \tag{6.4}$$

which, from the same viewpoint of (6.2), approximates a continuum density $n'(\mathbf{x};\theta+\Delta\theta)$ with corresponding error ϵ'_N,

$$n'_N(\mathbf{x};\theta+\Delta\theta)=n'(\mathbf{x};\theta+\Delta\theta)+\epsilon'_N. \tag{6.5}$$

As for ϵ_N, $\epsilon'_N\to 0$ for $N\to\infty$. While $\Delta\mathbf{x}_p$ would lead to standard Lyapunov exponent analysis of N-body dynamics [40–42], our sensitivity objectives involve Δn_N,

$$\Delta n_N(\mathbf{x})=n'_N(\mathbf{x};\theta+\Delta\theta)-n_N(\mathbf{x};\theta)=\Delta n(\mathbf{x})+\Delta\epsilon_N, \tag{6.6}$$

where $\Delta n(\mathbf{x})\equiv n'(\mathbf{x};\theta+\Delta\theta)-n(\mathbf{x};\theta)$ and $\Delta\epsilon_N\equiv\epsilon'_N-\epsilon_N$.

Based on (6.6), the sensitivity of particle distribution n_N to θ has two contributions:

$$\frac{\Delta n_N}{\Delta\theta}=\frac{\Delta n}{\Delta\theta}+\frac{\Delta\epsilon_N}{\Delta\theta}. \tag{6.7}$$

The non-commutability between $\Delta\theta\to 0$ and $N\to\infty$ is manifest in the second: $\Delta\epsilon_N/\Delta\theta$. Since $\Delta\epsilon_N$ will reflect the divergent trajectories of the particles due to $\Delta\theta$, we have

$$\Delta\epsilon_N\sim\Delta\theta\exp(\lambda t), \tag{6.8}$$

where λ is the maximum Lyapunov exponent of the N-body PIC dynamics (2.5). This is true even though

$$\lim_{N\to\infty}\Delta\epsilon_N=0, \tag{6.9}$$

since both $\epsilon_N\to 0$ and $\epsilon'_N\to 0$ for $N\to\infty$. For the particle-exact method, the continuum limit distributions are approximated with N particles, so $\Delta\epsilon_N>0$ and $\Delta\theta\to 0$ yields the spurious sensitivity seen in Section 5.2. Whereas, the particle-pdf approach of Section 3.2 introduces particle discretization after differentiation. The discrete gradient $\left(\frac{\partial n}{\partial\theta}\right)_N$ still has error $\hat{\epsilon}_N$,

$$\left(\frac{\partial n}{\partial\theta}\right)_N=\frac{\partial n}{\partial\theta}+\hat{\epsilon}_N, \tag{6.10}$$

however, there is no subsequent differentiation of this error term as for $\Delta\epsilon_N/\Delta\theta$ in (6.7). In this way, the particle-pdf method can stably compute the sensitivity so long as the continuum-limit sensitivity itself $\Delta n/\Delta\theta$ remains Lyapunov stable, requiring

$$\Delta n \sim \Delta\theta. \tag{6.11}$$

In this case, (6.10) is well-behaved with $\hat{\epsilon}_N \to 0$ as $N \to \infty$.

The Lyapunov instability (6.8) and $\Delta\epsilon_N$ convergence (6.9), which express the consequence of non-commutability, are now quantified for the model system. Direct calculation of $\Delta\epsilon_N$ is, in general, not possible since $n(\mathbf{x})$ and $n'(\mathbf{x})$ are unavailable. Instead, we estimate $\Delta\epsilon_N$ via Δn_N, assuming (6.11) so that $\Delta\epsilon_N$ dominates Δn_N in (6.7). This assumption will be confirmed via the finite-volume solution of the Debye shielding configuration of Chapter 4. The behavior of (6.8), with $\Delta\epsilon_N$ diverging in time, and (6.9) with $\Delta\epsilon_N$ converging with N correspond to different temporal regimes of its evolution. For $\lambda t \lesssim 1$, $\Delta\epsilon_N$ diverges exponentially along with particle trajectories $\Delta\mathbf{x}_p$; for $\lambda t \gg 1$, it saturates in time at a value that itself converges for $N \to \infty$.

To assess this in computations, we introduce metrics for quantifying Δn_N and Δn. For this purpose, it is sufficient to consider a generic linear functional QoI (3.1),

$$\mathcal{J}[n;\theta] = \int J(\mathbf{x}) n(\mathbf{x})\, d\mathbf{x}, \tag{6.12}$$

where $\mathcal{J}$ is assumed smooth, consistent with limiting $J(\mathbf{x}) : \mathbf{X} \to \mathbb{R}$ as a generic Lipschitz function with unity Lipschitz constant [44]. This property, which enables firmer conclusions, is justified when $\partial_\theta \mathcal{J}$ is more sensitive to $\partial_\theta n_N(\mathbf{x})$ than $J(\mathbf{x})$, as expected for chaotic dynamics. The perturbation of $\mathcal{J}_D \equiv \mathcal{J}[n_N;\theta]$ is estimated from the particle perturbations $\Delta\mathbf{x}_p$,

$$\begin{aligned}
\Delta\mathcal{J}_D &= \int J(\mathbf{x})\Delta n_N(\mathbf{x})\, d\mathbf{x} = \frac{1}{N}\sum_{p=1}^{N}\int J(\mathbf{x})\left[\delta(\mathbf{x}-\mathbf{x}'_p) - \delta(\mathbf{x}-\mathbf{x}_p)\right]\, d\mathbf{x} \\
&= \frac{1}{N}\sum_{p=1}^{N}\left[J(\mathbf{x}'_p) - J(\mathbf{x}_p)\right] = \frac{1}{N}\sum_{p=1}^{N}\frac{J(\mathbf{x}'_p) - J(\mathbf{x}_p)}{\Delta\mathbf{x}_p}\Delta\mathbf{x}_p \\
&\leq \frac{1}{N}\sum_{p=1}^{N}|\Delta\mathbf{x}_p| \equiv \overline{\Delta\mathbf{x}_p},
\end{aligned} \tag{6.13}$$

where the $\leq$ bound derives from the Lipschitz-1 property of $J(\mathbf{x})$. This leads to a mean particle perturbation $\overline{\Delta\mathbf{x}_p}$ [40–42], which grows in time and saturates at the scale of the system L. For long times, $\Delta\mathcal{J}_D/\Delta\theta \sim L/\Delta\theta$ independent of N, which is an over-estimate and does not necessarily support (6.9).

To provide a better estimate of QoI sensitivity we introduce bounded-Lipschitz distance [43], which quantifies $\Delta\mathcal{J}$ without distinguishing particles,

$$L_b(\Delta n_N) = \sup_{J\in\mathcal{D}} \left| \int J(\mathbf{x}) n'_N(\mathbf{x})\, d\mathbf{x} - \int J(\mathbf{x}) n_N(\mathbf{x})\, d\mathbf{x} \right|, \tag{6.14}$$

and is equivalent to Wasserstein-1 distance [43, 45],

$$W_1(\Delta n_N) = \inf_{\sigma\in\Sigma} \iint |\mathbf{x}-\mathbf{x}'| \sigma(\mathbf{x},\mathbf{x}')\, d\mathbf{x} d\mathbf{x}', \tag{6.15}$$

where $\sigma : \mathbf{x}\times\mathbf{x}' \to [0,1]$ is any joint distribution with marginals $n_N(\mathbf{x})$ and $n'_N(\mathbf{x}')$. For the purpose of this analysis, the argument of L_b and W_1 is simplified from $L_b(n'_N, n_N)$ and $W_1(n'_N, n_N)$ in their standard definitions [43, 45]. In essence, the W_1-distance is independent of the particle labels; the difference $|\mathbf{x}-\mathbf{x}'|$ is based on the closest perturbed particle $\mathbf{x}'_q$ to the original trajectory $\mathbf{x}_p$ for all p and q.

Using (6.15) enables analysis of the relationship between Δn and Δn_N, thus the assertion of (6.8) and (6.9) in terms of $\overline{\Delta\mathbf{x}_p}$ as it grows in time. This is possible since $\overline{\Delta\mathbf{x}_p}$ corresponds to the integral in (6.15) with a specific joint distribution σ_P that does not reflect the indistinguishability of particles:

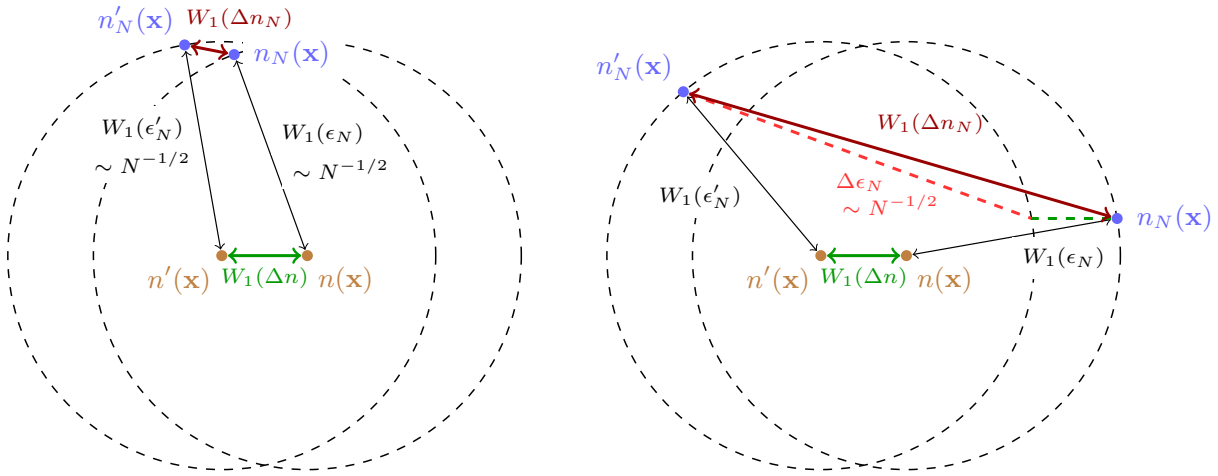
$$\sigma_P(\mathbf{x},\mathbf{x}') = \sum_{p=1}^{N} \frac{1}{N} \delta(\mathbf{x}-\mathbf{x}_p)\delta(\mathbf{x}'-\mathbf{x}'_p) \tag{6.16a}$$

$$\iint |\mathbf{x}-\mathbf{x}'| \sigma_P(\mathbf{x},\mathbf{x}')\, d\mathbf{x} d\mathbf{x}' = \sum_{p=1}^{N} \frac{1}{N} |\Delta\mathbf{x}_p| \equiv \overline{\Delta\mathbf{x}_p}. \tag{6.16b}$$

For $\overline{\Delta\mathbf{x}_p} \ll L/N$, σ_P is the minimizer of the integral in (6.15), meaning (6.13) is equivalent to (6.14). Therefore, as the particle perturbation $\overline{\Delta\mathbf{x}_p}$ grows exponentially, $W_1(\Delta n_N)$ is anticipated to also grow at the same rate as asserted in (6.8),

$$W_1(\Delta n_N) \sim \overline{\Delta\mathbf{x}_p} \sim \Delta\theta \exp(\lambda t) \qquad \text{when} \qquad \Delta\mathbf{x}_p \lesssim \frac{L}{N}. \tag{6.17}$$

This changes when $\mathbf{x}'_p$ is no longer the perturbed particle closest to $\mathbf{x}_p$, which is anticipated for $\Delta\mathbf{x}_p \gtrsim L/N$. Then σ_P is no longer the minimizer, and $\overline{\Delta\mathbf{x}_p} > W_1(\Delta n_N)$. The bound of $W_1(\Delta n_N)$ can be found



(a) $\Delta\mathbf{x}_p \sim \Delta\theta$ (b) $\Delta\mathbf{x}_p \gg \dfrac{L}{N}$

Figure 6.1: Relation between 4 distributions $n'(\mathbf{x})$, $n(\mathbf{x})$, $n'_N(\mathbf{x})$ and $n_N(\mathbf{x})$ in a W_1-metric space. (a) At early times, when particle perturbation $\Delta\mathbf{x}_p \sim \Delta\theta$, the corresponding perturbation in particle distribution, $W_1(\Delta n_N)$, is similar to continuum-limit perturbation $W_1(\Delta n)$. (b) At long times, when $\Delta\mathbf{x}_p \gg L/N$, finite-$N$ error perturbation $\Delta\epsilon_N$ (the red dashed line) exceeds the continuum-limit perturbation $W_1(\Delta n)$ (green dashed line).

via triangle inequality for Wasserstein distance [46],

$$\begin{aligned} W_1(\Delta n_N) &\equiv W_1\left[(n'_N - n') - (n_N - n) + (n' - n)\right] \\ &\leq W_1(\epsilon_N) + W_1(\epsilon'_N) + W_1(\Delta n), \end{aligned} \tag{6.18}$$

where $W_1(\epsilon_N)$ and $W_1(\epsilon'_N)$ are the Wasserstein distance between an empirical sample and true distribution, for which convergence follows a central-limit theorem [47, 48],

$$W_1(\epsilon_N) \sim W_1(\epsilon'_N) \sim N^{-\frac{1}{2}}, \tag{6.19}$$

as seen in Figure 5.4 (b) for our PIC discretization.

This bound corresponds to the saturation limit of $W_1(\Delta n_N)$. For example consider $n'(\mathbf{x})$, $n(\mathbf{x})$, $n'_N(\mathbf{x})$ and $n_N(\mathbf{x})$ on an W_1-metric space, as shown in Figure 6.1. In Figure 6.1 (a), when $\Delta\mathbf{x}_p$ is small for $\lambda t \lesssim 1$, $n'_N(\mathbf{x})$ and $n_N(\mathbf{x})$ are well aligned, so $W_1(\Delta n_N) \sim W_1(\Delta n) \sim \Delta\theta$. As $\Delta\mathbf{x}_p$ grows, $W_1(\Delta n_N)$ also grows so the alignment breaks down, though $n'_N(\mathbf{x})$ and $n_N(\mathbf{x})$ still well-approximate $n(\mathbf{x})$ and $n'(\mathbf{x})$ with $W_1 \sim N^{-1/2}$ as in (6.19). If $W_1(\Delta n)$ is small enough, we also have $W_1(\Delta n_N) \sim N^{-1/2}$, as shown in

Figure 6.1 (b). For long times, $\lambda t \gg 1$, the limit of these perturbation metrics yield

$$W_1(\Delta n_N) = W_1(\Delta n) + \mathcal{O}[N^{-1/2}] \qquad \text{when} \qquad \Delta \mathbf{x}_p \sim L, \tag{6.20}$$

which shows the N-dependent $\Delta\epsilon_N$, confirming (6.9). In essence, (6.17) and (6.20) are re-statements of (6.8) and (6.9) for W_1, for increasing $\Delta \mathbf{x}_p$ rather than t.

Dividing (6.20) by $\Delta\theta$ for computing sensitivity,

$$\frac{W_1(\Delta n_N)}{\Delta\theta} = \frac{W_1(\Delta n)}{\Delta\theta} + \frac{\mathcal{O}\left[N^{-1/2}\right]}{\Delta\theta}. \tag{6.21}$$

The challenge of particle-exact sensitivity, is explicit in $\mathcal{O}[N^{-1/2}]/\Delta\theta$ in (6.21). The particle-pdf sensitivity is useful so long as the continuum-limit sensitivity is stable, with

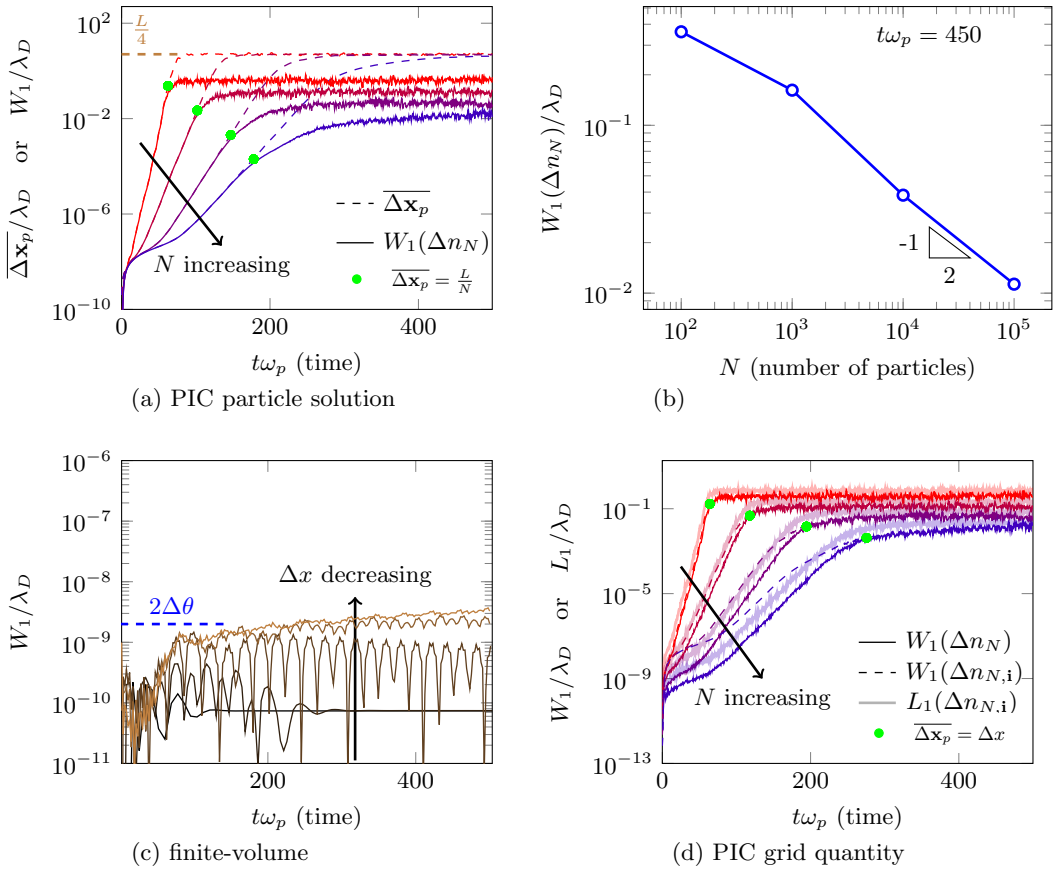
$$W_1(\Delta n) = \lim_{N\to\infty} W_1(\Delta n_N) \sim \Delta\theta, \tag{6.22}$$

which is a re-statement of the assertion (6.11). Assuming (6.22) that $\Delta\epsilon_N$ dominates $W_1(\Delta n_N)$, the Lyapunov instability of $\Delta\epsilon_N$ (6.8) and its saturation limit (6.9) can be now assessed numerically with (6.17) and (6.20) for $W_1(\Delta n_N)$.

6.2 Application to the Debye shielding model

The Lyapunov instability of $W_1(\Delta n_N)$ (6.17) and its saturation (6.20) are assessed quantitatively. For $\theta = 1.5v_0$ and $\Delta\theta = 10^{-9}v_0$, $\overline{\Delta \mathbf{x}_p}$ in (6.13) and W_1-distance (6.15) are calculated and compared for different N. The Wasserstein-1 distance [43, 45], was calculated with dual-simplex algorithm in MATLAB [49].

Figure 6.2 shows the growth and the satuation of $W_1(\Delta n_N)$ that confirms (6.17) and (6.20). In Figure 6.2 (a), $\overline{\Delta \mathbf{x}_p}$ grows to the same $\overline{\Delta \mathbf{x}_p} \sim L$ limit regardless of N, though at an N-dependent rate. In contrast, $W_1(\Delta n_N)$ growth deviates from $\overline{\Delta \mathbf{x}_p}$ starting when it exceeds inter-particle scale L/N, reaching different limits depending on N. Figure 6.2 (b) shows the $\mathcal{O}[N^{-1/2}]$ scaling of the $W_1(\Delta n_N)$ saturation limit, which is consistent with (6.20). The continuum-limit perturbation $W_1(\Delta n)$, if it grows at all, does not dominate the N-scaling of $W_1(\Delta n_N)$ in Figure 6.2 (b). For comparison, W_1 is calculated for the finite-volume simulation of Section 5.2, to assess (6.22). Due to the dissipative upwind scheme and TVD flux limiter [33], this provides a lower bound for $W_1(\Delta n)$. In Figure 6.2 (c), $W_1(\Delta n_{\mathrm{FV}})$ converges to $\simeq 2\Delta\theta$ with



(a) PIC particle solution (b)

(c) finite-volume (d) PIC grid quantity

Figure 6.2: (a) Parameter perturbation $\Delta\theta$ impact on $\overline{\Delta \mathbf{x}_p}$ and $W_1(\Delta n_N)$ in the Debye shielding model for $N = \{10^2, 10^3, 10^4, 10^5\}$ particles (from red to blue), and (b) $W_1(\Delta n_N)$ versus the number of particles at $\omega_p t = 450$; (c) the corresponding $\Delta\theta$ impact on $W_1(\Delta n_{\mathrm{FV}})$ in finite-volume simulation for $N_g = \{2^6, 2^7, 2^8, 2^9, 2^{10}\}$ grid points (from black to brown); and (d) the corresponding $\Delta\theta$ impact on $L_1(\Delta n_{N,\mathbf{i}})$ in PIC for $N = \{10^2, 10^3, 10^4, 10^5\}$ particles (from red to blue) for $N_g = 64$.

mesh refinement. Together, we see

$$W_1(\Delta n_{\mathrm{FV}}) \sim \Delta\theta \lesssim W_1(\Delta n) < W_1(\Delta n_N) \sim N^{-1/2} \ll \overline{\Delta \mathbf{x}_p} \sim L, \tag{6.23}$$

which confirms (6.20) and indicates (6.22) for Debye shielding configuration.

Our focus has been on $N \to \infty$ behavior. However PIC involves additional mesh discretizations. To confirm that this does not alter conclusions, two perturbation metrics are calculated. First, an L_1-norm of mesh-based number density is computed,

$$L_1(\Delta n_{N,\mathbf{i}}) \equiv \sum_{\mathbf{i}\in\mathrm{mesh}}^{N_{\mathbf{i}}} \left| n'_{N,\mathbf{i}} - n_{N,\mathbf{i}} \right| \ \Delta\mathbf{x}, \tag{6.24}$$

which is a standard metric for Lyapunov exponent of Eulerian fields [50]. Also the same W_1 distance (6.15) for $\Delta n_{N,\mathbf{i}}$ is computed to estimate the QoI sensitivity. Figure 6.2 (d) shows the growth of $L_1(\Delta n_{N,\mathbf{i}})$ and $W_1(\Delta n_{N,\mathbf{i}})$ for different N and constant Δx, comparing with $W_1(\Delta n_N)$ from particle distributions. Once $\overline{\Delta \mathbf{x}_p}$ exceeds the mesh size Δx, $W_1(\Delta n_{N,\mathbf{i}})$ saturates near the same limit of its particle counterpart $W_1(\Delta n_N)$. This indicates that mesh interpolation may affect the perturbation growth for short times, but for long times the saturation limit remains subject to the density of the finite-N particle representation. $L_1(\Delta n_{N,\mathbf{i}})$ also grows similarly to $W_1(\Delta n_N)$.

Chapter 7

Computational intensity

7.1 Empirical cost of particle-pdf approaches compared to PIC

Computation costs for the Debye shielding configuration of Chapter 4 are measured for all three particle-pdf variants on an Intel Xeon E5-2695 2.1GHz CPU with 128GB of main memory. Operations match those of the corresponding PIC simulation except for computing sensitivity, which correspond to (3.38) through (3.41). As discussed in Section 3.2.2, these stages involve $(\mathbf{x}, \mathbf{v})$ phase-space interpolation, though the actual cost to evaluate interpolation coefficients is comparable to the original PIC algorithm, since the coefficients for x interpolation are re-used from the previous stages and that scales linearly with N, not the size of the phase-space mesh. In Figure 7.1, the total time costs were 2.96 times that of the corresponding PIC simulation for the non-collocated scheme, 2.10 times for the collocated scheme, and 1.89 times for the particle-addition scheme. The collocated scheme does not require the advection stages (3.37a), (3.37f), and (3.37e), and also computation cost in charge assignment (3.37b) is reduced since interpolation coefficients can be reused. For the particle addition scheme, new particles and their interpolation coefficients are pre-computed and stored in a pre-processing stage, the number density evaluation (3.39a) is omitted, and the cost in particle source strength evaluation (3.39b) is reduced. This difference is independent of the number of particles, as confirmed in Figure 7.2. The operation counts in each numerical scheme are listed in detail in Table 7.1.

7.2 Computational cost scaling

Table 7.2 shows the operations per time step of the particle-pdf approach and compares it to the finite-volume method [20]. While the major cost of the finite-volume method comes from advection and f velocity gradient, which scales with the mesh size, for the particle methods it is the interpolation between particle and mesh, which scales with the number of particles. Since a large number of particles per cell are typically used in PIC simulation ($N/N_m^d \gg 1$), the mesh dependence of the Poisson solver (3.37c) and E-field evaluation (3.37d) are not expected to be significant for applications. As a result, the operation count scales with the

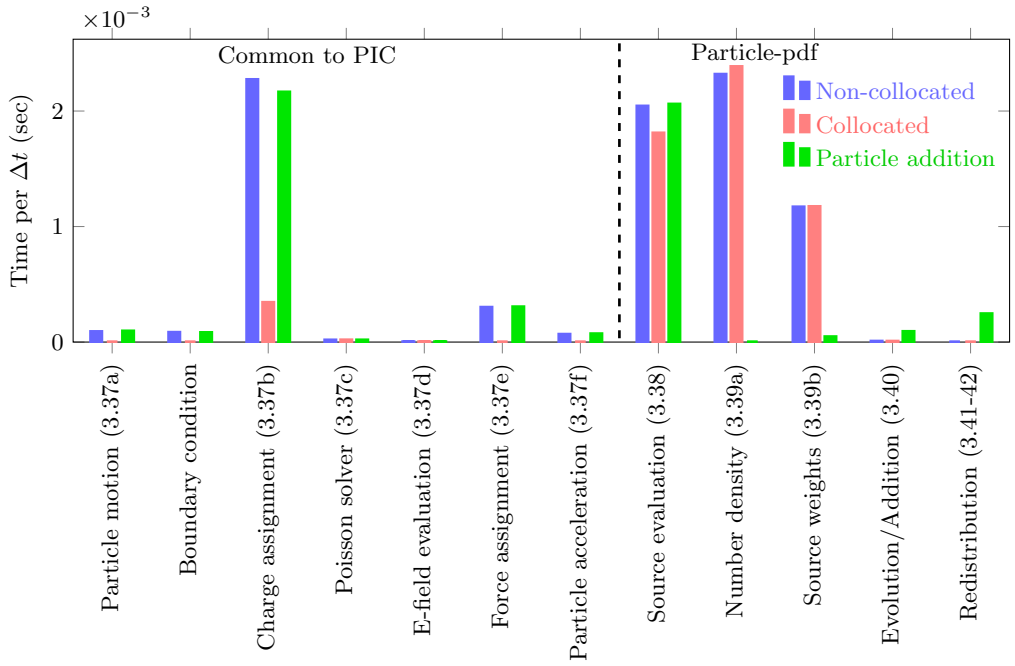


Figure 7.1: Computational time per function in particle-pdf schemes for the Debye shielding model and $N = 10^5$. Operations shared with the original PIC are indicated.

Stage	**Collocated**	**Non-collocated**	**Particle addition**
Particle advection (3.37a), (3.37f)	0	$\mathcal{O}[dN]$	
Charge assignment (3.37b)	$\mathcal{O}[dl^dN]$	$\mathcal{O}[(C_1l + l^d)dN]$	
Poisson solver (3.37c)	$27N_m^d$		
E-field evaluation (3.37d)	$2N_m^d$		
Force assignment (3.37e)	0	$\mathcal{O}[(l+1)^ddN]$	
Source evaluation (3.38)	$\mathcal{O}[(C_2ld + l^{2d}d^2)N]$		
Number density (3.39a)	$\mathcal{O}[(C_3l + l^{2d})dN]$		
Source strength (3.39b)	$\mathcal{O}[(l+1)^{2d}N]$		
Evolution/Addition (3.40)	$\mathcal{O}[N]$		
Redistribution (periodic) (3.41–3.42)	0		$\mathcal{O}[(C_4ld + l^{2d}d^2)N]$

Table 7.1: Operation count scaling with (average) number of particles N, mesh size (per coordinate direction) N_m, B-spline order l, and space dimensionality d. The C's are constants for particle-mesh interpolation. Conjugate Gradient method for Poisson solver and centered finite-difference stencils for E-field gradient are provided, although both PIC and finite-volume methods use the same algorithms for them so it is less important for comparison.

number of particles N. However, a naive implementation of the phase-space mesh will introduce memory challenges. While we do not consider this point in detail, we anticipate that this memory issue could be addressed by standard parallelization techniques for PIC [51]. Additionally, a sparse representation of the phase-space mesh might significantly reduce the memory footprint.

Stage	**Particle-pdf (weight-evolution)**	**Finite-volume**
Particle advection	$\mathcal{O}[dN]$	$2C_1N_m^{2d}$
Charge assignment	$\mathcal{O}\big[dl^dN\big]$	$C_2N_m^{2d}$
Poisson solver	$27N_m^d$	$27N_m^d$
E-field evaluation	$2N_m^d$	$2N_m^d$
Force assignment	$\mathcal{O}\big[l^ddN\big]$	0
Source evaluation	$\mathcal{O}\big[l^{2d}d^2N\big]$	$\mathcal{O}\big[(3d+C_1)N_m^{2d}\big]$
Number density	$\mathcal{O}\big[l^{2d}dN\big]$	0
Particle source	$\mathcal{O}\big[l^{2d}N\big]$	0
Source integration	$2N$	N_m^{2d}
Overall	$\sim (Cl+l^{2d})dN$	$\sim (d+C)N_m^{2d}$

Table 7.2: Operation count scaling per time step with (average) number of particles N, mesh size (per coordinate direction) N_m, B-spline order l, and space dimensionality d. The C's are constants. A conjugate gradient method is assumed for the Poisson solver and centered finite-difference stencils assumed for E-field gradient are provided, although both PIC and finite-volume methods use the same algorithms for them so these details are unimportant.

Figure 7.2 (a) confirms the scaling per time step estimates in Table 7.2. Since typical plasma simulations are implemented in higher dimensions ($d > 1$), and that the particles in general efficiently represent the distribution while most of the grid points lie in empty phase-space of plasma, this scaling points to the advantage of the particle-pdf approach over a mesh-based method (finite-volume) for the same cases for which PIC would be advantageous. Using a mesh twice as large as in Chapter 4 does not affect the cost of PIC and particle-pdf approach shown in Figure 7.2. Moreover, the time-step restriction (5.4) is only associated with the finite-volume scheme. For this reason, the total simulation time the finite-volume method increases faster than for particle-pdf methods, as shown in Figure 7.2 (b).

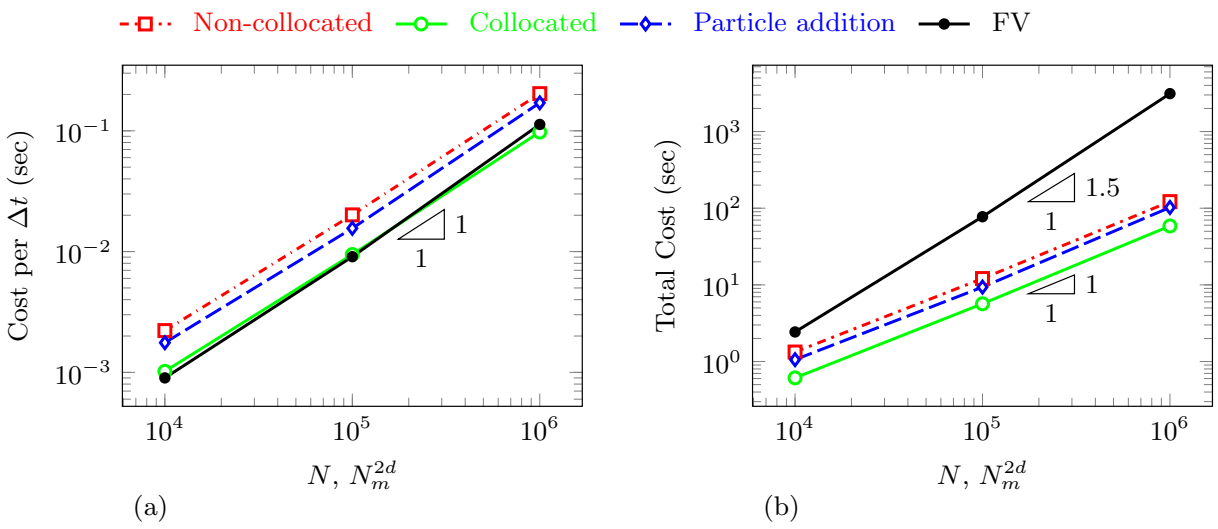


Figure 7.2: Computational time (a) per time step and (b) the total solution.

Chapter 8

Adaptive example: a sheath edge

This common configuration includes several additional components: non-periodic boundary conditions, non-uniform ion distribution, and a model for ionization. An adaptive variant of the weight-updating method and the particle-addition method can be used advantageously.

8.1 Baseline configuration

The mobility difference between electrons and ions disrupts quasi-neutrality within a few Debye lengths of an electrode (Figure 8.1). Following a standard PIC implementation [31], we consider the sensitivity of the potential drop in the resulting non-equilibrium sheath region [1, 31, 52] to the temperature ratio between ions T_i and electrons T_e far from the electrode for a neutral plasma with singly-charged ions: $n_{i0} = n_{e0} = n_0$. Both ions and electrons are initially uniformly distributed in thermal equilibrium, so

$$f_z(x, v, t = 0) = \frac{n_0}{\sqrt{2\pi} v_{T,z}} \exp\left(- \frac{v^2}{2 v_{T,z}^2} \right) \qquad \text{for } z = i \text{ or } e, \tag{8.1}$$

with $v_{T,z} = \sqrt{k_B T_z / m_z}$. Equilibrium is maintained far from the wall with an ionization source [31], which appears in the Vlasov equation (2.1a) as

$$\frac{\partial f_z}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f_z + \frac{q_z}{m_z} \mathbf{E} \cdot \nabla_{\mathbf{v}} f_z = S_z \qquad \text{for } z = i \text{ or } e. \tag{8.2}$$

The velocity distribution of each species' source is the flux for the corresponding Maxwellian,

$$S_z = \frac{1}{L_s} \frac{|v|}{2 v_{T,z}^2} \exp\left(- \frac{v^2}{2 v_{T,z}^2} \right) \int f_i v \big|_{x=0} \, dv \qquad \text{for } z = i \text{ or } e, \tag{8.3}$$

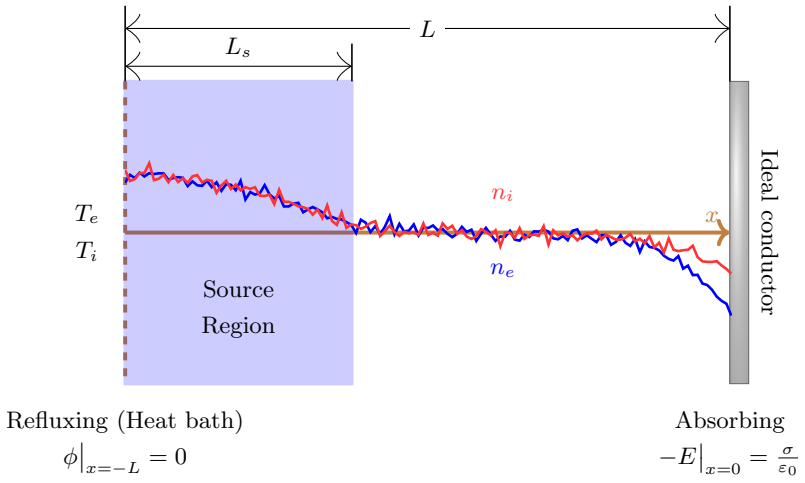


Figure 8.1: Diagram of one-dimensional sheath edge formation

which is constructed such that the rate matches the uptake of ions at the $x = 0$ wall. At $x = -L$ both ions and electrons are reflected (often termed refluxed) at the equilibrium temperature:

$$f_z\big|_{x=-L,v>0} = \frac{1}{v_{T,z}^2} \exp\left(-\frac{v^2}{2v_{T,z}^2}\right) \int_{v<0} f_z v\big|_{x=-L}\, dv \qquad \text{for } z = i \text{ or } e. \tag{8.4}$$

All ions and electrons are perfectly absorbed at $x = 0$ on the electrode, which changes its surface charge as

$$\frac{d\sigma_s}{dt} = q_e \int f_e\big|_{x=0} v\, dv + q_i \int f_i v\big|_{x=0}\, dv. \tag{8.5}$$

The surface charge σ_s provides a boundary condition at $x = 0$ for the Poisson equation (2.1b), and the potential at $x = -L$ is set zero

$$\left.\frac{\partial \phi}{\partial x}\right|_{x=0} = \frac{\sigma_s}{\epsilon_0} \quad \text{and} \quad \phi\big|_{x=-L} = 0. \tag{8.6}$$

The distributions of electrons and ions in Figure 8.2 shows their significantly different thermal velocities.

8.2 Sensitivity calculation

Procassini et al. [31] discussed the dependence of the potential drop to the temperature ratio $\theta_\tau = T_e/T_i$, which motivates our QoI

$$\mathcal{J} = \frac{1}{t_f - t_i} \int_{t_i}^{t_f} [\phi(0) - \phi(L;t)]\, dt, \tag{8.7}$$

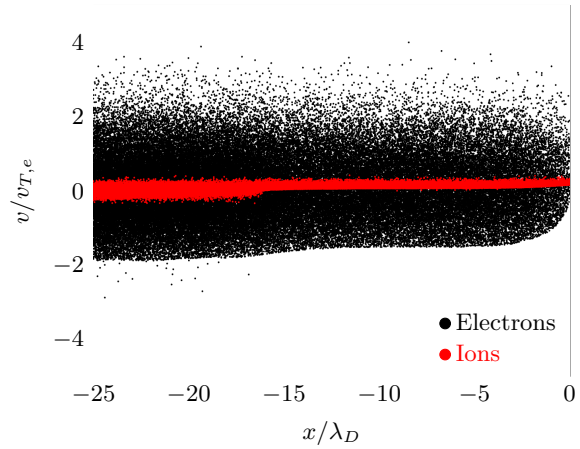


Figure 8.2: An equilibrium instantaneous phase-space distribution of electrons (black) and ions (red) for the sheath edge of Figure 8.1.

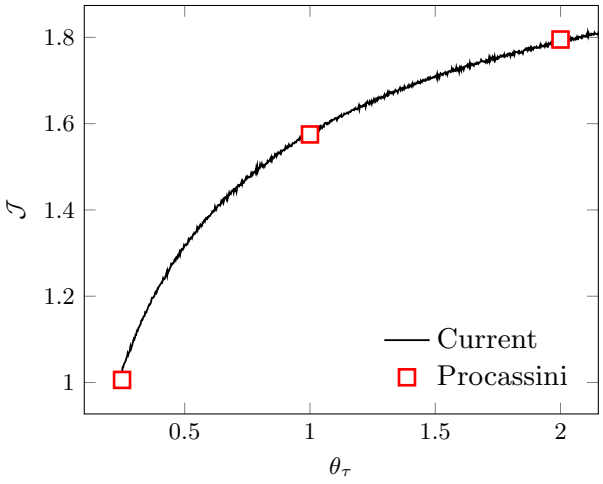


Figure 8.3: Brute force QoI (8.7) dependence on $\theta_\tau = \frac{T_e}{T_i}$.

with $t_i = 600/\omega_{p,e}$ and $t_f = 1200/\omega_{p,e}$. Its dependency on the temperature ratio θ_τ is shown in Figure 8.3 for exhaustive evaluation of $\mathcal{J}(\theta_\tau)$ for 1000 values of θ_τ. The sensitivity to θ_τ we seek is

$$\frac{\partial \mathcal{J}}{\partial \theta_\tau} = -\frac{1}{t_f - t_i} \int_{t_i}^{t_f} {}_\partial\lfloor\phi\rfloor(L;t)\, dt. \tag{8.8}$$

The corresponding sensitivity source term in (3.11) is

$$\mathcal{H}_z = \frac{q_z}{m_z}\, {}_\partial\lfloor \mathbf{E} \rfloor \cdot \nabla_{\mathbf{v}} f_z - \frac{\partial S_z}{\partial \theta_\tau} \qquad \text{for } z = e \text{ or } i, \tag{8.9}$$

where the sensitivity of ionization source terms are

$$\frac{\partial S_e}{\partial \theta_\tau} = \frac{|\tilde{v}|}{2L_s} \exp\left(-\frac{\tilde{v}^2}{2}\right) \int {}_\partial\lfloor f_i \rfloor \tilde{v} \Big|_{x=L} \, d\tilde{v} \tag{8.10a}$$

$$\begin{aligned} \frac{\partial S_i}{\partial \theta_\tau} = {} & \frac{\mu\theta_\tau|\tilde{v}|}{2L_s} \exp\left(-\frac{\mu\theta_\tau\tilde{v}^2}{2}\right) \int {}_\partial\lfloor f_i \rfloor \tilde{v} \Big|_{x=L} \, d\tilde{v} \\ & + \left[\frac{\mu|\tilde{v}|}{2L_s} - \frac{\mu^2\theta_\tau|\tilde{v}|^3}{4L_s}\right] \exp\left(-\frac{\mu\theta_\tau\tilde{v}^2}{2}\right) \int f_i \tilde{v} \Big|_{x=L} \, d\tilde{v}, \end{aligned} \tag{8.10b}$$

with $\tilde{v} = v/v_{T,e}$, $\mu = m_i/m_e$ and $\theta_\tau = T_e/T_i$. The heat bath condition at $x = -L$ provides a sensitivity boundary condition,

$${}_\partial\lfloor f_e \rfloor\big|_{x=0,v>0} = \exp\left(-\frac{\tilde{v}^2}{2}\right) \int_{v<0} {}_\partial\lfloor f_e \rfloor\big|_{x=0} \tilde{v} \, d\tilde{v}, \tag{8.11a}$$

$$\begin{aligned} {}_\partial\lfloor f_i \rfloor\big|_{x=0,v>0} = {} & \mu\theta_\tau \exp\left(-\frac{\mu\theta_\tau\tilde{v}^2}{2}\right) \int_{v<0} {}_\partial\lfloor f_i \rfloor\big|_{x=0} \tilde{v} \, d\tilde{v} \\ & + \left[\mu - \frac{\mu^2\theta_\tau\tilde{v}^2}{2}\right] \exp\left(-\frac{\mu\theta_\tau\tilde{v}^2}{2}\right) \int_{v<0} f_i\big|_{x=0} \tilde{v} \, d\tilde{v}. \end{aligned} \tag{8.11b}$$

Compared to the boundary flux (8.4) for PIC, numerical implementation of (8.11b) is complicated by the flux source term, which cannot be represented by statistical distribution of particles. Following the same procedure as for (3.14) in Section 3.2.3, we introduce a numerical population of N_{influx} particles entering from $x = -L$ over a time step,

$$\hat{n}_{\text{influx}}\big|_{x=-L} = \sum_{s=1}^{N_{\text{influx}}} S_x(x - \hat{x}_s) S_v(v - \hat{v}_s). \tag{8.12}$$

Particle position and velocity $(\hat{x}_s, \hat{v}_s) \in [-L, -L + \hat{v}_s\Delta t] \times [0, \infty)$ are sampled from a probabilistic density function $p(x, v)$, which satisfies a condition similar to (3.29),

$$p(x,v) \neq 0 \qquad \forall (x,v) \in \left\{ (x,v) \;\middle|\; \int_{t_k}^{t_{k+1}} {}_\partial\lfloor f_i \rfloor\big|_{x=0,v>0} v \, dt \neq 0 \right\}, \tag{8.13}$$

so that $\hat{n}_{\text{influx}} \approx N_{\text{influx}}\, p(x, v)$ ensures sufficient particles in the necessary region to support it. The influx of ${}_\partial\lfloor f_i \rfloor$ during a time step is then represented with the influx particles,

$$\int_{t_k}^{t_{k+1}} {}_\partial\lfloor f_i \rfloor\big|_{x=0,v>0} v \, dt \approx \sum_{s=1}^{N_{\text{influx}}} \hat{W}'_s S_x(x - \hat{x}_s) S_v(v - \hat{v}_s), \tag{8.14}$$

with the weights

$$\hat{W}_s' = \frac{1}{N_{\text{influx}}\, p(\hat{x}_s, \hat{v}_s)} \left[\int_{t_k}^{t_{k+1}} {}_{\partial} \lfloor f_i \rfloor \big|_{x=0, v>0} v \; dt \right]_{\hat{x}_s, \hat{v}_s}. \tag{8.15}$$

This way any generic flux (even negative) can be represented with statistical distribution of particles, so long as $\hat{n}_{\text{influx}}$ satisfies (8.13). For demonstration we choose the uniform $p(x, v)$,

$$p(x, v) = \frac{2}{(5v_{T,i})^2 \Delta t} = \frac{2\mu\theta_\tau}{25\Delta t}, \tag{8.16}$$

for $-L \le x \le -L + v\Delta t$ and $0 \le v \le 5v_{T,i}$. The weight of an influx sensitivity ion is then

$$\begin{aligned} \hat{W}_s' = \frac{2\mu\tau}{25\Delta t N_{\text{influx}}} \Bigg[& \mu\theta_\tau \exp\left(-\frac{\mu\theta_\tau \tilde{v}_s^2}{2}\right) F_1 \\ & + \left(\mu - \frac{\mu^2\theta_\tau \tilde{v}_s^2}{2}\right) \exp\left(-\frac{\mu\theta_\tau \tilde{v}_s^2}{2}\right) F_2 \Bigg], \end{aligned} \tag{8.17}$$

with F fluxes represented by M_{outflux} sensitivity ions leaving the domain at $x = -L$ and N_{outflux} ions from PIC simuation,

$$F_1 = \int_{t_k}^{t_{k+1}} \int_{v<0} {}_{\partial} \lfloor f_i \rfloor \big|_{x=0} \tilde{v} \; d\tilde{v} dt \approx \sum_{s=1}^{M_{\text{outflux}}} \hat{W}_s \tag{8.18a}$$

$$F_2 = \int_{t_k}^{t_{k+1}} \int_{v<0} f_i \big|_{x=0} \tilde{v} \; d\tilde{v} dt \approx \sum_{p=1}^{N_{\text{outflux}}} W_p. \tag{8.18b}$$

8.3 Sheath results

The non-collocated scheme introduced in Section 3.2 is demonstrated first. Based on Figure 8.2, uniform mesh spacing is tailored to the different particles, with $\Delta v_e = 0.078\, v_{T,e}$ and $\Delta v_i = 0.078\, v_{T,i} = 0.1\Delta v_e$. Compared to the baseline potential, Figure 8.4 (a) shows that the potential sensitivity drops significantly at the source boundary $x = -L + L_s$, rather than in the sheath region, which indicates that changing θ_τ mainly shifts the ion Bohm velocity [53]. This is consistent with the sensitivity distributions in Figure 8.4 (b) and (c), where the electron sensitivity indicates variation in Boltzmann distribution due to the potential, while the strong sensitivity to the ions near the Bohm velocity indicates a shift of the ion Bohm velocity. The time history of the integrand $J(t) = \phi(0) - \phi(L; t)$ from (8.7) and its sensitivity in Figure 8.4 (d) indicates that the sensitivity fluctuates as in the PIC simulations. The sensitivity-pdf method also compares well with the brute-force computed $\mathcal{J}$ curve in Figure 8.4 (e), despite the irregular parametric dependence.

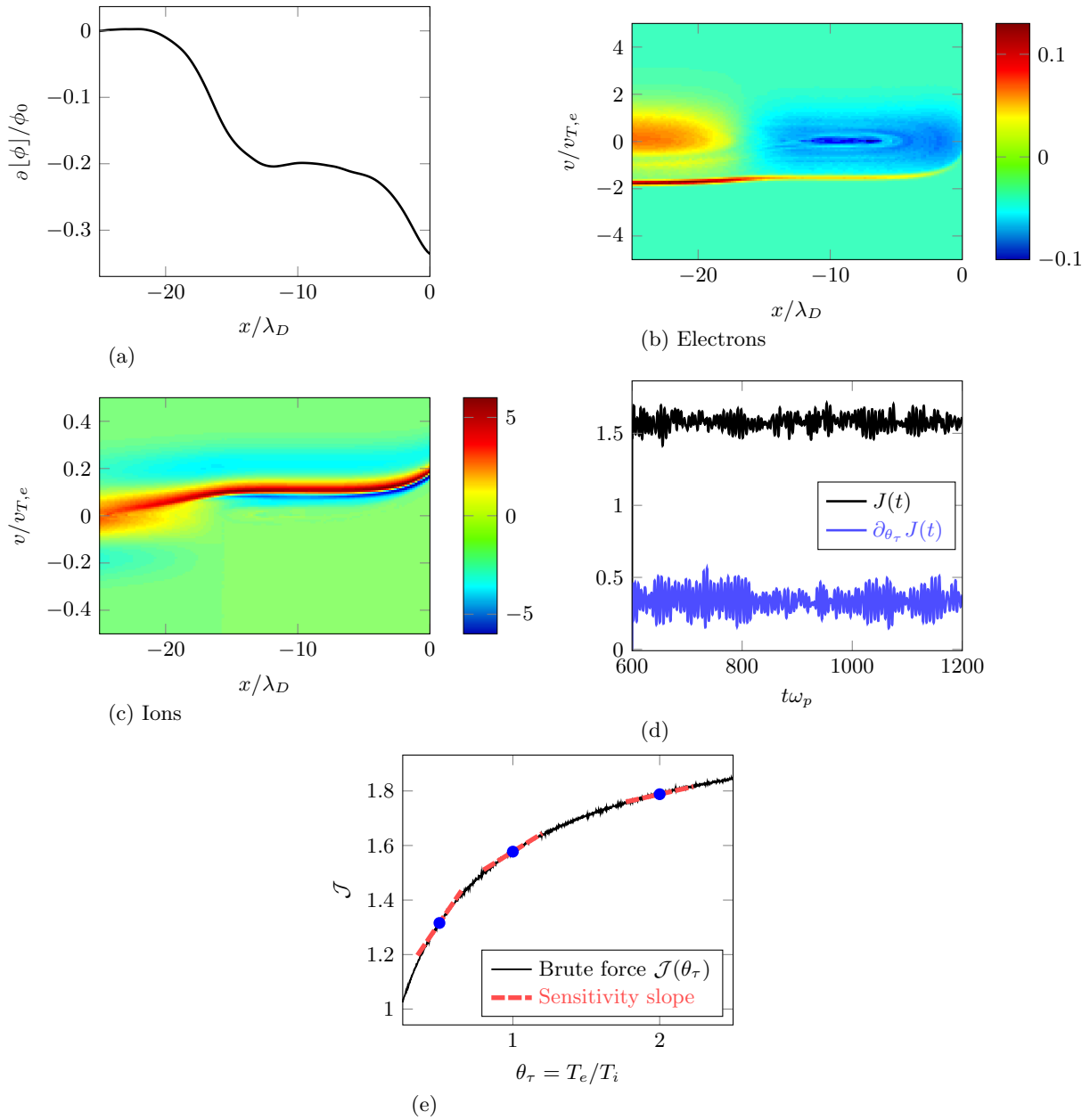


Figure 8.4: Sensitivity-pdf computation of the sheath edge with $\theta_\tau = 1.0$: (a) the electrostatic potential sensitivity; mean sensitivity distributions from t_i to t_f of (b) electrons and (c) ions; (d) the instantaneous QoI (8.7) and its sensitivity (8.8); and (e) the sensitivities (8.8) visualized with line segments for $\theta_\tau = 0.5$, 1.0 and 2.0 compared to the exhaustive brute-force estimates.

8.4 Adaptive scheme

Particle-based methods can be particularly attractive when the convection velocity varies significantly more than the thermal velocity. In such a case, mesh-based discretizations require dense velocity meshes to both resolve velocity distributions and span a wide velocity domain. The CFL constraint (5.4) can accentuate consequences of this. The particle-pdf method is relatively insensitive to this and can be further enhanced to be adaptive.

To illustrate this, the ion source term in (8.3) is augmented with a convection velocity v_c,

$$S_i = \frac{1}{L_s}\frac{1}{\sqrt{2\pi}v_{T,i}}\exp\left(-\frac{(v-v_c)^2}{2v_{T,i}^2}\right)\int f_i\big|_{x=0} v\, dv, \tag{8.19}$$

where v_c oscillates

$$v_c = \begin{cases} \frac{v_0}{2}\big(1-\cos[\omega_c(t-t_1)]\big) & t \geq t_1 \\ 0 & \text{otherwise,} \end{cases} \tag{8.20}$$

with $t_1 = 1230/\omega_{p,e}$ and $\omega_c = 120\omega_{p,e}$. The parameter-of-interest is the amplitude v_0. We compute the sensitivity of same QoI (8.7) though with shorter time period,

$$\mathcal{J} = -\frac{1}{t_3-t_2}\int_{t_2}^{t_3}\left[\phi(0)-\phi(L;t)\right]\, dt, \tag{8.21}$$

where $t_2 = 1336/\omega_{p,e}$ and $t_3 = 1360/\omega_{p,e}$.

The non-collocated method used for the steady case and particle addition method are adaptively combined, so that sensitivity particles mostly sample important regions. Algorithm 1 shows the simple heuristics used to do this while satisfying the constraint (3.29).

The sensitivity is computed with $\mathcal{H}_{\min} = 10^{-10}n_0\omega_{p,e}v_{T,e}^{-1}$ and $\hat{n}_{\min} = 1.0\Delta x^{-1}\Delta v^{-1}$. Figure 8.5 (a) and (b) show the ion sensitivity particles distributed and supporting the necessary region, satisfying (3.29). The velocity mesh in this simulation is as dense as in Section 8.3, but spans about 8 times larger velocity space domain. Yet, as shown in Figure 8.5 (c), only about twice the adaptive scheme sensitivity particles are needed as for PIC simulation. Figure 8.6 (a) shows the time history of the QoI integrand (8.21) and its sensitivity, for one period of convection velocity oscillation ω_c. The sensitivities computed by this adaptive scheme, as shown in Figure 8.6 (b), provide a good estimate of variation in $\mathcal{J}$.

Algorithm 1 Adaptive scheme

▷ the support constraint (3.29)

Require: $\hat{n}_{\mathbf{i},\mathbf{i}_v} \neq 0 \qquad \forall(\mathbf{i},\mathbf{i}_v) \in \left\{(\mathbf{i},\mathbf{i}_v)\middle||\mathcal{H}_{\mathbf{i},\mathbf{i}_v}| \neq 0\right\}$

Ensure: $\hat{n}_{\mathbf{i},\mathbf{i}_v} \geq \hat{n}_{\min} \qquad \forall(\mathbf{i},\mathbf{i}_v) \in \left\{(\mathbf{i},\mathbf{i}_v)\middle||\mathcal{H}_{\mathbf{i},\mathbf{i}_v}| \geq \mathcal{H}_{\min}\right\}$

Compute $\hat{n}_{\mathbf{i},\mathbf{i}_v}$ by (3.39a)
for every $s \in [1, M]$ **do**
  for every $(\mathbf{i},\mathbf{i}_v) \in [1, l+1]^{2d}$ **do**
    while $|\mathcal{H}_{\mathbf{i},\mathbf{i}_v}| \geq \mathcal{H}_{\min}$ **and** $\hat{n}_{\mathbf{i},\mathbf{i}_v} \leq \hat{n}_{\min}$ **do**
      Add particle by (3.31): $(\hat{\mathbf{x}}_s, \hat{\mathbf{v}}_s) \in \left[\mathbf{x}_\mathbf{i} - \frac{\Delta x}{2}, \mathbf{x}_\mathbf{i} + \frac{\Delta x}{2}\right] \times \left[\mathbf{v}_{\mathbf{i}_v} - \frac{\Delta v}{2}, \mathbf{v}_{\mathbf{i}_v} + \frac{\Delta v}{2}\right]$
      Update $\hat{n}_{\mathbf{i},\mathbf{i}_v}$ by (3.39a)
      $M \leftarrow M + N_{\text{new}}$
  Evaluate h_s by (3.39b)
Update weights by (3.40a)
if $M > M_{\max}$ **then**
  Redistribute particles by (3.41)

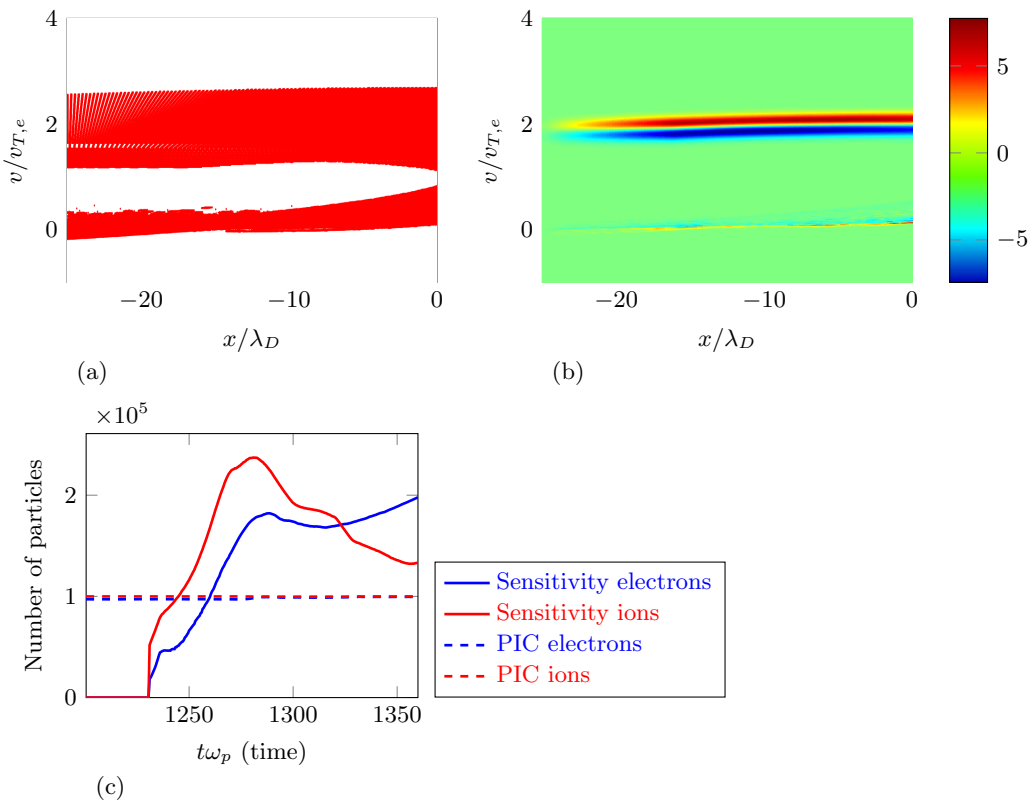


Figure 8.5: Adaptive sensitivity-pdf sensitivity for the harmonically forced sheath: ion sensitivity particle distribution (a) shows that they are clustered near sensitivity regions in (b), which shows the ion sensitivity distribution at $t = 1300\omega_{p,e}^{-1}$, with $v_0 = 2.0v_{T,e}$. (c) The adaptively adjusted number of sensitivity particles is comparable to the PIC simulation.

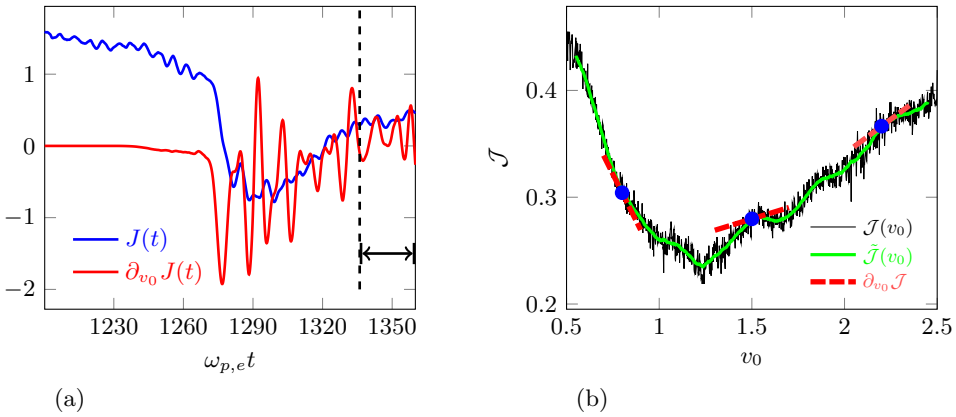


Figure 8.6: (a) The QoI (8.21) and its sensitivity to v_0 for $v_0 = 2.2v_{T,e}$, where the interval after dashed line represents the time between t_2 and t_3 in (8.21). (b) Sensitivities at $v_0 = 0.8v_{T,e}, 1.5v_{T,e}$ and $2.2v_{T,e}$ match the estimates (green), which is a $\Delta v_0 = 0.1$ moving window averaged from the brute-force QoI (black).

Chapter 9

Additional discussion and summary

The sensitivity-pdf approach we introduce is shown to meet the main challenge presented by the chaotic dynamics of the particles. It consistently and efficiently provides sensitivities for QoIs despite chaotic particle dynamics. Rather than invoking the ergodicity assumption, which can be limiting, this method computes sensitivities by discretizing the differentiated Vlasov–Poisson equation with particles. This avoids the consequences of non-commutability between the derivative and the continuum limit, at comparable computational cost to the original PIC method.

The key assumption of the proposed sensitivity-pdf approach is the property of the $N \to \infty$ continuum limit described in (6.22), where a large number of interacting particles under chaotic dynamics macroscopically constitutes a state that is Lyapunov stable. Although this seems true for many cases [15–17], its specific criterion remains unknown. When this assumption is not applicable, computing sensitivities is possible only under the ergodicity assumption. Nonetheless, this continuum limit assumption grants us a valuable opportunity to compute sensitivity of transient responses through particle methods, without requiring ergodicity.

The non-commutability introduced in Section 2.3 is analogous to the concept of dual-consistency for adjoint-based methods [19]: for dual-consistent methods, the adjoint of the discretization is not only discrete-exact but also a consistent discretization of the continuous adjoint. However, the particle-pdf method provides a sensitivity consistent with the continuum limit, but it is not particle-exact: it suffers from similar statistical limitations and possible sensitivity to mesh sizes and time steps as the PIC scheme on which it is based. Whether or not a dual-consistent discretization exist is unclear, since it seems that computational particles should not actually imitate the actual (chaotic) dynamics of physical particles. Wang et al. [29] show that periodic redistribution of particles can suppress errors coming from diverging particle trajectories, though it is unclear that the redistribution would obviate the non-commutability challenge, especially upon considering the effect of interpolation shown in Figure 6.2 (d).

In this paper, the accuracy of particle-pdf sensitivity is assessed by comparison with a QoI curve constructed by brute force. This is emblematic of a well-known difficulty for sensitivity of chaotic dynamical systems [10, 13, 54], since the exact sensitivity value is unavailable. Taking more particles or longer time-

averaging may decrease the variation, but it remains effectively non-differentiable, which frustrates error analysis. Finite-volume solutions provide an estimate of the continuum limit sensitivity, up to the accuracy of the finite-volume discretization.

For cases with many parameters-of-interest, joining these methods with adjoint-based methods would be valuable. Since the adjoint equation is similar to its original governing equation, the particle discretization of the adjoint is expected to be similarly effective. Péraud et al. [55] report a similar case, where an adjoint-based deviational Monte Carlo method is formulated for the linearized Boltzmann equation. Also, extension to PIC methods augmented with short-range interaction or stochastic collisional components, would appear to be straightforward, though this too would need further investigation.

Part II

A Gradient-based Optimization Framework for Chaotic Turbulent Flows

Chapter 10

Introduction

10.1 Optimization in flow computations

Recent advances in numerical simulation methods and computer resources have enabled predictions of fluid dynamics in realistic and complex flow conditions and geometries. Predictions, however, do not themselves lead to better scientific understanding or engineering design. They often require tools to harness the full space–time data they provide, in order to augment their utility. One such tools is optimization, which seeks to find the optimal characteristics of a system for a specific goal. Mathematically, it refers to finding the optimal variables $\mathbf{\Theta}$ that minimize (or maximize) a quantity $\mathcal{J}$ that represents the goal, which is the quantity-of-interest (QoI) or objective functional. We refer to minimization of $\mathcal{J}$ throughout this study; maximization problem can be easily accommodated as minimization of $-\mathcal{J}$.

We consider optimization problems in flow computations that have two main characteristics. First, they involve large numbers of optimization variables, generally exceeding hundreds. Optimization in such a high-dimension can benefit from the gradient of the $\mathcal{J}$ to find a local optimum in its neighborhood [56]. We do not pursue the global optimization, which would be hard to assert. Second, we consider flow dynamics governed by known governing equations without any model closures, though it does seem that the methods introduced could be extended to such cases. This naturally leads us to use techniques for equality-constrained optimization [57–59]. Adjoint methods provide powerful tools for the gradient-based, equality-constrained optimization. For the governing equations of fluid dynamics, adjoint solutions provide the gradient in high-dimensional space with a similar computational cost of baseline predictive simulation.

Adjoint gradient-based optimization has been used for various applications, within which we envision two broad objectives. First, the optimization can accelerate engineering designs by avoiding extensive parametric investigation. Both passive and active flow controls can be designed [60, 61], such as airfoil geometry [56, 62], jet nozzle shape [63], and acoustic dampers [64]. Such applications also include error estimation and control of the predictive simulations themselves, ranging from grid adaptation [65, 66] to data-assimilated modeling [67–71]. The other application is investigation of poorly understood flow mechanisms. For example,

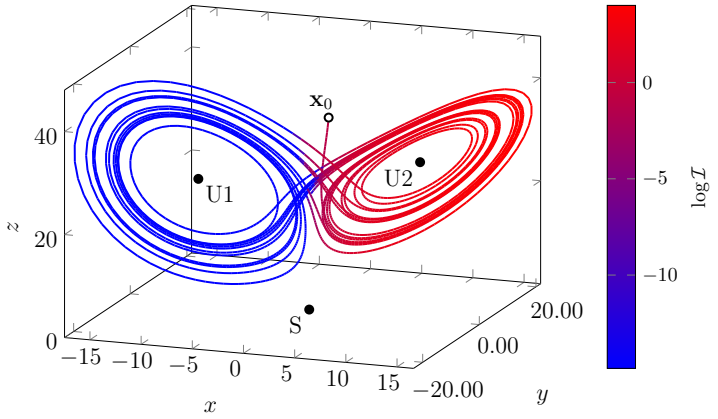


Figure 10.1: Baseline trajectory of the Lorenz equation starting from the initial condition $\mathbf{x}_0$ (black circle). Black filled dots indicate fixed points of the system. Color shows the magnitude of the instantaneous objective functional for optimal control, which favors the rotation around $U1$ over $U2$. The details of the optimal control problem will be introduced in Chapter 14.

various flow instabilities beyond the linear regime can be revealed and analyzed by optimization, such as thermoacoustic instability [72], bypass transition to turbulence [73–75], and Rayleigh–Taylor instability [76]. Another example is sound generation mechanisms and control [77]. Even when the mechanism is not well understood and its controllability is obscure, the optimization can guide a pathway to control [78, 79].

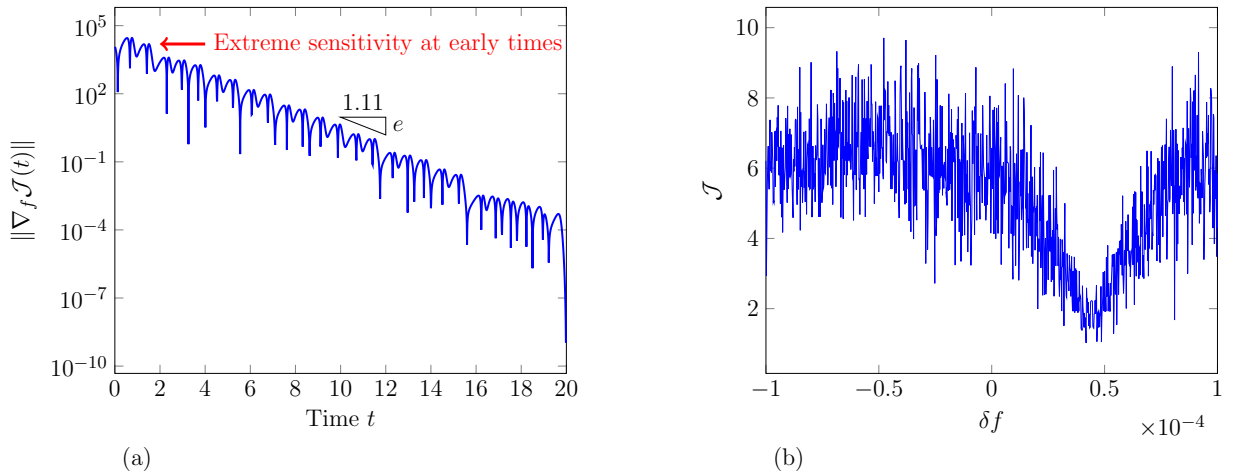
10.2 Challenge of optimization for turbulent flows

Optimizing turbulent flows can be challenging in various aspects. We focus particularly on challenges due to their chaotic dynamics.

The impact of chaotic dynamics on optimization can be illustrated with the simple Lorenz equation [80],

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z + f(t), \end{aligned} \tag{10.1}$$

whose state (x, y, z) orbits around two unstable fixed points, as visualized in Figure 10.1. A $\mathcal{J}$ can be defined to favor the neighborhood of one of the fixed points, and a time-varying controller $f(t)$ is sought to control the state to minimize $\mathcal{J}$. The full details of this optimal control problem are introduced in Chapter 12, where a standard gradient-based optimization is developed to optimize $f(t)$ to minimize $\mathcal{J}$. This is done using the gradient $\nabla_f \mathcal{J}$ to inform search directions. However, chaos hinders this approach in two ways. First, the gradient $\nabla_f \mathcal{J}$ exhibits extreme sensitivity to $f(t \approx 0)$, which is manifest as the exponential growth of



(a) (b)

Figure 10.2: The impact of chaos on the optimal control problem of Lorenz system: (a) the gradient of $\mathcal{J}$ to $f(t)$ growing exponentially backward in time, and (b) highly non-convex $\mathcal{J}[f]$ along the direction of gradient $\nabla_f \mathcal{J}(t)$. For (b), these data are generated by brute-force evaluation of $\mathcal{J}$ for 10^4 values of δf, with $f(t) = \delta f \nabla_f \mathcal{J}(t)$ (see Chapter 12).

the gradient in reverse time seen in Figure 10.2 (a). Such amplification causes the entire optimization to be biased toward control to $f(t \approx 0)$. In addition to this challenge, $\mathcal{J}$ becomes highly non-convex with irregular variations in the optimization space, as shown in Figure 10.2 (b). These create a large number of local minima, each with tiny neighborhoods, many of which do not reduce $\mathcal{J}$ significantly [61]. A standard gradient-based optimization thus stalls, as shown in Figure 10.3. These impacts of chaos on optimization will be further demonstrated for chaotic advective systems in Chapter 13.

These two characteristics of chaos defy the underlying strategy of gradient-based optimization that seeks a nearby local optimum. Even if an optimization problem has an effective optimum, it is nearly impossible to avoid pitfalls of many poor local optima and find an effective one. In this regard, the challenge of chaos is confounding: when a $\mathcal{J}$ cannot be optimized significantly, is it because $\mathcal{J}$ is inherently bounded, or simply that we did not yet find a good optimum?

This challenge is recognized in standard gradient-based optimizations for unsteady turbulent flows. The divergence of the adjoint sensitivity field—the so-called butterfly effect—is observed no matter what $\mathcal{J}$ is defined [76, 81]. The error amplification due to this effect is often thought to degrade optimization performance [76, 79, 82]. Likewise, non-convexity of $\mathcal{J}$ has also been invoked in regard to turbulent flows [61, 83]. Because of non-convexity, the optimization is either limited in its time horizon or attempts can be made with multiple initial guesses to obtain a better local optimum [76, 83]. Kim *et al.* [79] attempted to reduce the far-field sound of a turbulent jet using an adjoint method. The optimized control ended up being concentrated in the early period of the time horizon. Also, it was not clear whether or not the optimization

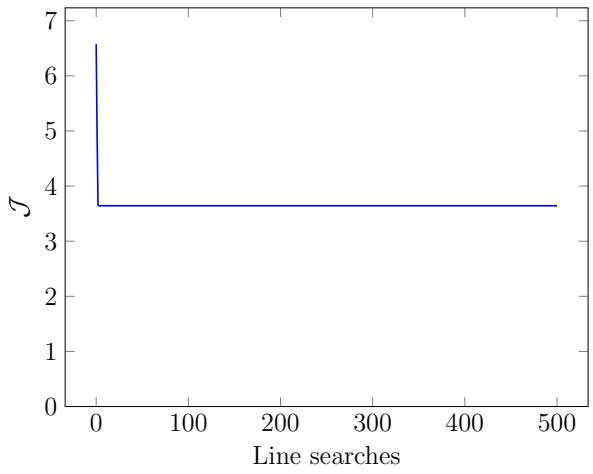


Figure 10.3: Optimization result from the standard gradient-based method.

was limited by an inherent lack of controllability or the chaotic dynamics of turbulent flows. Pérez *et al.* [84] attempted to optimize a control for a three-dimensional compressible backward-facing step flow, however an inefficient time-dependent control is obtained due to the diverging adjoint-field associated with chaos.

Nonetheless, there are few, if any, alternatives to the gradient-based optimization methods. There exist many non-convex optimization techniques [85], such as Lipschitz optimization [86, 87], statistical-model-based global optimization [88, 89], and the random search method [90, 91]. However, the non-convexity caused by the chaotic dynamics is often too challenging [86, 87]. Also, they either utilize some assumed statistical models for $\mathcal{J}$ [88, 89], or resort to extensive parametric search [90, 91], which is prohibitive for costly simulations. The many techniques for non-convex optimization are not easily applied to turbulent flows, which are both high dimensional and significantly non-convex and, in addition, expensive to compute.

10.3 Hope for optimization of turbulent flows

While Figure 10.2 makes the optimal control of the Lorenz equation seem impossible, this system is controllable. Figure 10.4 shows such a controlled trajectory. This implies that it is only the chaos of the system that obscures gradient-based methods from finding such an effective optimum. Furthermore, even with its susceptibility to small errors, this optimal solution can still be used to craft a robust closed-loop feedback control law, and thereby provide valuable data that would be inaccessible without optimization. The details of finding this solution and further investigation of its usefulness is in Chapter 14 to provide context for the corresponding turbulence problem.

There is a reason to believe that there can be similar utility for turbulent flows as well. Our starting-point conjecture supporting this is that some turbulent flows have a relatively chaotic part that obscures gradient-

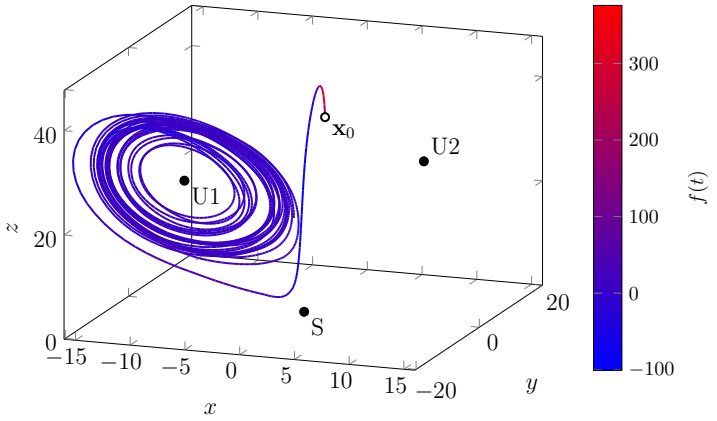


Figure 10.4: A controlled trajectory of the Lorenz equation. Color shows the magnitude of the instantaneous control forcing. It will be explained in Chapter 14 how this control is designed.

based methods from finding the part that is less chaotic but crucial for optimization. It might be expected that this chaotic part is associated with small-scale turbulent fluctuations, while the less chaotic, crucial part is associated with relatively large-scale coherent structures of the flow. The same idea underlies efforts to average or filter turbulence fluctuations to extract a reduced model for turbulent flows. In Chapter 13, we introduce a one-dimensional model based on the Kuramoto–Sivashinsky (K–S) equation to show this property and how it might also be at play in turbulent flows. For such cases, if understanding and methods permit, it might be possible to extract the dynamics of large-scale flow structures, or apply averaging/filtering techniques to avoid the chaotic part that obscures the optimization. Unfortunately, in many cases, this has not been possible.

It is unlikely that there will be a proof of this conjecture in the case of turbulence. However, there is evidence supporting it. One example is the linear response theory for dissipative chaos [8]. It proved that ensemble/time-averages based on ergodic state distributions are differentiable, which supports the possibility of optimization for some chaotic dynamical systems. $\mathcal{J}$ with these averages are not necessarily highly non-convex, so gradient-based methods will remain effective as long as their gradients can be computed. Based on this theory, many algorithms have been proposed for computing gradients of ergodic quantities: ensemble-averaging of adjoint sensitivities [6, 11], using the fluctuation–dissipation theorem [12], a pdf-based approach [9, 92], least-square shadowing [13, 93, 94], cumulant-truncation [54], and space-splitting [95]. However, most of them are not yet applicable for optimization: they can suffer from poor convergence due to diverging sensitivity [6, 11]; they require prohibitive computational costs, comparable to the original optimization problem itself [9, 13, 92–94]; and they may not be accurate when the system differs significantly from the formal assumption of being ergodic or uniformly hyperbolic [12, 54, 92, 94]. Chandramoorthy and Wang [95] recently proposed a promising gradient computation method that requires computation of all covariant

Lyapunov vectors, however this too is not yet feasible for large-scale turbulent flows. In Part I of the dissertation, we successfully compute the gradient of statistical distribution from chaotic particle dynamics in continuum limit [96], however the method relies on the particular properties of the particle dynamics, so direct extension to turbulent flows is not possible.

Support also comes from experience with particular flow systems. One example is a series of attempts to understand and control turbulent jet noise. While the interplay between the jet turbulence and radiated sound is intricate, the success of coherent wave-packet structures to describe some characteristics suggests that these orderly large-scale structures may be controllable [77, 97, 98]. Low-frequency, low-wavenumber eigenmodes from stability analysis show reasonable agreement with time-averaged statistics [99, 100]. Also, successful passive control of low-frequency, aft-angle jet noise with nozzle modification further supports controllability of these large-scale flow structures [101–105]. However, extracting the real-time dynamics of such coherent structures into a reduced-order model remains challenging. The underlying nonlinearity of the turbulence seems to manifest as intermittency of the large-scale coherent structure, making a complete separation of their dynamics difficult [98]. This necessitates a large number of wave-packet modes to resolve data [106], or the resulting reduced-order model can be fragile, or even unstable. Many ansatz are suggested for useful bases to extract these structures [98, 107], however none of them can rival the predictive capability of a well-resolved simulation of the full dynamics.

The last example we present, though there are many more, is passive control and optimization for thermoacoustic oscillation of rocket engines and gas turbines [64, 72, 108, 109]. Although their operation involves intense turbulence [64, 109], sensitivity analyses and optimization focus on the key mechanism of the thermoacoustic instability [64, 72, 108]. From this viewpoint, turbulence only induces fluctuations around the mean flow, which provides the seed that excites thermoacoustic instability. However, to obtain further prognostic information to prevent the instability, analysis of fluctuations in turbulent combustors is required, which leads to the study of their chaotic behavior [109]. Intermittency and multi-dimensional chaos in turbulent combustion dynamics has been extensively studied and measured [110–113], but a design optimization harnessing this understanding in the form of a predictive model remains a goal.

10.4 An overview of this study

We develop an optimization framework for chaotic turbulent flows. This might be of questionable utility, but various examples introduced in Section 10.3 implicitly support the conjecture that there may be a less-chaotic part of flow that is sufficiently independent from more strongly chaotic turbulent fluctuations. In

such cases it is natural to modify (or simplify) the description of the flow dynamics, either by averaging or filtering out unnecessary chaotic dynamics or by extracting only the useful part for the optimization. However, a complete separation between these components is not always possible or clearly known, and inability to do so renders the modified dynamics inaccurate. Hence, if possible, a regularization procedure that circumvents non-convex features at the optimization level is attractive. An advantage is that such an approach leaves the exact governing equation intact.

A subsidiary goal of this study is to quantify the impact of chaos on the optimization, such as illustrated for the Lorenz equation in Section 10.2. The error amplification in the gradient computation has been believed to be the main degrading factor for optimization [76, 79, 81, 82], for which the Lyapunov exponent is a well-established metric [114, 115]. However, the gradient error evaluated in Part I of the dissertation suggests that the error amplification may not be the main hindrance. Rather, the range for which the gradient predicts $\mathcal{J}$ decays exponentially in time [96]. While it is related to the non-convexity of $\mathcal{J}$, this aspect is less recognized [61, 83]. There are some quantities which are related to the non-convexity of $\mathcal{J}$, such as entropy and fractal dimension [116–120], however they do not directly inform how much and how fast an objective functional becomes non-convex.

Therefore, to support the primary goal, we develop an indicator for non-convexity to assess the challenge. For this goal, it would be best to define an invariant quantity of the dynamical system, such as the previously mentioned Lyapunov exponent, entropy and fractal dimension, which are invariant under coordinate transformation and thus the representative properties of the dynamical system. This by itself requires an in-depth theoretical investigation, which is beyond the scope of this study and likely not possible for turbulence. Instead, we suggest a practical measure of the decay rate (or time scale) of the linear range presented in Part I of the dissertation. While in Part I it is evaluated for a specific $\mathcal{J}$ and parameter, we generalize it to be applicable for generic $\mathcal{J}$ and optimization parameters.

The method overall is an extension of the standard adjoint gradient-based method for equality-constrained optimization formulated in Chapter 11 for governing equation for a generic dynamical state. The governing compressible flow equation is also introduced for the specific application. These are done in the context of the subsequent development.

Chapter 12 uses the Lorenz system to quantify the multiple ways that chaos impedes gradient-based optimization. This is connected with horseshoe mapping. Quantitative indicators are used to illustrate opportunities that can potentially be exploited in the less analytically tractable case of turbulence.

Specific examples are introduced in Chapter 13. First, standard adjoint-based control is demonstrated on two simple chaotic advective systems, for which the impact of chaos on optimization is clear: a one-

dimensional Kuramoto–Sivashinsky (K–S) Equation [121, 122] and a two-dimensional Kolmogorov flow [123–125]. This is further illustrated and confirmed for turbulent flows. A one-dimensional model problem, based on advection plus K–S equation (Adv+KS), is developed to reflect the multiple scales of turbulence, relatively deterministic large scales and relatively chaotic small scales. The same behavior is confirmed for a three-dimensional turbulent Kolmogorov flow.

In Chapter 14, a penalty-based optimization framework is proposed to circumvent non-convex feature. It is demonstrated with the Lorenz equation example, and the usefulness of the (unstable) optimum solution is further discussed. Then, in Chapter 15, the framework is demonstrated effective on the model systems introduced in Chapter 13. It is further demonstrated and confirmed that its effectiveness can extend to turbulent flows.

Chapter 11

Optimal control formulation

In this chapter we formulate the constrained optimization problem from a bird's-eye view, starting from its origin, and including the adjoint formulation that will provide gradients. Analyzing this with model problems and applications to turbulent flows in Chapters 12 and 13 leads to the formulation of the algorithm developed in Chapter 14.

11.1 Equality-constrained optimization

We focus on systems whose physics are governed by a few deterministic principles, usually conservation laws. As such, we denote the governing equation as

$$\mathcal{N}[\mathbf{q};\boldsymbol{\Theta}] = \mathbf{0}, \qquad \mathcal{N} \in \mathbb{N}, \tag{11.1}$$

where $\mathbf{q} \in \mathbb{Q}$ is the state of the system and $\boldsymbol{\Theta} \in \mathbb{T}$ is a control input. In our examples, $\mathcal{N}$ will be the Lorenz equation in Chapter 12, the Kuramoto–Sivashinsky equation in Chapter 13, and the compressible-fluid flow equation, which will be introduced in Section 11.4 and used for demonstration in Chapter 13. We use $\mathbb{N}$, $\mathbb{Q}$, and $\mathbb{T}$ to denote the spaces on which (11.1), $\mathbf{q}$, and $\boldsymbol{\Theta}$ are respectively defined. For example, if (11.1) is the ideal gas law,

$$p = \rho R T, \tag{11.2}$$

then $\mathbb{N} = \mathbb{R}^+$ (positive reals) with $\mathbf{q}$ a vector $(\rho, p, T)^T \in \mathbb{Q} = (\mathbb{R}^+)^3$ composed of density, pressure and temperature. In general $\mathbb{N} \equiv \mathbb{Q}$, so that the state $\mathbf{q}$ is determined completely by the governing equation $\mathcal{N}$. The control space $\mathbb{T}$ in general can be defined independently of $\mathbb{Q}$. Here we state that $\mathbb{Q}$ and $\mathbb{T}$ are Hilbert spaces and defer further mathematical definitions.

Optimization seeks to minimize a scalar objective functional $\mathcal{J} : \mathbb{Q} \times \mathbb{T} \to \mathbb{R}$:

$$\text{minimize } \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] \in \mathbb{R} \qquad \text{such that} \quad \mathcal{N}[\mathbf{q};\boldsymbol{\Theta}] = \mathbf{0}. \tag{11.3}$$

The objective functional can be defined to reflect many aspects of interest, typically progress of the control toward a target state and expense of control input. Thus, (11.3) is a generic statement of the objective we pursue.

Finding the global minimum over all feasible pairs of $(\mathbf{q}, \boldsymbol{\Theta})$ is difficult at best and typically infeasible, though this is unnecessary since substantially improving $\mathcal{J}$ is often useful enough. Hence, we are satisfied with a $(\mathbf{q}, \boldsymbol{\Theta})$ approaching a local minimum in a reasonable size of neighborhood [57, Chap 2.1].

Definition 11.1 (local minimum)**.** *Let $\mathbb{Q}$ and $\mathbb{T}$ be Hilbert spaces, and let a functional be $\mathcal{J} : \mathbb{Q} \times \mathbb{T} \to \mathbb{R}$. Suppose a non-empty subset $\mathrm{dom}(\mathcal{N}) \equiv \{(\mathbf{q}, \boldsymbol{\Theta}) | \mathcal{N}[\mathbf{q}, \boldsymbol{\Theta}] = 0, \mathbf{q} \in \mathbb{Q}, \boldsymbol{\Theta} \in \mathbb{T}\}$ exists. Then $(\mathbf{q}^*, \boldsymbol{\Theta}^*) \in \mathrm{dom}(\mathcal{N})$ is a local minimizer of the functional $\mathcal{J}$ if $\exists \epsilon > 0$ such that*

$$\mathcal{J}[\mathbf{q}^*, \boldsymbol{\Theta}^*] \leq \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] \qquad \forall (\mathbf{q}, \boldsymbol{\Theta}) \in \mathrm{dom}(\mathcal{N}) \quad s.t. \quad \|\boldsymbol{\Theta} - \boldsymbol{\Theta}^*\|_{\mathbb{T}} < \epsilon, \tag{11.4}$$

where $\| \cdot \|_{\mathbb{T}}$ is the norm defined on $\mathbb{T}$. $\mathcal{J}[\mathbf{q}^*, \boldsymbol{\Theta}^*]$ is called a local minimum, and ϵ is called the size of neighborhood of the local minimum.

For identifying a local minimum, convexity is an important property [126].

Definition 11.2 (convexity)**.** *A subset C of a Hilbert space is convex if for $\forall \mathbf{x}, \mathbf{y} \in C$ and $\forall \alpha \in [0, 1]$,*

$$\alpha \mathbf{x} + (1 - \alpha)\mathbf{y} \in C. \tag{11.5}$$

Given a convex subset C, a functional $J : C \to \mathbb{R}$ is convex if for $\forall \mathbf{x}, \mathbf{y} \in C$ and $\forall \alpha \in [0, 1]$,

$$\mathcal{J}[\alpha \mathbf{x} + (1 - \alpha)\mathbf{y}] \leq \alpha \mathcal{J}[\mathbf{x}] + (1 - \alpha)\mathcal{J}[\mathbf{y}]. \tag{11.6}$$

If the equality does not hold for $\forall \mathbf{x} \neq \mathbf{y} \in C$ and $\forall \alpha \in [0, 1]$, $\mathcal{J}$ is strictly convex.

For a convex functional, Theorem 11.1 is useful to identify a local minimizer [57, 126].

Theorem 11.1 (global minimizer of a convex function)**.** *If a functional $\mathcal{J} : C \to \mathbb{R}$ is convex, then any local minimizer $\mathbf{x}^* \in C$ is a global minimizer of $\mathcal{J}$ in C. In addition, if $\mathcal{J}$ is differentiable, then any stationary point is a global minimizer of $\mathcal{J}$ in C.*

Proof. See Nocedal and Wright [57, Theorem 2.5]. □

For certain readily identifiable convex functionals, the search for a global solution reduces to the hunt for a local one [126]. While dom($\mathcal{N}$) in Definition 11.1 may not be convex by itself, it may include many convex subsets with a reasonable size. Then identifying a local minimizer in one of them is sufficient for our purpose.

Another important consequence of Theorem 11.1 is that a local minimum can be characterized as a stationary point, without investigating every point in a neighborhood. For equality-constrained optimization, an attractive strategy to identify a stationary point is based on Lagrange multipliers, as conceived by Lagrange and later generalized as the Karuch–Kuhn–Tucker (KKT) condition, which includes inequality constraints [57].

11.2 The method of Lagrange multipliers for a real vector space

The method of Lagrange multipliers is well-established [57–59]. Our purpose here is only to illustrate our use of the adjoint method, as it is typically understood. In order to avoid tedious mathematical definitions, in this section we limit ourselves to real vector spaces,

$$\mathbb{Q} = \mathbb{R}^m \qquad \mathbb{N} = \mathbb{R}^n \qquad \mathbb{T} = \mathbb{R}^c, \tag{11.7}$$

where $n = m$ and c are positive integers.

We define a Lagrangian function for the optimization problem (11.3),

$$\begin{aligned} \mathcal{L}[\mathbf{q}, \mathbf{q}^\dagger, \boldsymbol{\Theta}] &= \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] - \left\langle \mathbf{q}^\dagger, \mathcal{N}[\mathbf{q}; \boldsymbol{\Theta}] \right\rangle \\ &= \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] - \mathbf{q}^{\dagger T} \mathcal{N}[\mathbf{q}; \boldsymbol{\Theta}], \end{aligned} \tag{11.8}$$

where $\mathbf{q}^\dagger \in \mathbb{R}^n$ is the Lagrange multiplier vector, and $\langle \cdot, \cdot \rangle$ is the inner product associated with the Hilbert space on which (11.1) is defined, which in this section it is taken to be the inner product for $\mathbb{R}^n$. A local solution of (11.3) then satisfies the important condition of Theorem 11.2 related to this Lagrangian.

Theorem 11.2 (First-order optimality condition)**.** *Suppose that* $(\mathbf{q}^*, \boldsymbol{\Theta}^*)$ *is a local solution of (11.3) on the space defined in (11.7). We denote as* $\mathcal{N}_l$ *the* l*-th component of the vector* $\mathcal{N}$*, and suppose that the gradients* $\nabla_{\mathbf{q}} \mathcal{N}_l$ *with* $l = 1, \ldots, n$ *are linearly independent. Then there is a Lagrange multiplier vector* $\mathbf{q}^{\dagger *}$*, such that the following conditions are satisfied with* $(\mathbf{q}^*, \boldsymbol{\Theta}^*)$*,*

$$\nabla_{\mathbf{q}} \mathcal{L} \equiv \nabla_{\mathbf{q}} \mathcal{J} - \left(\nabla_{\mathbf{q}} \mathcal{N} \right)^T \mathbf{q}^\dagger = 0 \tag{11.9a}$$

$$\nabla_{\boldsymbol{\Theta}}\mathcal{L} \equiv \nabla_{\boldsymbol{\Theta}}\mathcal{J} - \left(\nabla_{\boldsymbol{\Theta}}\mathcal{N}\right)^T \mathbf{q}^\dagger = 0 \tag{11.9b}$$

$$\nabla_{\mathbf{q}^\dagger}\mathcal{L} \equiv \mathcal{N}[\mathbf{q};\boldsymbol{\Theta}] = 0. \tag{11.9c}$$

And $(\mathbf{q}^*, \mathbf{q}^{\dagger *}, \boldsymbol{\Theta})$ *corresponds to a stationary point of the Lagrangian* $\mathcal{L}[\mathbf{q}, \mathbf{q}^\dagger, \boldsymbol{\Theta}]$.

Proof. See Nocedal and Wright [57]. □

Many numerical optimization algorithms are founded explicitly or implicitly on Theorem 11.2, and pursue a solution that satisfies its condition. We mainly envision two kinds. The first kind of algorithms pursues a solution of the nonlinear equation (11.9) for $\mathbf{q}$, $\mathbf{q}^\dagger$, and $\boldsymbol{\Theta}$. For example, sequential quadratic programming (SQP) methods applies Newton's method to solve (11.9) iteratively [57, 127]. However, these methods require the Hessian ($\mathbb{R}^m \times \mathbb{R}^m$ matrix) of the governing equation. For turbulent flow simulations, m typically exceeds millions, so its computation is prohibitive in both memory and operation counts. The other kind of algorithms is the focus of this study, which is typically implemented as direct adjoint looping (DAL) [72, 73, 75–79]. This framework is centered on the gradient information, which the Lagrangian $\mathcal{L}$ provides, and utilizes it iteratively within gradient-based algorithms. We consider the first-order variation of $\mathcal{L}$,

$$\begin{aligned}
\delta\mathcal{L} &= \nabla_{\mathbf{q}^\dagger}\mathcal{L}^T\delta\mathbf{q}^\dagger + \nabla_{\mathbf{q}}\mathcal{L}^T\delta\mathbf{q} + \nabla_{\boldsymbol{\Theta}}\mathcal{L}^T\delta\boldsymbol{\Theta} \\
&= \mathcal{N}[\mathbf{q};\boldsymbol{\Theta}]^T\delta\mathbf{q}^\dagger && \text{— Governing equation} \\
&\quad + \left\{\nabla_{\mathbf{q}}\mathcal{J} - \left(\nabla_{\mathbf{q}}\mathcal{N}\right)^T\mathbf{q}^\dagger\right\}^T\delta\mathbf{q} && \text{— Adjoint equation} \\
&\quad + \left\{\nabla_{\boldsymbol{\Theta}}\mathcal{J} - \left(\nabla_{\boldsymbol{\Theta}}\mathcal{N}\right)^T\mathbf{q}^\dagger\right\}^T\delta\boldsymbol{\Theta}. && \text{— Gradient}
\end{aligned} \tag{11.10}$$

For a feasible state $\mathbf{q}$ and control $\boldsymbol{\Theta}$ the governing equation (11.1) is satisfied so the first term in (11.10) is zero. With a choice of Lagrange multiplier $\mathbf{q}^\dagger$ that satisfies (11.9a), the second term in (11.10) also becomes zero, and the first-order variation in the Lagrangian is explicitly dependent only on control $\delta\boldsymbol{\Theta}$,

$$\delta\mathcal{L} = \nabla_{\boldsymbol{\Theta}}\mathcal{L}^T\delta\boldsymbol{\Theta} \equiv \left\{\nabla_{\boldsymbol{\Theta}}\mathcal{J} - \left(\nabla_{\boldsymbol{\Theta}}\mathcal{N}\right)^T\mathbf{q}^\dagger\right\}^T\delta\boldsymbol{\Theta}, \tag{11.11a}$$

where

$$\mathcal{N}[\mathbf{q};\boldsymbol{\Theta}] = 0 \tag{11.11b}$$

$$\nabla_{\mathbf{q}}\mathcal{J} - \left(\nabla_{\mathbf{q}}\mathcal{N}\right)^T\mathbf{q}^\dagger = 0. \tag{11.11c}$$

Equation (11.11a) provides $\delta\mathcal{L}$ with respect to $\delta\boldsymbol{\Theta}$ in a linear sense, thus the gradient of $\mathcal{L}$ to $\boldsymbol{\Theta}$. Note that with the governing equation constraint (11.11b) $\mathcal{L} \equiv \mathcal{J}$ in (11.8) and the gradient (11.11a) is independent of $\delta\mathbf{q}$. Therefore, (11.11a) also indicates the gradient of $\mathcal{J}$ to $\boldsymbol{\Theta}$, with the equality constraint $\mathcal{N}[\mathbf{q};\boldsymbol{\Theta}] = 0$.

The gradient typically provides input for either line search method and trust-region method [57, 126]. Trust-region methods set up a predictive model for $\mathcal{J}$ using a local gradient and approximate Hessian information. We focus on line searches, which seem more compatible with the cost of evaluations of $\mathcal{J}$ for turbulence, although our methods should be broadly adaptable to gradient-informed searches. Details and comparison between line search method and trust-region method are available [57, 126, 128].

For line searches, with the gradient from (11.11), a local solution for the optimization problem (11.3) is sought in the framework of fixed-point iteration. This is the basis of the adjoint-based optimization, which will be introduced in the next section more specifically as direct adjoint looping (DAL). At k-th iteration, a line search direction is determined as an unit vector,

$$\delta\boldsymbol{\Theta}_k = \text{direction}(\nabla_{\boldsymbol{\Theta}}\mathcal{L}_1, \nabla_{\boldsymbol{\Theta}}\mathcal{L}_2, \ldots, \nabla_{\boldsymbol{\Theta}}\mathcal{L}_k), \tag{11.12}$$

where $\nabla_{\boldsymbol{\Theta}}\mathcal{L}_k$ denotes the gradient for the k-th iteration. Various algorithms can be used for this: steepest-descent method uses the gradient direction $\delta\boldsymbol{\Theta}_k = \nabla_{\boldsymbol{\Theta}}\mathcal{L}_k / \|\nabla_{\boldsymbol{\Theta}}\mathcal{L}_k\|$; and the conjugate-gradient method projects out all previous directions.

Once the line search direction is determined, a one-dimensional minimization problem is set up,

$$\alpha_k = \underset{\alpha}{\text{argmin}}\, \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}_{k-1} + \alpha\delta\boldsymbol{\Theta}_k] \tag{11.13a}$$

for a search step α_k, such that

$$\mathcal{N}[\mathbf{q}_k; \boldsymbol{\Theta}_{k-1} + \alpha_k\delta\boldsymbol{\Theta}_k] = 0. \tag{11.13b}$$

As for the overall optimization, finding the global line-search minimizer $\alpha \in (-\infty, \infty)$ is difficult. Various line minimization algorithms seek an approximate local minimizer. One way is to estimate α_k from a quadratic approximation of $\mathcal{J}(\alpha)$ [61],

$$\mathcal{J}[\mathbf{q}; \boldsymbol{\Theta}_{k-1} + \alpha\delta\boldsymbol{\Theta}_k] = \mathcal{J}[\mathbf{q}; \boldsymbol{\Theta}_{k-1}] + \alpha(\nabla_{\boldsymbol{\Theta}}\mathcal{J})^T\delta\boldsymbol{\Theta}_k + \frac{\alpha^2}{2}\delta\boldsymbol{\Theta}_k^T\mathcal{H}\delta\boldsymbol{\Theta}_k + \mathcal{O}(\alpha^3), \tag{11.14}$$

where $\mathcal{H} = \nabla_{\boldsymbol{\Theta}}\nabla_{\boldsymbol{\Theta}}\mathcal{J}$ is the Hessian of the cost functional. The closest local minimum estimated from (11.14)

is

$$\alpha_{\text{est}} = -\frac{(\nabla_{\boldsymbol{\Theta}}\mathcal{J})^T \delta\boldsymbol{\Theta}_k}{\delta\boldsymbol{\Theta}_k^T \mathcal{H} \delta\boldsymbol{\Theta}_k}, \tag{11.15}$$

though, again, this requires the Hessian and thus it is discouraged to even approximate for turbulent flow simulations. Inverse parabolic interpolation [128] estimates α_{est} with three pairs of function evaluations, instead of $\mathcal{H}$. Given three pairs $\{\alpha_1, \mathcal{J}(\alpha_1)\}$, $\{\alpha_2, \mathcal{J}(\alpha_2)\}$, and $\{\alpha_3, \mathcal{J}(\alpha_3)\}$ with $\alpha_1 < \alpha_2 < \alpha_3$ and $\mathcal{J}(\alpha_2)$ the minimum among them, the quadratic expansion (11.14) then provides the α estimated,

$$\alpha_{\text{est}} = \alpha_2 - \frac{1}{2}\frac{(\alpha_2-\alpha_1)^2[\mathcal{J}(\alpha_2)-\mathcal{J}(\alpha_3)] - (\alpha_2-\alpha_3)^2[\mathcal{J}(\alpha_2)-\mathcal{J}(\alpha_1)]}{(\alpha_2-\alpha_1)[\mathcal{J}(\alpha_2)-\mathcal{J}(\alpha_3)] - (\alpha_2-\alpha_3)[\mathcal{J}(\alpha_2)-\mathcal{J}(\alpha_1)]}. \tag{11.16}$$

Though (11.16) is not guaranteed stable on its own, convergence can be guaranteed by particular algorithms to find the $(\alpha_1, \alpha_2, \alpha_3)$, such as golden-section search [128]. More sophiscated approaches can provide better accuracy with more evaluations of functions, gradients, and Hessians, however this basic approach is sufficient for our goals [57, 128]. Algorithm 2 summarizes the overall procedure of gradient-based optimization. Once the solution $(\mathbf{q}_k, \mathbf{q}_k^\dagger, \boldsymbol{\Theta}_k)$ satisfies the terminal criterion in Algorithm 2, (11.9b) is satisfied under the

Algorithm 2 Nonlinear gradient-based line-search optimization

Given: initial guess $\boldsymbol{\Theta}_0$, tolerance ε, maximum search limit $K_{\max}$
Result: $(\mathbf{q}^*, \mathbf{q}^{\dagger *}, \boldsymbol{\Theta}^*) = \operatorname{argmin} \mathcal{L}[\mathbf{q}, \mathbf{q}^\dagger, \boldsymbol{\Theta}]$
Solve the governing equation $\mathcal{N}[\mathbf{q}; \boldsymbol{\Theta}_0] = 0$ for $\mathbf{q}_0$
$\mathcal{L} = \mathcal{J}[\mathbf{q}_0, \boldsymbol{\Theta}_0]$
Solve the adjoint equation $\nabla_{\mathbf{q}}\mathcal{J} - (\nabla_{\mathbf{q}}\mathcal{N})^T \mathbf{q}^\dagger = 0$ for $\mathbf{q}_0^\dagger$
$\nabla_{\boldsymbol{\Theta}}\mathcal{L}_1 = \nabla_{\boldsymbol{\Theta}}\mathcal{J} - (\nabla_{\boldsymbol{\Theta}}\mathcal{N})^T \mathbf{q}_0^\dagger$
for $k = 1, \ldots, K_{\max}$ **do**
  $\delta\boldsymbol{\Theta}_k = \text{direction}(\nabla_{\boldsymbol{\Theta}}\mathcal{L}_1, \nabla_{\boldsymbol{\Theta}}\mathcal{L}_2, \ldots, \nabla_{\boldsymbol{\Theta}}\mathcal{L}_k)$ ▷ Eq. (11.12)
  $(\mathbf{q}_k, \alpha_k) = \operatorname{argmin} \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}_{k-1} + \alpha\delta\boldsymbol{\Theta}_k]$ such that $\mathcal{N}[\mathbf{q}_k; \boldsymbol{\Theta}_{k-1} + \alpha\delta\boldsymbol{\Theta}_k] = 0$ ▷ Eq. (11.13)
  $\boldsymbol{\Theta}_k = \boldsymbol{\Theta}_{k-1} + \alpha_k\delta\boldsymbol{\Theta}_k$ ▷ Determine $(\mathbf{q}_k, \boldsymbol{\Theta}_k)$
  $\mathcal{L} = \mathcal{J}[\mathbf{q}_k, \boldsymbol{\Theta}_k]$
  Solve $\nabla_{\mathbf{q}}\mathcal{J} - (\nabla_{\mathbf{q}}\mathcal{N})^T \mathbf{q}^\dagger = 0$ for $\mathbf{q}_k^\dagger$ ▷ Determine $\mathbf{q}_k^\dagger$
  $\nabla_{\boldsymbol{\Theta}}\mathcal{L}_{k+1} = \nabla_{\boldsymbol{\Theta}}\mathcal{J} - (\nabla_{\boldsymbol{\Theta}}\mathcal{N})^T \mathbf{q}_k^\dagger$
  if $\|\nabla_{\boldsymbol{\Theta}}\mathcal{L}_{k+1}\| < \varepsilon$ **then**
    $(\mathbf{q}^*, \mathbf{q}^{\dagger *}, \boldsymbol{\Theta}^*) = (\mathbf{q}_k, \mathbf{q}_k^\dagger, \boldsymbol{\Theta}_k)$
    Exit

threshold ε, with both (11.9a) and (11.9c) exactly satisfied, thus the solution is close to the first-order optimality condition in Theorem 11.2.

Note that for the interleaved iterations between $(\mathbf{q}, \boldsymbol{\Theta})$ and $\mathbf{q}^\dagger$, the governing equation $\mathcal{N}[\mathbf{q}; \boldsymbol{\Theta}] = 0$ is enforced strictly, thus $\mathcal{L} \equiv \mathcal{J}$ is preserved throughout the optimization. It will be shown in Chapter 12 how this strict constraint, combined with chaotic dynamical systems, will frustrate the search for a meaningful local minimum by limiting the exploration. Its temporary relaxation will be a component of the

new algorithm.

11.3 Optimal control: Pontryagin's minimum principle

The method of Lagrange multipliers can be extended to optimal control of dynamical systems where (11.1) is a differential equation in time, so the governing equation is

$$\mathcal{N}[\mathbf{q};\boldsymbol{\Theta}] \equiv \frac{\partial \mathbf{q}}{\partial t} - \mathcal{R}[\mathbf{q};\boldsymbol{\Theta}] = 0, \tag{11.17a}$$

with initial condition

$$\mathbf{q}(t_i) = \mathbf{q}_0. \tag{11.17b}$$

Our formulation will be developed with turbulence applications in mind, though it can be extended to more general optimization.

For the purpose of introduction we consider systems with finite degrees of freedom. This will provide a foundation for formulation of our proposed algorithm in Chapter 14. We take $\mathbf{q}$ and $\mathcal{N}$ to be defined in a continuous function space with m degrees of freedom and $\boldsymbol{\Theta}$ with $c \leq m$ degrees of freedom,

$$\mathbb{Q} \equiv \mathbb{N} = U^m \qquad \mathbb{T} = U^c, \tag{11.18}$$

where $U = H^0(\mathbb{R}_0^+)$ is the space of L^2-functions from $[0,\infty)$ to $\mathbb{R}$. The inner product for $\mathbb{Q} \equiv U^m$ is

$$\langle \mathbf{p}, \mathbf{q} \rangle_{\mathbb{Q}} = \int_{t_i}^{t_f} \langle \mathbf{p}(t), \mathbf{q}(t) \rangle_{\mathbb{Q}^+} \; dt, \tag{11.19}$$

with t_i, t_f the initial and final times. For convenience, we define the subspace $\mathbb{Q}^+ = \mathbb{R}^m$ for the instantaneous state $\mathbf{q}(t)$ at any time t, and its inner product

$$\langle \mathbf{p}(t), \mathbf{q}(t) \rangle_{\mathbb{Q}^+} = \mathbf{p}^T(t)\mathbf{q}(t). \tag{11.20}$$

This notation $\langle \cdot, \cdot \rangle_{\mathbb{Q}^+}$ will be used throughout this study. Inner products for $\mathbb{T} \equiv U^c$ can be defined in a similar manner.

As for (11.3), optimal control seeks a trajectory of state $\mathbf{q}$ and control $\boldsymbol{\Theta}$ that minimizes a scalar objective

functional $\mathcal{J}$,

$$\mathcal{J}[\mathbf{q}, \mathbf{\Theta}] \equiv \Phi[\mathbf{q}(t_f)] + \int_{t_i}^{t_f} \mathcal{I}[\mathbf{q}(t), \mathbf{\Theta}(t)]\, dt, \qquad \text{such that} \quad \frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \mathbf{\Theta}] = \mathbf{0}, \tag{11.21}$$

where Φ is an objective functional for the final state $\mathbf{q}(t_f)$, and $\mathcal{I}$ is an instantaneous analog of $\mathcal{J}$ associated with the state and the control at time t.

We define a Lagrangian per (11.8) as

$$\begin{aligned} \mathcal{L}[\mathbf{q}, \mathbf{q}^\dagger, \mathbf{\Theta}] &= \mathcal{J}[\mathbf{q}, \mathbf{\Theta}] - \left\langle \mathbf{q}^\dagger, \frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \mathbf{\Theta}] \right\rangle_{\mathbb{Q}} \\ &= \Phi[\mathbf{q}(t_f)] + \int_{t_i}^{t_f} \mathcal{I}[\mathbf{q}(t), \mathbf{\Theta}(t)]\, dt - \int_{t_i}^{t_f} \mathbf{q}^{\dagger T} \left\{ \frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \mathbf{\Theta}] \right\} dt, \end{aligned} \tag{11.22}$$

where we introduce the co-state or the *adjoint* variable $\mathbf{q}^\dagger \in \mathbb{Q}$, a generalization of the Lagrange multiplier in section 11.2. In order to formulate an analog to Theorem 11.2 for optimal control $\mathbf{\Theta}^*$ and associated adjoint $\mathbf{q}^{\dagger *}$, we first derive the first-order variation of the Lagrangian,

$$\begin{aligned} \delta\mathcal{L} &= \mathcal{L}[\mathbf{q} + \delta\mathbf{q}, \mathbf{q}^\dagger + \delta\mathbf{q}^\dagger, \mathbf{\Theta} + \delta\mathbf{\Theta}] - \mathcal{L}[\mathbf{q}, \mathbf{q}^\dagger, \mathbf{\Theta}] \\ &= \frac{\partial \Phi}{\partial \mathbf{q}}^T \delta\mathbf{q}(t_f) + \int_{t_i}^{t_f} \left\{ \frac{\partial \mathcal{I}}{\partial \mathbf{q}}^T \delta\mathbf{q}(t) + \frac{\partial \mathcal{I}}{\partial \mathbf{\Theta}}^T \delta\mathbf{\Theta}(t) \right\} dt \\ &\quad - \int_{t_i}^{t_f} \mathbf{q}^{\dagger T} \left\{ \frac{d\delta\mathbf{q}}{dt} - \frac{\partial \mathcal{R}}{\partial \mathbf{q}} \delta\mathbf{q} - \frac{\partial \mathcal{R}}{\partial \mathbf{\Theta}} \delta\mathbf{\Theta} \right\} dt, \end{aligned} \tag{11.23}$$

where $\frac{\partial \mathcal{I}}{\partial \mathbf{q}} \in \mathbb{Q}$, $\frac{\partial \mathcal{I}}{\partial \mathbf{\Theta}} \in \mathbb{T}$ and $\frac{\partial \mathcal{R}}{\partial \mathbf{q}} : \mathbb{Q} \to \mathbb{Q}$, $\frac{\partial \mathcal{R}}{\partial \mathbf{\Theta}} : \mathbb{T} \to \mathbb{Q}$ are defined in the sense of a Frechét derivative. Recasting (11.23) as inner products,

$$\begin{aligned} \delta\mathcal{L} = &\left\langle \frac{\partial \Phi}{\partial \mathbf{q}}, \delta\mathbf{q}(t_f) \right\rangle_{\mathbb{Q}^+} + \left\langle \frac{\partial \mathcal{I}}{\partial \mathbf{q}}, \delta\mathbf{q} \right\rangle_{\mathbb{Q}} + \left\langle \frac{\partial \mathcal{I}}{\partial \mathbf{\Theta}}, \delta\mathbf{\Theta} \right\rangle_{\mathbb{T}} \\ &- \left\langle \mathbf{q}^\dagger, \frac{d\delta\mathbf{q}}{dt} - \frac{\partial \mathcal{R}}{\partial \mathbf{q}} \delta\mathbf{q} - \frac{\partial \mathcal{R}}{\partial \mathbf{\Theta}} \delta\mathbf{\Theta} \right\rangle_{\mathbb{Q}} \\ &- \left\langle \delta\mathbf{q}^\dagger, \frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \mathbf{\Theta}] \right\rangle_{\mathbb{Q}}, \end{aligned} \tag{11.24}$$

we can now define the adjoint operator [19, 129].

Definition 11.3 (Adjoint operator)**.** *For a linear operator $A : X \to Y$ defined on Hilbert spaces X and Y, a linear operator $A^\dagger : Y \to X$ is called the adjoint of A if it satisfies*

$$\langle Ax, y \rangle_Y = \langle x, A^\dagger y \rangle_X \qquad \forall (x, y) \in X \times Y. \tag{11.25}$$

Assuming that the adjoint of $\frac{\partial \mathcal{R}}{\partial \mathbf{q}}$ exists, we reorganize (11.24),

$$\begin{aligned}
\delta\mathcal{L} = &-\left\langle \delta\mathbf{q}^\dagger, \frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q};\boldsymbol{\Theta}] \right\rangle_{\mathbb{Q}} && \text{— Governing equation} \\
&+ \left\langle \frac{\partial \Phi}{\partial \mathbf{q}} - \mathbf{q}^\dagger(t_f), \delta\mathbf{q}(t_f) \right\rangle_{\mathbb{Q}^+} + \left\langle \frac{d\mathbf{q}^\dagger}{dt} + \frac{\partial \mathcal{R}}{\partial \mathbf{q}}^\dagger \mathbf{q}^\dagger + \frac{\partial \mathcal{I}}{\partial \mathbf{q}}, \delta\mathbf{q} \right\rangle_{\mathbb{Q}} && \text{— Adjoint equation} \\
&+ \left\langle \frac{\partial \mathcal{R}}{\partial \boldsymbol{\Theta}}^\dagger \mathbf{q}^\dagger + \frac{\partial \mathcal{I}}{\partial \boldsymbol{\Theta}}, \delta\boldsymbol{\Theta} \right\rangle_{\mathbb{T}} + \left\langle \mathbf{q}^\dagger(t_i), \delta\mathbf{q}(t_i) \right\rangle_{\mathbb{Q}^+} . && \text{— Gradient}
\end{aligned} \tag{11.26}$$

Each inner product in (11.26) reveals first-order partial dependency of $\mathcal{L}$ on $\mathbf{q}$, $\mathbf{q}^\dagger$, and $\boldsymbol{\Theta}$: the first inner product corresponding to $\mathbf{q}^\dagger$, the second and third to $\mathbf{q}$, and the fourth to $\boldsymbol{\Theta}$. For optimization of the initial condition, as for nonlinear non-modal stability analysis [72–76], the final inner product provides the gradient to the initial condition; if the initial condition is fixed, so $\delta\mathbf{q}(t_i) = 0$ and the final inner product is zero. Analogous to Theorem 11.2, at the extrema of the Lagrangian all these dependencies must be zero, which provides us a necessary condition for the optimal trajectory $\mathbf{q}^*$ and control $\boldsymbol{\Theta}^*$ [130]. This is formalized in Theorem 11.3.

Theorem 11.3 (Pontryagin's minimum principle)**.** *If $\mathbf{q}^*$ and $\boldsymbol{\Theta}^*$ are the optimal trajectory of the state and control for (11.21), then there exists an adjoint trajectory $\mathbf{q}^\dagger$ such that*

$$\text{Governing equation} \begin{cases} \dfrac{d\mathbf{q}^*}{dt} - \mathcal{R}[\mathbf{q}^*;\boldsymbol{\Theta}^*] = \mathbf{0} & \text{(11.27a)} \\ \mathbf{q}^*(t_i) = \mathbf{q}_0 & \text{(11.27b)} \end{cases}$$

$$\text{Adjoint equation} \begin{cases} \dfrac{d\mathbf{q}^\dagger}{dt} + \dfrac{\partial \mathcal{R}}{\partial \mathbf{q}}^\dagger \mathbf{q}^\dagger + \dfrac{\partial \mathcal{I}}{\partial \mathbf{q}} = \mathbf{0} & \text{(11.27c)} \\ \mathbf{q}^\dagger(t_f) = \dfrac{\partial \Phi}{\partial \mathbf{q}} & \text{(11.27d)} \end{cases}$$

$$\text{Optimality condition} \quad \boldsymbol{\Theta}^* = \underset{\boldsymbol{\Theta}}{\operatorname{argmin}}\, \mathcal{L}[\mathbf{q}^*, \mathbf{q}^\dagger, \boldsymbol{\Theta}]. \tag{11.27e}$$

Proof. See Pontryagin [131, Chap 2]. □

Equations (11.27a) and (11.27b) are constraints for the state variable, (11.27c) and (11.27d) are corresponding constraints for the associated adjoint variable, and (11.27e) is the condition for optimal control. As for the real vector space in section 11.2, we pursue optimal control (11.27e) by utilizing the gradient

information from the weak form (11.26),

$$\text{Gradient} \quad \delta\mathcal{L} = \left\langle \frac{\partial \mathcal{R}}{\partial \boldsymbol{\Theta}}^{\dagger} \mathbf{q}^{\dagger} + \frac{\partial \mathcal{I}}{\partial \boldsymbol{\Theta}}, \delta\boldsymbol{\Theta} \right\rangle_{\mathbb{T}} + \left\langle \mathbf{q}^{\dagger}(t_i), \delta\mathbf{q}(t_i) \right\rangle_{\mathbb{Q}^+}, \tag{11.28a}$$

where

$$\text{Governing equation} \begin{cases} \dfrac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \boldsymbol{\Theta}] = \mathbf{0} & (11.28b) \\ \mathbf{q}(t_i) = \mathbf{q}_0 & (11.28c) \end{cases}$$

$$\text{Adjoint equation} \begin{cases} \dfrac{d\mathbf{q}^{\dagger}}{dt} + \dfrac{\partial \mathcal{R}}{\partial \mathbf{q}}^{\dagger} \mathbf{q}^{\dagger} + \dfrac{\partial \mathcal{I}}{\partial \mathbf{q}} = \mathbf{0} & (11.28d) \\ \mathbf{q}^{\dagger}(t_f) = \dfrac{\partial \Phi}{\partial \mathbf{q}}. & (11.28e) \end{cases}$$

Algorithm 2 can be applied then to optimize.

11.4 Application to compressible flow

Adjoint-based optimization (direct adjoint looping) is in principle an extension of Theorem 11.3 to continua, in our case a gas, where the state is a function in space. This involves additional equality constraints, such as boundary conditions, which require additional formulation of the Lagrange multiplier. Here we specify the flow state, control, and the governing equation for compressible flow dynamics, following Vishnampet [132], though functionally the same as others [62, 76, 78, 79].

11.4.1 Equation governing a compressible fluid

For a domain $\mathbb{D}$, a non-empty bounded open subset of three-dimensional Euclidean space $\mathbb{R}^3$, we denote $V = H^0(\mathbb{D}) \times H^0(\mathbb{R}_0^+)$ as the space of L^2-functions from $\mathbb{D} \times [0, \infty)$ to $\mathbb{R}$. We define the flow state $\mathbf{q} \in \mathbb{Q} = V^5$ as

$$\mathbf{q} = \begin{pmatrix} \rho & \rho u_1 & \rho u_2 & \rho u_3 & \rho E \end{pmatrix}^T, \tag{11.29}$$

with ρ the density, $\mathbf{u} = (u_1, u_2, u_3) \in V^3$ the velocity, and $\rho E = \rho C_v T + \frac{1}{2}\rho u_i u_i$ the total energy. The inner product on $\mathbb{Q} \equiv V^5$ is defined as

$$\langle \mathbf{p}, \mathbf{q} \rangle_{\mathbb{Q}} = \int_{t_i}^{t_f} \langle \mathbf{p}(t), \mathbf{q}(t) \rangle_{\mathbb{Q}^+} \, dt, \tag{11.30}$$

with the inner product of instantaneous flow states $\mathbf{p}(t), \mathbf{q}(t) \in \mathbb{Q}^+ \equiv H^0(\mathbb{D})^5$ at time t,

$$\langle \mathbf{p}(t), \mathbf{q}(t) \rangle_{\mathbb{Q}^+} = \int_{\mathbb{D}} \mathbf{p}^T(\mathbf{x}, t) \mathbf{q}(\mathbf{x}, t) \, d\mathbf{x}, \tag{11.31}$$

where $\mathbf{p}(\mathbf{x}, t), \mathbf{q}(\mathbf{x}, t) \in \mathbb{R}^5$ are the flow states at position $\mathbf{x}$ and time t.

The governing equation without control is

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial}{\partial x_i} \left(\mathbf{F}_i^I - \mathbf{F}_i^V \right) = 0, \tag{11.32}$$

where $\mathbf{F}_i^I \in \mathbb{Q}$ and $\mathbf{F}_i^V \in \mathbb{Q}$ are respectively advective and diffusive fluxes in the x_i direction,

$$\mathbf{F}_i^I = \begin{pmatrix} \rho u_i \\ \rho u_1 u_i + p\delta_{i1} \\ \rho u_2 u_i + p\delta_{i2} \\ \rho u_3 u_i + p\delta_{i3} \\ u_i(\rho E + p) \end{pmatrix} \quad \text{and} \quad \mathbf{F}_i^V = \begin{pmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \tau_{3i} \\ u_j \tau_{ji} - q_i \end{pmatrix}. \tag{11.33}$$

The gas is taken to be ideal,

$$p = \rho R T, \tag{11.34}$$

with p pressure, T temperature, $R = (\gamma - 1) C_v$ the gas constant, and the ratio of specific heats $\gamma = C_p / C_v = 1.4$. The stress–strain constitutive relation for a Newtonian fluid is

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij}, \tag{11.35}$$

and Fourier's law of heat conduction is

$$q_i = -\kappa \frac{\partial T}{\partial x_i}, \tag{11.36}$$

with κ the thermal conductivity. The viscosity is modeled with power-law dependence on temperature,

$$\frac{\mu}{\mu_\infty} = \left(\frac{T}{T_\infty} \right)^{0.666}, \tag{11.37}$$

as a model of air [133], and subscript ∞ represents a quiescent ambient flow state. In (11.35), $\lambda = \mu_B - \frac{2}{3}\mu$ is the second coefficient of viscosity. For air, the bulk viscosity is taken to be $\mu_B = 0.6\mu$ [79, 132, 133].

The flow state $\mathbf{q}$ is further constrained by boundary conditions on the domain boundary $\partial\mathbb{D}$,

$$B_{\partial\mathbb{D}}\left[\mathbf{q}\right] = 0, \tag{11.38}$$

and various boundary condition are listed by Vishnampet [132, Table 4.1]. The specific boundary conditions used in flow simulations are introduced in Chapter 13.

We assume a generic form of control $\mathbf{f}$ to be added to the governing equation (11.32),

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial}{\partial x_i}\left(\mathbf{F}_i^I - \mathbf{F}_i^V\right) - \mathbf{W}_\Gamma(\mathbf{x}) \circ \mathbf{f}(\mathbf{x}, t) = 0, \tag{11.39}$$

where $\mathbf{W}_\Gamma = (W_\rho^\Gamma, W_{\rho u_1}^\Gamma, W_{\rho u_2}^\Gamma, W_{\rho u_3}^\Gamma, W_{\rho E}^\Gamma)^T \in \mathbb{Q}$ is a mollifying compact support that defines the actuator and the control region $\Gamma \subseteq \mathbb{D}$, and $\circ$ denotes element-wise multiplication,

$$\mathbf{W}_\Gamma \circ \mathbf{f} = \begin{pmatrix} W_\rho^\Gamma f_\rho & W_{\rho u_1}^\Gamma f_{\rho u_1} & W_{\rho u_2}^\Gamma f_{\rho u_2} & W_{\rho u_3}^\Gamma f_{\rho u_3} & W_{\rho E}^\Gamma f_{\rho E} \end{pmatrix}^T. \tag{11.40}$$

We recast (11.39) in the form of (11.17) as

$$\frac{\partial \mathbf{q}}{\partial t} = \mathcal{R}[\mathbf{q}, \mathbf{f}], \tag{11.41}$$

with the right-hand side

$$\mathcal{R}[\mathbf{q}, \mathbf{f}] = -\frac{\partial}{\partial x_i}\left(\mathbf{F}_i^I - \mathbf{F}_i^V\right) + \mathbf{W}_\Gamma(\mathbf{x}) \circ \mathbf{f}(\mathbf{x}, t).$$

Together, (11.41) and (11.38) governs compressible flow with control.

11.4.2 Numerical Discretization

Following Vishnampet [132], the spatial derivatives are discretized with flexible parameter r to be $2r$-order accurate centered-difference stencils at interior points of the domain and r-order accurate biased stencils near boundaries. This produces a banded matrix operators with the summation-by-parts (SBP) property, analogous to the intergration-by-parts of continuous derivative operators. For $r = 2, 3, 4$, these are referred to as the SBP 2–4, 3–6, 4–8 schemes, respectively. Approximating spatial derivatives with SBP operators is beneficial for computing the dual-consistent adjoint, which is exact to the arithmetic precision and at the same time consistently converges to the adjoint of the continuous governing equation [19, 76, 132, 134].

Second and mixed derivatives are discretized using repeated first-derivative SBP operators. This neces-

sitates the use of artificial dissipation [132, 135], since countered first derivative operators do not damp the highest wavenumbers supported by the mesh. To do this, we add a right-hand side term in (11.41)

$$\mathcal{R}_{\text{diss}} = -\sigma_{\text{diss}} \sum_{i=1}^{3} \overline{\mathbf{DI}}_i \mathbf{q}, \tag{11.42}$$

with strength $\sigma_{\text{diss}} > 0$. For the detail formulation of $\overline{\mathbf{DI}}_i$, we refer to Vishnampet [132].

The governing equation (11.41) is integrated in time by a standard explicit Runge–Kutta fourth-order (RK4) scheme. The inner product (11.30) is approximated with a quadrature norm,

$$\langle \vec{p}, \vec{q} \rangle_{\mathbb{Q}} = \sum_{n=1}^{N_t} \sum_{s=1}^{N_s} \langle \vec{p}^{\,n,s}, \vec{q}^{\,n,s} \rangle_{\mathbb{Q}^+} \beta^{n,s} \Delta t, \tag{11.43}$$

where the superscript $(\cdot)^{n,s}$ indicates the variable at the s-th stage of n-th time step. For our RK4 scheme, $N_s = 4$, $\beta^{n,1} = \beta^{n,4} = 1/6$ and $\beta^{n,2} = \beta^{n,3} = 1/3$. The inner product for the instantaneous flow state is discretized as

$$\langle \vec{p}^{\,n,s}, \vec{q}^{\,n,s} \rangle_{\mathbb{Q}^+} = (\vec{p}^{\,n,s})^T \mathbf{P}\, \vec{q}^{\,n,s}, \tag{11.44}$$

where the diagonal matrix $\mathbf{P}$ is the quadrature norm for space integral, with which SBP property of derivative operators are satisfied. The discrete-exact, dual-consistent adjoint solver developed by Vishnampet [132] is used, which, in the framework of (11.28), provides the gradient (11.28a) by solving the adjoint equation (11.28d-e) corresponding to (11.41).

Chapter 12

Chaos and gradient-based optimization

Two main aspects of how chaos disrupts gradient-based optimization are demonstrated with the optimal control problem of the Lorenz equation presented in Chapter 10. Standard results from the theory of dynamical systems are used to anticipate what property of chaotic dynamical systems challenges turbulence control, using simple mathematical models for illustration. As part of this analysis of simple cases, we introduce quantitative indicators that can also be applied to turbulence to characterize these same features in its less tractable setting.

12.1 Lorenz equation illustration

12.1.1 System definition

We recall the Lorenz equation (10.1) [80] for the state $\mathbf{q} = (x, y, z) \in \mathbb{Q} = U^3$ with actuation $\mathbf{\Theta} = f(t) \in \mathbb{T} = U$,

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z + f(t), \end{aligned}$$

with $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$. The initial condition for the example calculations is $\mathbf{q}_0 = (1.49, 1.49, 37)^T$. It is integrated with a standard explicit fourth-order Runge–Kutta (RK4) scheme, with time step $\Delta t = 0.01$. The two unstable fixed points U_1 and U_2 are

$$\begin{aligned} U_1 &= (-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho - 1)^T = (-6\sqrt{2},\, -6\sqrt{2},\, 27)^T \\ U_2 &= (\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho - 1)^T = (6\sqrt{2},\, 6\sqrt{2},\, 27)^T, \end{aligned} \tag{12.1}$$

and the saddle point is $S = (0, 0, 0)^T$. Figure 10.1 in the introduction shows the state trajectory for these parameters.

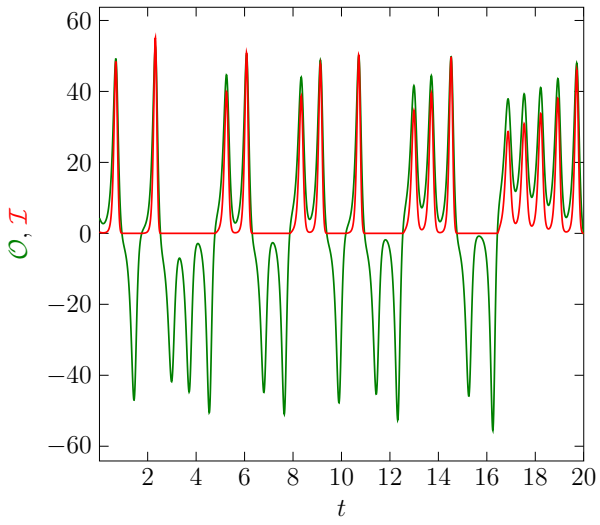


Figure 12.1: The observable $\mathcal{O}$ (12.2) and the instantaneous objective functional $\mathcal{I}$ (12.4) of the baseline trajectory.

12.1.2 Control

The control seeks orbits of U_1, avoiding orbits of U_2. The indicator

$$\mathcal{O}[\mathbf{q}] = 2x + y, \tag{12.2}$$

is positive for U_2 and negative for U_1. The objective functional $\mathcal{J}$ is defined to favor negative $\mathcal{O}$,

$$\mathcal{J} = \frac{1}{t_f - t_i} \int_{t_i}^{t_f} \mathcal{I}[\mathbf{q}] \, dt, \tag{12.3}$$

with

$$\mathcal{I}[\mathbf{q}] = \begin{cases} \frac{1}{2} \left(\frac{\mathcal{O}[\mathbf{q}]}{5} \right)^2 & \mathcal{O}[\mathbf{q}] \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{12.4}$$

which is shown in Figure 12.1. The optimization period is set to be $t_i = 0$ and $t_f = 20$.

The standard gradient-based optimization introduced in Chapter 11 is implemented to seek the control $f(t)$ that minimizes $\mathcal{J}$ (12.3). As was shown in Figure 10.2, the gradient of $\mathcal{J}$ to $f(t)$ grows exponentially in reverse time, and $\mathcal{J}$ is highly non-convex with numerous local extrema. Starting from $\mathbf{q}$ with $f = 0$ in Figure 10.1, the standard gradient-based method finds an f that reduces $\mathcal{J}$ by 44.6%, as shown in Figure 10.3. We further investigate the found f and associated $\mathbf{q}(t)$ to characterize the control strategy pursued with the baseline method.

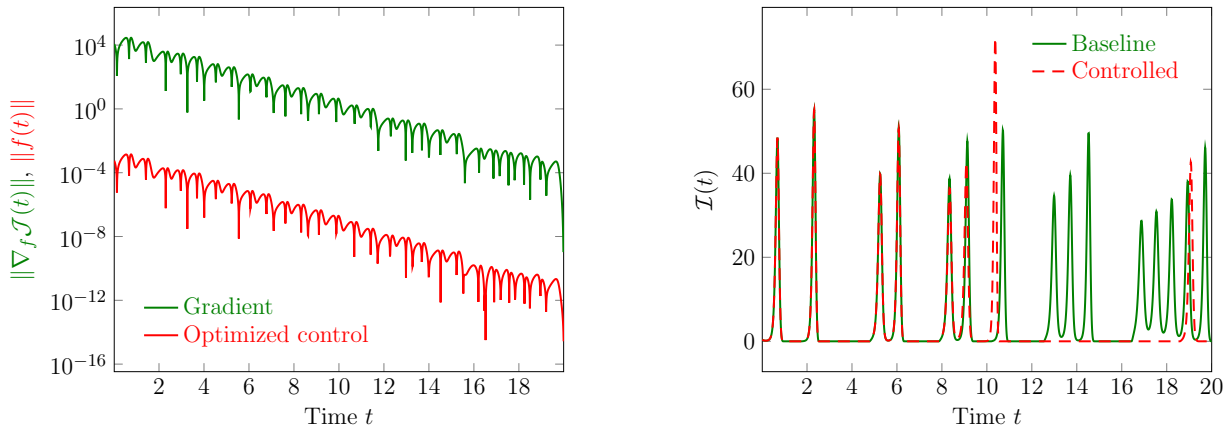


Figure 12.2: The standard gradient-based optimization for the Lorenz system (10.1). (a) The magnitude of the control gradient $\nabla_f \mathcal{J}$ and the optimized control $f(t)$ in time. (b) The instantaneous objective functional $\mathcal{I}(t)$ (12.4) for the baseline trajectory and the controlled trajectory.

12.1.3 Biased line search due to gradient growth

For gradient-based optimization in Chapter 11, a local minimum is sought in the gradient direction, which is interpreted as potentially reducing $\mathcal{J}$. However, the exponentially amplified gradients at early times in turn dominate the overall gradient. Figure 12.2 (a) confirms that the control is indeed concentrated, just as the gradient itself, in early times and therfore not expected to be maximally effective.

Such a distribution in time, however, masks the advantageous moments at which f can best minimize $\mathcal{J}$. Figure 12.2 (b) illustrates this by showing $\mathcal{I}(t)$ from (12.4) for the baseline trajectory, where the peaks correspond to U_2 orbits. Unlike $f(t)$, which is concentrated at the initial time, $\mathcal{I}$ peaks are distributed throughout the evolution. Moreover, $\mathcal{I}(t)$ for the controlled trajectory in Figure 12.2 (b) shows that the optimized control mostly affects late times, leaving the $\mathcal{I}(t)$ peaks in $t < 10$ nearly unaffected, even though they are closer in time to the strongest $f(t)$.

It would be tempting to say that the system (10.1) is not controllable in the sense that $\mathcal{J}$ has a high lower-bound. More specifically, it may seem that $\mathcal{I}$ events in the early time period are uncontrollable, because there is a time delay for a control $f(t)$ to influence on $\mathcal{I}(t)$. However, this is not the case. Since the Lorenz equation is a fairly simple ODE, it is possible to prove the controllability of the system (10.1) and design a nonlinear feedback control [136, 137], as summarized in Appendix C. Figure 12.3 (a) shows that $f(t)$ from this feedback control is synchronized with $\mathcal{O}$, being activated instantaneously whenever $\mathcal{O}$ nears 0. Figure 12.3 (b) shows that this control suppressed all peaks of $\mathcal{I}(t)$ throughout the optimization time period, hence the late-time peaks of $\mathcal{I}(t)$ do not need to exploit $f(t)$ at early times. Similarly, the early-time $\mathcal{I}$ peaks need not remain uncontrolled for the sake of late-time control.

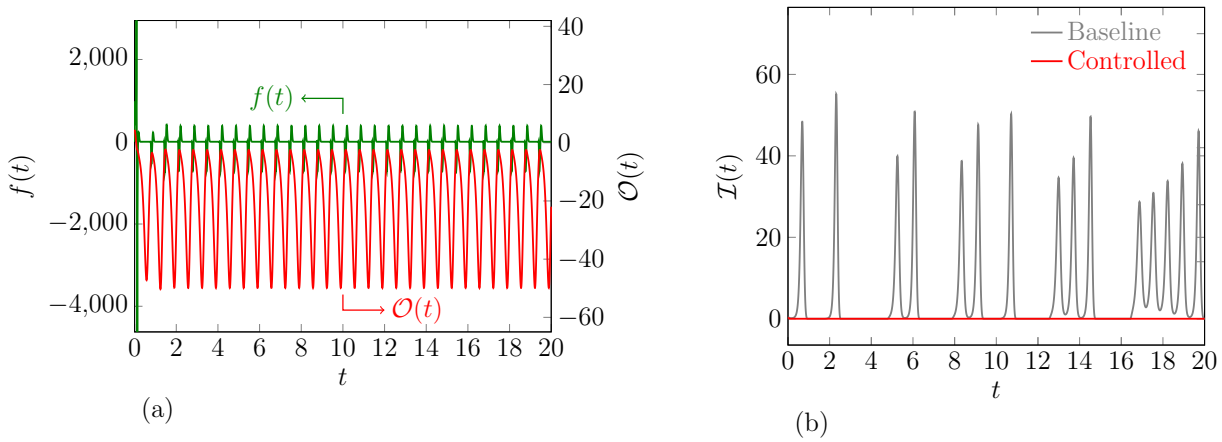


Figure 12.3: A nonlinear feedback control for the Lorenz system (10.1) designed in Appendix C. (a) The control force $f(t)$ and the corresponding observable $\mathcal{O}(t)$ (12.2) in time. (b) The instantaneous objective functional $\mathcal{I}(t)$ (12.4) for the baseline trajectory and the controlled trajectory.

12.1.4 Non-convexity of $\mathcal{J}$

It is also illuminating to compare the magnitude of $f(t)$ from the standard gradient-based method to the nonlinear feedback control. The nonlinear feedback control has the maximum magnitude $|f(t)| = 4623$ and oscillates in $-713 < f(t) < 378$ after $t > 0.5$, whereas the standard gradient-based method has peak $|f(t)| = 0.00141$ at $t = 0.76$. While it is impressive that such a small control can achieve more than 40% reduction, control amplitude is not penalized so there is no reason in this demonstration to pursue such an extraordinary control with low amplitude. While the extreme gradient magnitude in Figure 12.2 (a) may be interpreted as a potentially large $\mathcal{J}$ reduction, the realized $f(t)$ is extremely limited, which in turn does not match the $\mathcal{J}$ reduction of the nonlinear feedback control.

The non-convexity of $\mathcal{J}[\mathbf{\Theta}]$ limits the control magnitude. Figure 12.4 (a) shows that the line search steps taken in the optimization are small. The step size is only $\sim 10^{-24}$ after 18 line searches, suggesting that $f(t)$ converges to a local minimum. Figure 12.4 (b) shows $\mathcal{J}[\mathbf{\Theta}]$ finely sampled along a line-search direction, showing many local extrema of small scale. The shape of $\mathcal{J}$ in the entire $\mathrm{dom}(\mathcal{N})$ is expected to be similar, mainly in two ways: many local extrema as holes or hills are spread on the optimization space; or $\mathcal{J}$ has many ridges and valleys connected in a ragged way. Either way, limited size of these features obscures the gradient-based method from finding solutions far from the initial guess ($f = 0$ in this example).

Error amplification in computing gradients has been pointed out as the major cause of degrading optimization [76, 79, 82]. However, with this non-convexity of $\mathcal{J}$, the error involved in gradient computation is not necessarily the key impediment. Rather, it is the utility of the gradient for optimization that fails. Even the exact the gradient is valid only in the linear regime where the Taylor expansion of $\mathcal{J}[\mathbf{\Theta}_{k-1} + \alpha\delta\mathbf{\Theta}_k]$

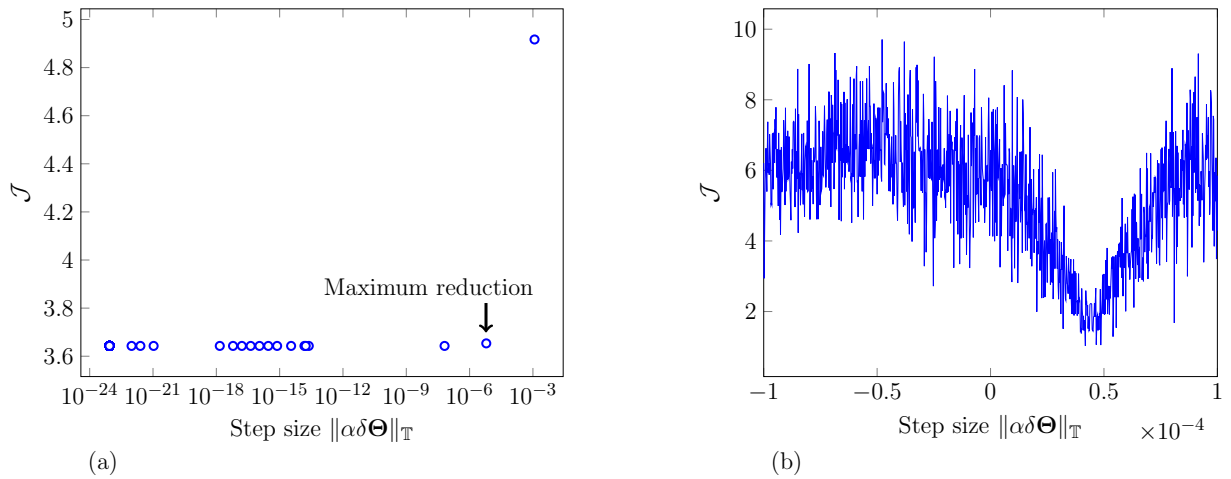


Figure 12.4: (a) The step sizes of each line searches (11.13) using the standard gradient-based method. The arrow indicates the line search step that achieve the most reduction of $\mathcal{J}$. (b) The objective functional $\mathcal{J}[\boldsymbol{\Theta}_{k-1} + \alpha\delta\boldsymbol{\Theta}_k]$ along the first line search direction (11.12).

in (11.14) remains useful. The linear regime becomes too small for useful nonlinear control (Figure 12.3) to be discovered. This limitation is intrinsic, due to the characteristic of gradient-based optimization: the gradients leads the search toward the closest local minimum in the linear regime.

While the standard gradient-based method uses the most basic optimization techniques, more sophisticated algorithms do not seem to remedy the situation. One may attempt to achieve more $\mathcal{J}$ reduction by taking an arbitrarily larger step size. In Figure 12.4 (b), for example, this might achieve the maximum reduction with $\|\alpha\boldsymbol{\Theta}\|_{\mathbb{T}} \approx 0.5 \times 10^{-4}$. There are established line search conditions that allow steps beyond the closest local minimum, with an expected $\mathcal{J}$ reduction estimated from the gradient [57]. Considering the pathologically large gradient in chaotic dynamical systems, such steps are still not expected to be sufficiently large or in a useful direction for control. For finding a better α at larger scale, the gradient-based line search becomes similar to a randomly chosen direction. The other gradient-based approaches introduced in Chapter 11 likewise suffer.

12.1.5 Error amplification and Taylor expansion breakdown

The error amplification and the Taylor expansion breakdown discussed in the previous section can be quantified. This will motivate the quantification of non-convexity of $\mathcal{J}$ introduced later in this chapter, with an aim toward extending this analysis to the challenge of flow turbulence.

We consider the finite-difference approximation of the gradient $\nabla_{\boldsymbol{\Theta}}\mathcal{J}$,

$$\frac{\Delta\mathcal{J}}{\Delta\Theta} = \frac{\mathcal{J}[\mathbf{q}; \boldsymbol{\Theta}_0 + \Delta\Theta\mathbf{e}_\theta] - \mathcal{J}[\mathbf{q}; \boldsymbol{\Theta}_0]}{\Delta\Theta}, \tag{12.5}$$

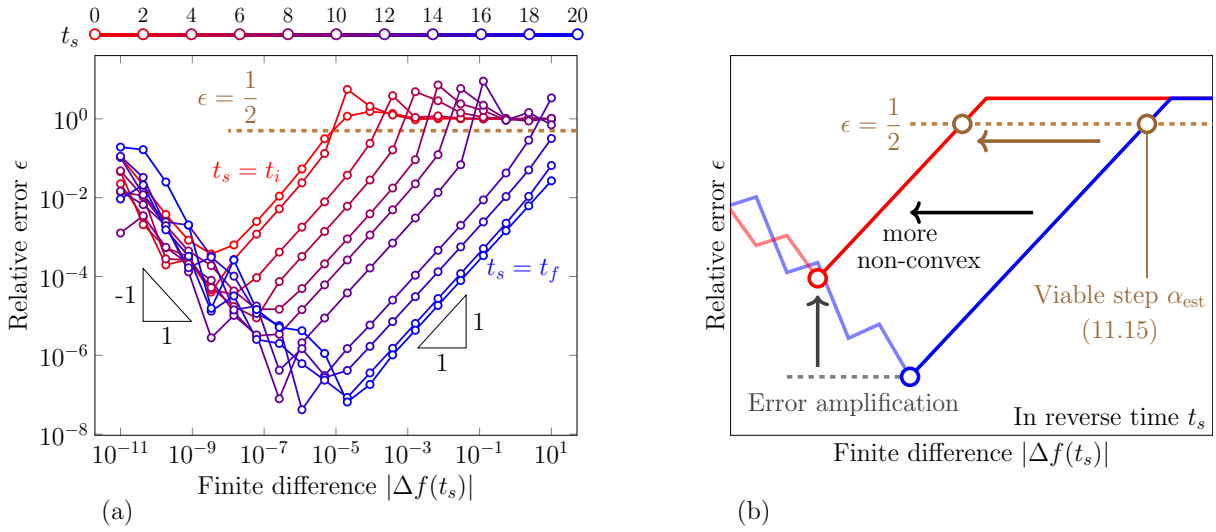


Figure 12.5: Utility of gradient in the Lorenz system. (a) The relative errors ϵ (12.6) of the finite-difference $\frac{\Delta \mathcal{J}}{\Delta f(t_s)}$ compared to $\frac{\partial \mathcal{J}}{\partial f(t_s)}$, the gradient of $\mathcal{J}$ (12.3) to the instantaneous forcing $\boldsymbol{\Theta} = f(t_s)$ at different times $t = t_s$. (b) A schematic of $\epsilon[\Delta f(t_s)]$ behavior in reverse time t_s for chaotic dynamical systems. τ_ϕ (12.36) is the time scale of these behaviors, which will be introduced in Section 12.3.

where $\mathbf{e}_\theta = \nabla_{\boldsymbol{\Theta}} \mathcal{J} / \|\nabla_{\boldsymbol{\Theta}} \mathcal{J}\|_{\mathbb{T}}$ is the unit vector in $\mathbb{T}$ along the gradient direction. This quantity has been compared to the gradient for the verification of the adjoint solver [82]. We also recognize that the relative difference between (12.5) and the adjoint gradient $\nabla_{\boldsymbol{\Theta}} \mathcal{J}$,

$$\epsilon[\Delta\Theta] = \left| \frac{\frac{\Delta \mathcal{J}}{\Delta \Theta} - \left\langle \frac{\partial \mathcal{J}}{\partial \boldsymbol{\Theta}}, \mathbf{e}_\theta \right\rangle_{\mathbb{T}}}{\left\langle \frac{\partial \mathcal{J}}{\partial \boldsymbol{\Theta}}, \mathbf{e}_\theta \right\rangle_{\mathbb{T}}} \right|, \tag{12.6}$$

quantifies how accurately the linearization with the gradient predicts the actual variation $\Delta \mathcal{J}$ for finite $\Delta\Theta$.

For the Lorenz example, ϵ is evaluated for $\frac{\partial \mathcal{J}}{\partial f(t_s)}$, with $\Delta\Theta = \Delta f(t_s)$, at the specific times $t_s \in [t_i, t_f]$ shown in Figure 12.5 (a). The $\mathcal{O}[\Delta f]$ and $\mathcal{O}[\Delta f^{-1}]$ asymptotes of ϵ shows the typical error scaling of a finite-difference, for truncation and finite-precision errors, respectively. Two impacts of chaos in Section 12.1.3 and 12.1.4 are manifest here, as illustrated in Figure 12.5 (b). First, minimum error of $\frac{\Delta \mathcal{J}}{\Delta f(t_s)}$ increases in reverse time of t_s, due to finite-precision errors amplified for longer time $(t_f - t_s)$. Furthermore, as $\mathcal{J}$ becomes more non-convex, the gradient is accurate over a diminishing range of $\Delta f(t_s)$. Appendix D further illustrates this with simple model $\mathcal{J}$'s. This gradient accuracy ϵ in Figure 12.5 clearly shows the decreasing utility of gradient due to chaos.

Figure 12.5 also confirms the expectation in Section 12.1.4, that the key impediment for optimization is non-convexity of $\mathcal{J}$ rather than the amplified error in gradient. In Chapter 11, typical step size is estimated

with (11.15) along the gradient direction $\mathbf{e}_\theta$,

$$\Delta\Theta_{\text{est}} = |\alpha_{\text{est}}| = \frac{\left\langle \frac{\partial \mathcal{J}}{\partial \boldsymbol{\Theta}}, \mathbf{e}_\theta \right\rangle_{\mathbb{T}}}{\langle \mathbf{e}_\theta, \mathcal{H}\mathbf{e}_\theta \rangle_{\mathbb{T}}}, \tag{12.7}$$

which, based on Tayler expansion (11.14), is expected to have a relative error

$$\epsilon[\Delta\Theta_{\text{est}}] = \frac{1}{2} + \mathcal{O}(\Delta\Theta_{\text{est}}^2). \tag{12.8}$$

In Figure 12.5 (a), these typical step sizes with $\epsilon = \frac{1}{2}$ are located at $\Delta\Theta > 10^{-5}$. As illustrated in Figure 12.5 (b), they are diminishing as $\mathcal{J}$ becomes non-convex. The error amplification mostly affects the minimum error at $\Delta\Theta < 10^{-5}$, much smaller than these estimated step sizes.

12.2 Signature of chaos: horseshoe mapping

The impacts on optimization observed in Section 12.1 can be connected more explicitly to the universal properties of chaotic dynamical systems. For illustration, we appeal to some standard mathematical models selected to reflect key dynamical properties of more complex systems.

12.2.1 Continuous dynamical system and discrete mapping

To facilitate the analysis we consider dynamical systems as discrete functions in time, mapping one instance to a latter instance [17, 114, 115]. To motivate this, we recall the governing equation (11.17) for a time-continuous dynamical system,

$$\frac{\partial \mathbf{q}}{\partial t} - \mathcal{R}[\mathbf{q}] = \mathbf{0} \tag{12.9a}$$

$$\mathbf{q}(t_i) = \mathbf{q}_0. \tag{12.9b}$$

The control $\boldsymbol{\Theta}$, which was in (11.17), is omitted in order to focus on the dynamical system properties. We represent the trajectory as $\mathbf{q}(\mathbf{q}_0, t)$, to explicitly denote that it is a continuous function of time t for initial condition $\mathbf{q}_0$.

The analogous discrete mapping $\mathbf{q}_n$ with a nominal time period ΔT is,

$$\mathbf{q}_{n+1} = \mathbf{q}(\mathbf{q}_n, \Delta T) \equiv \mathbf{q}_n + \int_{t_i+n\Delta T}^{t_i+(n+1)\Delta T} \mathcal{R}[\mathbf{q}]\, dt. \tag{12.10}$$

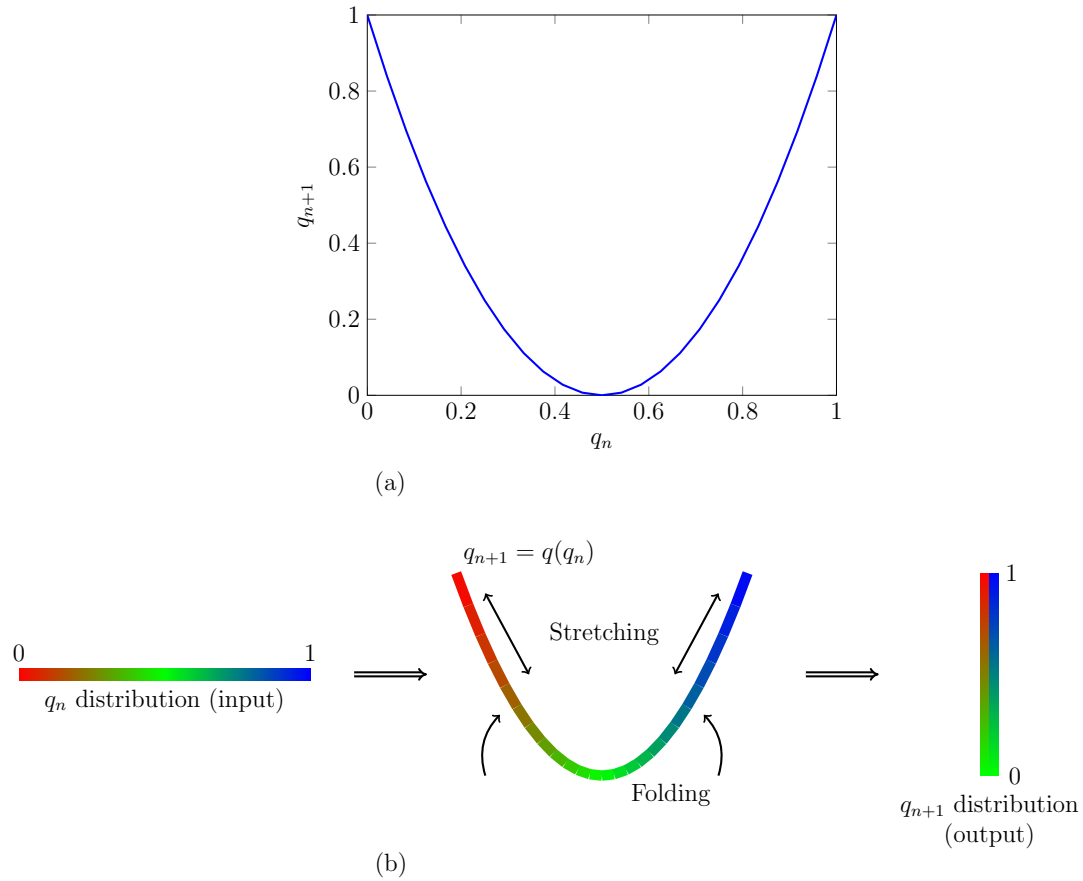


Figure 12.6: (a) Logistics map (12.11), and (b) the stretching and folding motion involved in each step.

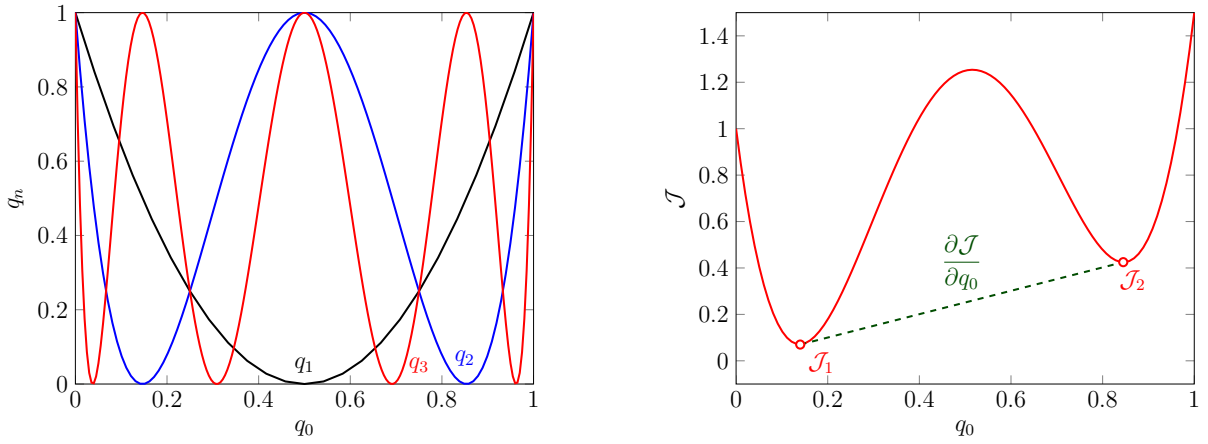
From this perspective, the evolution of any time-continuous dynamical systems can be considered a discrete mapping. The concept of mapping is convenient for analysis with its causal relation between the initial and subsequent states [17, 80, 114, 115].

12.2.2 Horseshoe mapping: illustrative example

As a specific example of (12.10), we use a variant of the one-dimensional logistics map [114],

$$q_{n+1} = q(q_n) \equiv (2q_n - 1)^2 \qquad q_n \in [0, 1]. \tag{12.11}$$

This is interpreted in Figure 12.6 (a) to stretch and fold a distribution of possible q_n to yield a q_{n+1} distribution. These stretching and folding features of this so-called horseshoe map are the two essential aspects of chaotic dynamical systems [17, 115]. The stretching amplifies the gradient of the states, by up to



(a) (b)

Figure 12.7: (a) Subsequent states q_1, q_2 and q_3 as functions of the initial state q_0 for the logistics map (12.11). (b) The objective function $\mathcal{J}(q_0)$ (12.13).

a factor of 2 at each step for this mapping,

$$\left|\frac{\partial q_{n+1}}{\partial q_0}\right| = \left|2(2q_n - 1)\frac{\partial q_n}{\partial q_0}\right| \leq 2\left|\frac{\partial q_n}{\partial q_0}\right|. \tag{12.12}$$

The well-known exponential growth of the gradient discussed in Section 12.1 is a consequence of such recursive stretching. The folding creates new local extrema in the next state $q_{n+1}(q_n)$, such as at $q_n = 0.5$ in Figure 12.6 (a). The number of local extrema also increases exponentially with increasing steps. Figure 12.7 (a) shows subsequent states $q_n(q_0)$, whose number of local extrema doubles at every mapping.

12.2.3 Non-convex $\mathcal{J}$ due to horseshoe mapping

The increasing number of local extrema in the state space of the dynamical system masks useful local minima of Definition 11.1 outside a small neighborhood. The simple objective function

$$\mathcal{J} = q_2 + \frac{1}{2}q_0 \tag{12.13}$$

can illustrate this. The two mappings that generate $q_2 = q(q(q_0))$ give $\mathcal{J}$ the three extrema, shown in Figure 12.7 (b). These are emblematic of the challenge introduced in Section 12.1. There is a distinct global minimum $\mathcal{J}_1$, however the local maximum at $q_0 = 0.5$ blocks the gradient optimization path to it: if an optimization for $\mathcal{J}$ starts near $\mathcal{J}_2$, the global minimum $\mathcal{J}_1$ cannot be found by standard gradient search.

Of course, this is a risk of the gradient methods described in Chapter 11, yet in many applications local

minima have large enough neighborhoods to be useful. However, for a chaotic dynamical system it poses a challenge, as shown in Section 12.1. The space of subsequent states becomes so non-convex by horseshoe mapping, that any objective functional dependent on them will also reflect this character. With many local minima of limited neighborhood, including useful ones, there is no expectation that the search will significantly improve the solution.

12.3 Quantification of the impact of chaos on optimization

We introduce two time scales that respectively quantify two characteristics of horseshoe mapping in the context of optimization. One quantifies the growth rate of the gradient, related to the stretching motion of the horseshoe mapping, and the other quantifies how fast (and how much) $\mathcal{J}$ becomes non-convex, related to the folding motion. Both will be employed to quantify more complex scenarios in Chapter 13.

12.3.1 Gradient growth

The Lyapunov exponent is well understood to quantify sensitivity to initial condition [114, 115]. Its inverse, the corresponding e-folding time scale [39], is used here as its quantitative indicator. We first introduce a specific definition of a Lyapunov exponent following Kuptsov and Parlitz [138], and then we introduce a procedure that can infer an approximate Lyapunov exponent in the context of optimization.

For a time-continuous dynamical system (11.17),

$$\frac{\partial \mathbf{q}}{\partial t} = \mathcal{R}[\mathbf{q}]$$

with initial condition

$$\mathbf{q}(t_i) = \mathbf{q}_0,$$

an infinitesimal trajectory deviation $\delta\mathbf{q}(t)$, starting from an initial perturbation $\delta\mathbf{q}(t_i) = \delta\mathbf{q}_0$, evolves according to the linearized governing equation,

$$\frac{\partial \delta\mathbf{q}}{\partial t} = \frac{\partial \mathcal{R}}{\partial \mathbf{q}}\,\delta\mathbf{q}, \tag{12.15}$$

where $\frac{\partial \mathcal{R}}{\partial \mathbf{q}} : \mathbb{Q} \to \mathbb{Q}$ is the Jacobian of the right-hand side $\mathcal{R}$, as in Chapter 11. For the Lorenz equation

(10.1),

$$\frac{\partial \mathcal{R}}{\partial \mathbf{q}} = \begin{pmatrix} -\sigma & \sigma & 0 \\ \rho - z & -1 & -x \\ y & x & -\beta \end{pmatrix} \quad \in \quad \mathbb{R}^{3\times 3}. \tag{12.16}$$

With this, the differential equation can be recast with a forward-time propagator or resolvent $\mathcal{F}[t_i, t_f]$,

$$\delta\mathbf{q}(t_f) = \mathcal{F}[t_i, t_f]\delta\mathbf{q}(t_i) \equiv \delta\mathbf{q}(t_i) + \int_{t_i}^{t_f} \frac{\partial \mathcal{R}}{\partial \mathbf{q}} \delta\mathbf{q}(t)\, dt. \tag{12.17}$$

This is the linearized counterpart of the full nonlinear forward-time propagator (12.10).

The growth of the deviation can be expressed in terms of $\mathcal{F}$ and $\delta\mathbf{q}(t_i)$ as

$$\frac{\|\delta\mathbf{q}(t_f)\|^2_{\mathbb{Q}^+}}{\|\delta\mathbf{q}(t_i)\|^2_{\mathbb{Q}^+}} = \frac{\langle \delta\mathbf{q}(t_f), \delta\mathbf{q}(t_f)\rangle_{\mathbb{Q}^+}}{\langle \delta\mathbf{q}(t_i), \delta\mathbf{q}(t_i)\rangle_{\mathbb{Q}^+}} = \frac{\langle \mathcal{F}\,\delta\mathbf{q}(t_i), \mathcal{F}\,\delta\mathbf{q}(t_i)\rangle_{\mathbb{Q}^+}}{\langle \delta\mathbf{q}(t_i), \delta\mathbf{q}(t_i)\rangle_{\mathbb{Q}^+}} = \frac{\langle \delta\mathbf{q}(t_i), \mathcal{F}^\dagger\mathcal{F}\,\delta\mathbf{q}(t_i)\rangle_{\mathbb{Q}^+}}{\langle \delta\mathbf{q}(t_i), \delta\mathbf{q}(t_i)\rangle_{\mathbb{Q}^+}}, \tag{12.18}$$

where $\mathcal{F}^\dagger$ is the adjoint of $\mathcal{F}$, so $\langle \mathbf{q}^\dagger(t_f), \mathcal{F}\mathbf{q}(t_i)\rangle_{\mathbb{Q}^+} = \langle \mathcal{F}^\dagger\mathbf{q}^\dagger(t_f), \mathbf{q}(t_i)\rangle_{\mathbb{Q}^+}$, and

$$\mathbf{q}^\dagger(t_i) = \mathcal{F}^\dagger[t_i, t_f]\mathbf{q}^\dagger(t_f), \tag{12.19}$$

with the adjoint variables $\mathbf{q}^\dagger(t_i), \mathbf{q}^\dagger(t_f) \in \mathbb{Q}^+$ at time t_i and t_f, respectively. The specific formulation of $\mathcal{F}^\dagger$ will be introduced subsequently. Considering that $\mathcal{F}^\dagger\mathcal{F}$ is a linear operator on $\mathbb{Q}^+$, (12.18) shows that the growth of $\|\delta\mathbf{q}\|^2_{\mathbb{Q}^+}$ for $t \in [t_i, t_f]$ is characterized by the eigenvalues of $\mathcal{F}^\dagger\mathcal{F}$.

The Lyapunov exponents can be expressed in terms of these eigenvalues. Denoting the eigenvalues of $\mathcal{F}^\dagger\mathcal{F}$ as $\sigma_k^2(t_i, t_f)$ during $[t_i, t_f]$, where $\sigma_1(t_i, t_f) \geq \sigma_2(t_i, t_f) \geq \cdots \geq 0$,

$$\tilde{\lambda}_k(t_i, t_f) = \frac{\log \sigma_k(t_i, t_f)}{t_f - t_i}, \tag{12.20}$$

This finite-time Lyapunov exponent [138–140] is trajectory-specific, not a property of the dynamical system, unless the trajectory $\mathbf{q}(t)$ covers the entire state space, which is only possible for $t \to \infty$ or an ensemble of infinitely many trajectories [138, 141].

A dynamical system has as many Lyapunov exponents as the state dimension $\dim(\mathbb{Q})$, and computing all of them precisely is computationally prohibitive for large-scale dynamical systems [138, 142]. However, for present purposes, it is sufficient to measure the fastest rate at which chaos impacts the optimization, which corresponds the leading Lyapunov exponent λ_1.

We introduce a procedure to estimate λ_1 in the context of optimization. For this, we introduce a final-state objective functional in (11.21),

$$\Phi[\mathbf{q}(t_f)] = \left\langle \mathbf{q}^\dagger(t_f), \mathbf{q}(t_f) \right\rangle_{\mathbb{Q}^+}, \tag{12.21}$$

where the choice of $\mathbf{q}^\dagger(t_f)$ will be specified subsequently. This is a scalar analog of the discrete mapping (12.10) in Section 12.2. The gradient of Φ to the initial condition $\mathbf{q}(t_i)$ is the special case of the optimal control problem in Section 11.3 with $\mathcal{I} \equiv 0$ and $\mathbf{\Theta} \equiv 0$. In this case, the variation (11.28a) in the weak form is then

$$\delta\mathcal{L} \equiv \delta\Phi = \left\langle \mathbf{q}^\dagger(t_i), \delta\mathbf{q}(t_i) \right\rangle_{\mathbb{Q}^+}, \tag{12.22a}$$

which provides the gradient $\frac{\partial\Phi}{\partial\mathbf{q}(t_i)} \equiv \mathbf{q}^\dagger(t_i)$. $\mathbf{q}^\dagger(t_i)$ is obtained by solving the adjoint equation (11.28d),

$$\frac{d\mathbf{q}^\dagger}{dt} + \frac{\partial\mathcal{R}}{\partial\mathbf{q}}^\dagger \mathbf{q}^\dagger = \mathbf{0}, \tag{12.22b}$$

starting from $\mathbf{q}^\dagger(t_f)$ at $t = t_f$. This gradient $\mathbf{q}^\dagger(t_i)$ is equivalent to the adjoint propagator (12.19). Comparing the variation of Φ,

$$\delta\Phi = \left\langle \mathbf{q}^\dagger(t_f), \delta\mathbf{q}(t_f) \right\rangle_{\mathbb{Q}^+} = \left\langle \mathbf{q}^\dagger(t_f), \mathcal{F}\delta\mathbf{q}(t_i) \right\rangle_{\mathbb{Q}^+} = \left\langle \mathcal{F}^\dagger\mathbf{q}^\dagger(t_f), \delta\mathbf{q}(t_i) \right\rangle_{\mathbb{Q}^+}, \tag{12.23}$$

where the second equality comes from (12.17) and the last from the definition of adjoint operator. Comparing (12.23) and (12.22a), $\mathbf{q}^\dagger(t_i) = \mathcal{F}^\dagger\mathbf{q}^\dagger(t_f)$, thus recasting the differential equation (12.22b),

$$\mathbf{q}^\dagger(t_i) = \mathcal{F}^\dagger[t_i, t_f]\mathbf{q}^\dagger(t_f) \equiv \mathbf{q}^\dagger(t_f) + \int_{t_i}^{t_f} \frac{\partial\mathcal{R}}{\partial\mathbf{q}}^\dagger \mathbf{q}^\dagger \, dt. \tag{12.24}$$

The adjoint counterpart of (12.18) can be used with $\frac{\partial\Phi}{\partial\mathbf{q}(t_i)} \equiv \mathbf{q}^\dagger(t_i)$ for computing Lyapunov exponents,

$$\frac{\left\|\mathbf{q}^\dagger(t_i)\right\|^2_{\mathbb{Q}^+}}{\left\|\mathbf{q}^\dagger(t_f)\right\|^2_{\mathbb{Q}^+}} = \frac{\left\langle \mathbf{q}^\dagger(t_i), \mathbf{q}^\dagger(t_i) \right\rangle_{\mathbb{Q}^+}}{\left\langle \mathbf{q}^\dagger(t_f), \mathbf{q}^\dagger(t_f) \right\rangle_{\mathbb{Q}^+}} = \frac{\left\langle \mathbf{q}^\dagger(t_f), \mathcal{F}\mathcal{F}^\dagger\mathbf{q}^\dagger(t_f) \right\rangle_{\mathbb{Q}^+}}{\left\langle \mathbf{q}^\dagger(t_f), \mathbf{q}^\dagger(t_f) \right\rangle_{\mathbb{Q}^+}}, \tag{12.25}$$

where $\mathcal{F}\mathcal{F}^\dagger$ has the same eigenvalues of $\mathcal{F}^\dagger\mathcal{F}$ [138]. Its eigenvalues represent growth in reverse time.

Any non-pathological choice of $\mathbf{q}^\dagger(t_f)$ will have at least a small component parallel to the leading eigenvector (backward Lyapunov vector [138]) that will therefore dominate the rest for a sufficiently long time $(t_f - t_i)$. Therefore, the leading finite-time Lyapunov exponent (12.20) can be inferred using (12.24) and

(12.25),

$$\tilde{\lambda}_1(t_i, t_f) = \frac{1}{t_f - t_i} \log \frac{\|\mathbf{q}^\dagger(t_i)\|_{\mathbb{Q}^+}}{\|\mathbf{q}^\dagger(t_f)\|_{\mathbb{Q}^+}}. \tag{12.26}$$

The adjoint state $\mathbf{q}^\dagger(t_f)$ is chosen with a random direction and $\left\|\mathbf{q}^\dagger(t_f)\right\|_{\mathbb{Q}^+} \leq 10^{-5}$. Specific $\mathbf{q}^\dagger(t_f)$ are introduced in Chapter 13.

To sample the possible state distribution, we ensemble-average N $\mathbf{q}(t)$ trajectories. All have the same interval length, sampled at different times. The leading Lyapunov exponent is approximated as

$$\lambda_1 \approx \frac{1}{N} \sum_{n=1}^{N} \tilde{\lambda}_1(t_{i,n}, t_{f,n}), \tag{12.27}$$

with n-th sample interval $[t_{i,n}, t_{f,n}]$. Problem-dependent intervals are specified in Chapter 13. The e-folding time is then

$$\tau_\lambda = \frac{1}{\lambda_1}. \tag{12.28}$$

12.3.2 Non-convexity of $\mathcal{J}$

Motivated by the analysis in Section 12.1.5, we measure the decay time scale of the viable step size (12.7) as an indicator of $\mathcal{J}$ non-convexity. To infer the viable step, we utilize the error ϵ (12.6). Based on Taylor expansion (11.14),

$$\epsilon[\Delta\Theta] = \frac{\langle \mathbf{e}_\theta, \mathcal{H}\mathbf{e}_\theta \rangle_{\mathbb{T}}}{\left\langle \frac{\partial \mathcal{J}}{\partial \boldsymbol{\Theta}}, \mathbf{e}_\theta \right\rangle_{\mathbb{T}}} \frac{\Delta\Theta}{2} + \mathcal{O}\left(\frac{\epsilon_r}{\Delta\Theta}\right) + \mathcal{O}(\Delta\Theta^2).$$

We note that the first-order coefficient of $\Delta\Theta$ is equivalent to the reciprocal of (12.7). So the viable step is inferred approximately with (12.6) at the minimum error,

$$\alpha_{\text{est}} = \frac{\left\langle \frac{\partial \mathcal{J}}{\partial \boldsymbol{\Theta}}, \mathbf{e}_\theta \right\rangle_{\mathbb{T}}}{\langle \mathbf{e}_\theta, \mathcal{H}\mathbf{e}_\theta \rangle_{\mathbb{T}}} \approx \frac{1}{2} \frac{\operatorname{argmin} \epsilon[\Delta\Theta]}{\min \epsilon[\Delta\Theta]}. \tag{12.29}$$

We define it as the viable step for $\mathcal{J}$ and $\boldsymbol{\Theta}$,

$$\delta\Theta[\mathcal{J}, \boldsymbol{\Theta}, t_i, t_f] = \frac{1}{2} \frac{\operatorname{argmin} \epsilon[\Delta\Theta]}{\min \epsilon[\Delta\Theta]}, \tag{12.30}$$

where its dependency on $\mathcal{J}$ and $\boldsymbol{\Theta}$ is explicitly specified.

We choose the same $\mathcal{J} = \Phi$ (12.21) with $\boldsymbol{\Theta} = \mathbf{q}(t_i)$ used for the Lyapunov exponent. In this way, the viable step represents a dynamical system property just as the Lyapunov exponent. The gradient $\frac{\partial \Phi}{\partial \mathbf{q}(t_i)} = \mathbf{q}^\dagger(t_i)$ is given from (12.22). A finite-difference (12.5) is evaluated with $\mathcal{J} = \Phi$ in (12.21) and

$\boldsymbol{\Theta} = \mathbf{q}(t_i)$,

$$\frac{\Delta\Phi}{\Delta\Theta} = \frac{\Phi[\mathbf{q}_{\Delta\Theta}(t_f)] - \Phi[\mathbf{q}(t_f)]}{\Delta\Theta}, \tag{12.31}$$

where $\mathbf{q}_{\Delta\Theta}$ is evaluated with the full governing equation (11.17) with perturbed initial condition,

$$\mathbf{q}_{\Delta\Theta}(t_i) = \mathbf{q}_0 + \Delta\Theta\mathbf{e}_\theta, \tag{12.32}$$

where $\mathbf{e}_\theta = \nabla_{\boldsymbol{\Theta}}\mathcal{J}/\|\nabla_{\boldsymbol{\Theta}}\mathcal{J}\|_{\mathbb{T}} = \mathbf{q}^\dagger(t_i)/\|\mathbf{q}^\dagger(t_i)\|_{\mathbb{Q}^+}$. The relative gradient error is recast from (12.6) as

$$\epsilon[\Delta\Theta] = \left| \frac{\frac{\Delta\Phi}{\Delta\Theta} - \|\mathbf{q}^\dagger(t_i)\|_{\mathbb{Q}^+}}{\|\mathbf{q}^\dagger(t_i)\|_{\mathbb{Q}^+}} \right|, \tag{12.33}$$

from which $\delta\Theta[\Phi, \mathbf{q}(t_i), t_i, t_f]$ is evaluated from (12.30). Our purpose is to quantify how typical $\mathcal{J}$ can become non-convex in time, so we evaluate the decay of $\delta\Theta$ (12.30) as

$$\tilde{\phi}(\Phi, \mathbf{q}(t_i), t_i, t_f) = -\frac{\log \delta\Theta[\Phi, \mathbf{q}(t_i), t_i, t_f]}{t_f - t_i}, \tag{12.34}$$

which is analogous to local Lyapunov exponent (12.20).

As for the Lyapunov exponent (12.27), in order to cover the entire state distribution, we ensemble-average $\tilde{\phi}$ (12.34),

$$\phi = \frac{1}{N}\sum_{n=1}^{N} \tilde{\phi}(\Phi_n, \mathbf{q}(t_{i,n}), t_{i,n}, t_{f,n}), \tag{12.35}$$

where $\Phi_n = \left\langle \mathbf{q}^\dagger(t_{f,n}), \mathbf{q}(t_{f,n}) \right\rangle_{\mathbb{Q}^+}$. The adjoint states $\mathbf{q}^\dagger(t_{f,n})$ and time intervals $[t_{i,n}, t_{f,n}]$ are from the same sample used for the Lyapunov exponent (12.27). This yields a time scale

$$\tau_\phi = \frac{1}{\phi}, \tag{12.36}$$

equivalent to e-folding time (12.28).

There are many quantities for complexity of a chaotic dynamical system, such as metric entropy [17, 114–120, 141]. We recognize that they may be indirectly connected with the non-convexity of the state $\mathbf{q}$ and associated $\mathcal{J}$, though most of them are intractable to compute for large-scale flow simulations. In Appendix E we reviewed some of them in connection with folding motion, and discuss challenges of their computation for large-scale flow simulations.

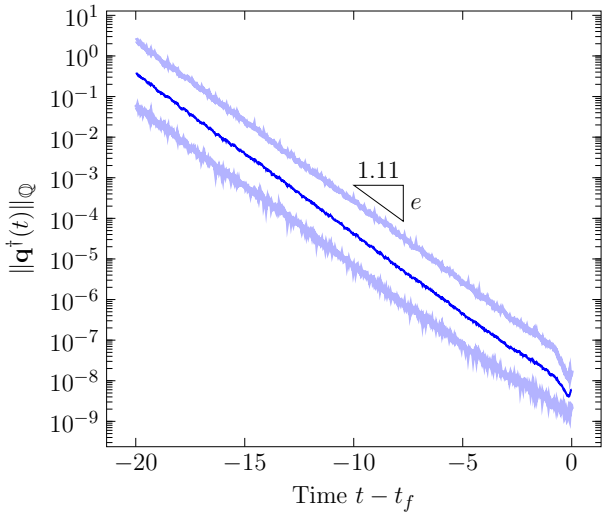


Figure 12.8: Evolution of $\left\|\mathbf{q}^\dagger(t)\right\|$ for the Lorenz system (10.1). The dark blue line indicates the geometric average over ensemble, and the light blue lines indicate the standard deviation around the average.

12.4 Application to the Lorenz example

The e-folding time (12.28) of the Lorenz equation (10.1) is estimated with $N = 500$ adjoint final states,

$$\mathbf{q}^\dagger(t_f) = (\xi_x, \xi_y, \xi_z), \tag{12.37}$$

where ξ_x, ξ_y, and ξ_z are pseudo-random numbers with uniform distribution $U[-10^{-5}, 10^{-5}]$. The adjoint initial state $\mathbf{q}^\dagger(t_i)$ is computed from each adjoint final state through (12.24) for $t_f - t_i = 20$ with a randomly selected $t_f \in U[20, 10^4]$. Figure 12.8 shows the time-evolution of the ensemble-average of $\mathbf{q}^\dagger(t)$, where the ensemble is averaged geometrically,

$$\overline{\mathbf{q}^\dagger} = \left(\prod_{k=1}^{N} \|\mathbf{q}_k^\dagger\|_{\mathbb{Q}^+}\right)^{\frac{1}{N}}. \tag{12.38}$$

A linear fit shows that the e-folding time (12.28) is $\tau_\lambda \approx 1.11$. At $t = t_i$, the gradient increases by a factor of $\exp[(t_f - t_i)/\tau_\lambda] \approx 7 \times 10^7$, which is consistent with the observation from Figure 12.2 (a).

The sample final adjoint states (12.37), used to estimate τ_λ, are reused to evaluate the functional Φ (12.21) for estimating τ_ϕ. The $\mathbf{q}^\dagger(t)$ provide the gradient (12.22) at time t that will be compared to the finite-difference (12.31). Figure 12.9 (a) shows a sample $\mathbf{q}^\dagger(t)$, which grows in reverse time t. 10 time points t are selected, and their $\mathbf{q}^\dagger(t)$ are compared to finite-differences $\frac{\Delta\Phi}{\Delta\Theta}$ (12.31) to estimate relative errors $\epsilon[\Delta\Theta]$ (12.33) as shown in Figure 12.9 (b). The viable step $\delta\Theta$ (12.30) in Figure 12.9 (c) is ensemble-averaged over $N = 100$ samples in Figure 12.9 (d). Their decay time scale τ_ϕ (12.36) is estimated to be $\tau_\phi \approx 1.135$.

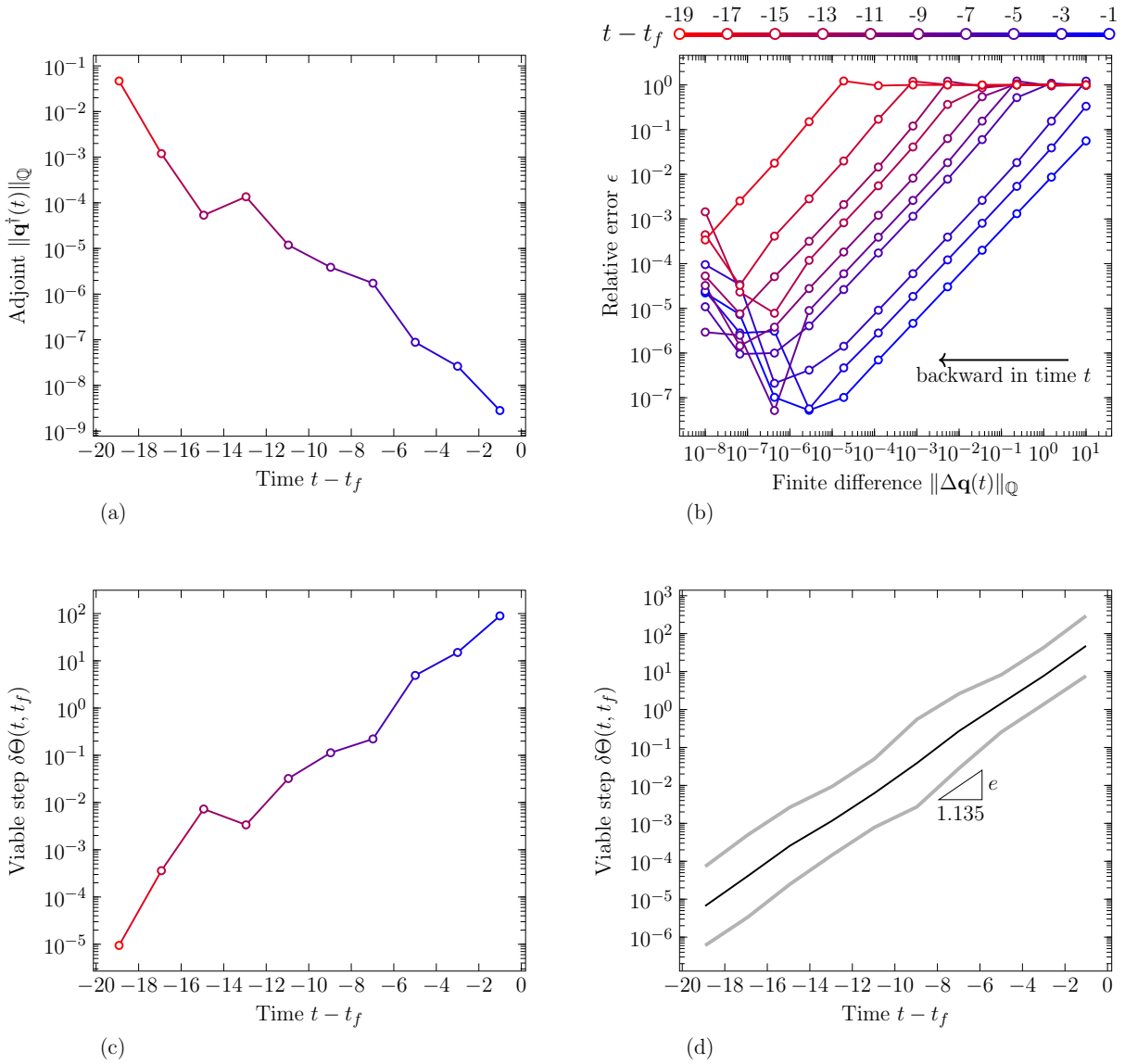


Figure 12.9: Computation of viable step of the linear functional (12.21). (a) The adjoint magnitude at the investigation times, (b) relative error of the finite-difference compared to the gradient at each investigation time, (c) the viable step in time estimated from the relative errors, and (d) ensemble average of the viable steps from sample adjoint final states.

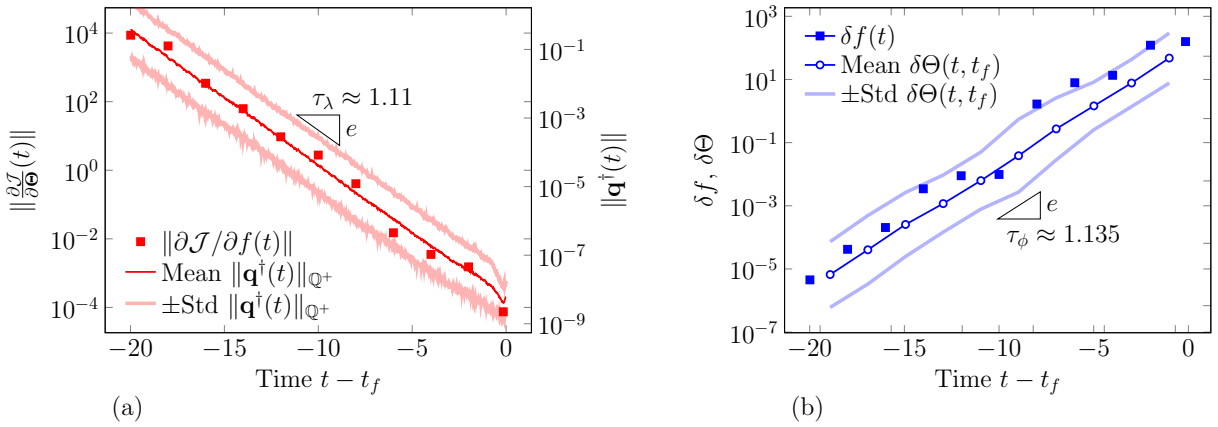


Figure 12.10: The impact of chaotic dynamics on the objective functional (12.3). The time axis is shifted with respect to the final simulation time t_f. (a) The gradient to a point-wise force $f(t_s)$ (circle) compared with ensemble average of $\mathbf{q}^\dagger(t)$ (solid) from Figure 12.8, and (b) the viable step associated with the gradient $\frac{\partial \mathcal{J}}{\partial f(t_s)}$ (circle) compared with ensemble average of $\delta\Theta(t_i, t_f)$ (solid) from Figure 12.9 (d).

The gradient growth and inferred viable step for the $\mathcal{J}$ (12.3) and control $f(t)$ in (10.1) confirms that these τ_λ and τ_ϕ quantify the impact of chaos on the optimization. Figure 12.10 (a) shows the gradient of $\frac{\partial \mathcal{J}}{\partial f(t)}$ at 10 time points, which grows in the time scale of τ_λ. Its associated viable step per (12.30) is also decaying in the time scale of τ_ϕ, as shown in Figure 12.10 (b). Not only the decay time scale, but also the magnitude of $\delta\Theta$ itself seems informative for the typical search step size in optimization. In Figure 12.10 (b), $\delta\Theta(t, t_f) \approx 10^{-5}$ at $t = t_i$ matches with the useful step size that reduced $\mathcal{J}$ in Figure 12.4 (a). After this, the step size in the optimization is significantly reduced, suggesting that the solution reached a local minimum. This efficacy will be further evaluated for more complex systems in Chapter 13.

We note that the similarity of $\tau_\phi \approx 1.135$ to $\tau_\lambda \approx 1.11$ is not expected for all cases, as will be seen when these analysis procedures are applied to more involved problems. Since the Lorenz system has only one positive Lyapunov exponent, many of its invariant quantities, such as the metric entropy, share the same value as the leading Lyapunov exponent.

Chapter 13

Example applications

The mechanisms by which chaos hinders optimization that were illustrated in Chapter 12 are now demonstrated and quantified for model chaotic systems with increasing complexity, and ultimately turbulence. In Section 13.1 we first demonstrate standard adjoint-based control for the one-dimensional Kuramoto–Sivashinsky (K–S) chaotic advection-diffusion system [121, 122, 143–145]. Two-dimensional Kolmogorov flow in Section 13.2 is a chaotic, though not turbulent, compressible flow (11.41), where vortices chaotically oscillate and meander in a small domain [124, 125]. Together these two systems allow us to detail the challenges analyzed in Chapter 12 before attempting turbulence. In Section 13.3 we set up an advection + K–S equation model system, which illustrates the challenge of turbulence with the scale-separation hypothesis of Chapter 10 in mind. A three-dimensional Kolmogorov flow in Section 13.4 provides an ultimate example of turbulent flows under scale-separation hypothesis, where the large-scale vorticity structures, similar to its two-dimensional counterpart, are now continually scattered into a broadbanded turbulence.

13.1 The Kuramoto–Sivashinsky system

13.1.1 Governing equation

The Kuramoto–Sivashinsky (K–S) equation is a phenomenological model for concentration-wave propagation in dissipative media [121] and laminar flame front stability [122]. We consider the generalized K–S equation [146] for the state $u(x,t) \in \mathbb{Q} = V$ with control $f(x,t) \in \mathbb{T} = V$,

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) + \alpha_1 \frac{\partial^2 u}{\partial x^2} + \alpha_2 \frac{\partial^4 u}{\partial x^4} - W_\Gamma(x) f(x,t) = 0 \qquad (x,t) \in [0, 2\pi] \times [0, \infty), \tag{13.1}$$

with $\alpha_1 = 1.0$, $\alpha_2 = 0.029910$, and W_Γ the compact mollifying support defining the control region, which will be introduced subsequently. With these α_1 and α_2 values, the system (13.1) exhibits chaos [143]. The spatial domain is 2π-periodic,

$$u(x,t) = u(x + 2\pi, t). \tag{13.2}$$

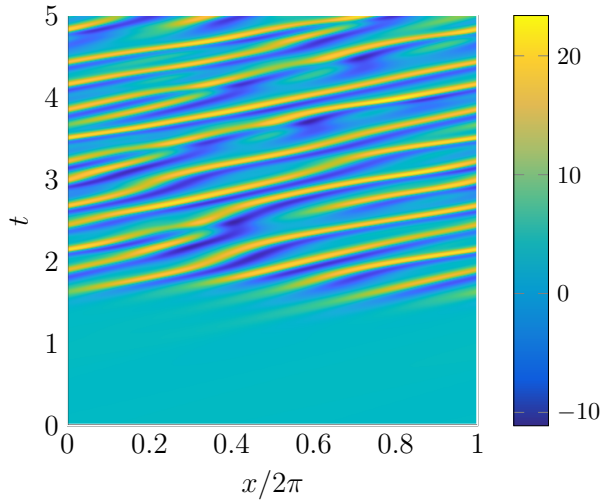


Figure 13.1: The evolution of $u(x,t)$ in space and time.

The initial condition is,

$$u(x,0) = 5 + 0.1\sin(x)\,, \tag{13.3}$$

so that the characteristic advection speed is $U_c = 5$. Figure 13.1 shows the evolution of $u(x,t)$.

13.1.2 Numerical method

For the demonstration calculations the spatial domain is discretized with $N_g = 512$ mesh points, and the spatial derivative $\partial/\partial x$ is discretized with the standard three-point second-order centered difference, which we denote as an operator $\mathbf{D} \in \mathbb{R}^{N_g} \times \mathbb{R}^{N_g}$. The higher derivatives in (13.1) are discretized with repeated first-derivatives, as $\mathbf{D}^2$ and $\mathbf{D}^4$ respectively. Since the second-order derivative is anti-diffusive ($\alpha_1 > 0$) and the fourth-order derivative is stiff for high wavenumber modes, an implicit time-integration aids stability. This is convenient since these terms are linear. Yet the nonlinear advective term $\frac{\partial}{\partial x}(\frac{1}{2}u^2)$ in (13.1) is simpler when explicitly time integrated. Therefore, a semi-explicit four-step Runge–Kutta scheme is used with time

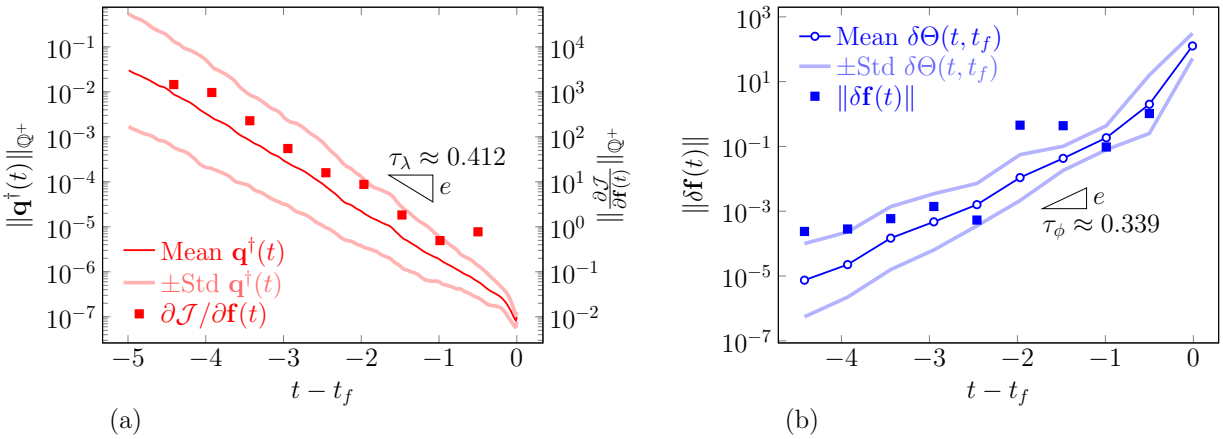


Figure 13.2: K–S equation (13.1): (a) The ensemble average of the adjoint $\mathbf{q}^\dagger(t)$ and the standard deviation around the average. The gradient of $\mathcal{J}$ (13.5) with respect to control forcing $f(t)$ is also for comparison. (b) The ensemble average of the viable step $\delta\Theta(t, t_f)$ and the standard deviation around the average. The viable step $\|\delta\mathbf{f}(t)\|$ associated with $\mathcal{J}$ is also plotted for comparison.

step $\Delta t = 0.01$,

$$
\begin{aligned}
\frac{2\vec{u}_{n,1} - 2\vec{u}_{n-1,4}}{\Delta t} &= -\mathbf{D}\left(\frac{\vec{u}_{n-1,4}^{\,2}}{2}\right) - \alpha_1\mathbf{D}^2\vec{u}_{n,1} - \alpha_2\mathbf{D}^4\vec{u}_{n,1} + \vec{W}_\Gamma\vec{f}_{n,1} \\
\frac{2\vec{u}_{n,2} - 2\vec{u}_{n-1,4}}{\Delta t} &= -\mathbf{D}\left(\frac{\vec{u}_{n,1}^{\,2}}{2}\right) - \alpha_1\mathbf{D}^2\vec{u}_{n,2} - \alpha_2\mathbf{D}^4\vec{u}_{n,2} + \vec{W}_\Gamma\vec{f}_{n,2} \\
\frac{\vec{u}_{n,3} - \vec{u}_{n-1,4}}{\Delta t} &= -\mathbf{D}\left(\frac{\vec{u}_{n,2}^{\,2}}{2}\right) - \alpha_1\mathbf{D}^2\vec{u}_{n,3} - \alpha_2\mathbf{D}^4\vec{u}_{n,3} + \vec{W}_\Gamma\vec{f}_{n,3} \\
\frac{6\vec{u}_{n,4} - 2\vec{u}_{n,3} - 4\vec{u}_{n,2} - 2\vec{u}_{n,1} + 2\vec{u}_{n-1,4}}{\Delta t} &= -\mathbf{D}\left(\frac{\vec{u}_{n,3}^{\,2}}{2}\right) - \alpha_1\mathbf{D}^2\vec{u}_{n,4} - \alpha_2\mathbf{D}^4\vec{u}_{n,4} + \vec{W}_\Gamma\vec{f}_{n,4},
\end{aligned}
\tag{13.4}
$$

where arrows $\vec{(\cdot)}$ denote spatially-discretized vectors in $\mathbb{R}^{N_g}$, and the subscript $(\cdot)_{n,s}$ indicates the s-th substep of n-th time step. In (13.4), only the nonlinear term from (13.1) is explicit. This leads to 4th-order accuracy in explicit time-integration and 1st-order in implicit time-integration. Independence of results on Δt is confirmed. The discrete-exact adjoint method is used to compute $\nabla_{\boldsymbol{\Theta}}\mathcal{J}$.

13.1.3 Quantification of chaos

Chaos of the K–S system is quantified with τ_λ and τ_ϕ as described in Section 12.3. To converge statistics, $N = 100$ adjoint final states $\mathbf{q}^\dagger(t_f)$ are taken to have pseudo-random values on mesh points, with the uniform random distribution $\mathcal{U}[-10^{-5}, 10^{-5}]$. For each $\mathbf{q}^\dagger(t_f)$, $\mathbf{q}^\dagger(t)$ is computed through (12.19) for $t_f - t_i = 5$. Initial time for each interval is $t_i = 5n$ with $n = 1, 2, \ldots, N$. Their ensemble-average is shown in Figure 13.2 (a). The e-folding time (12.28) of the K–S equation is estimated based on a linear fit to be

$\tau_\lambda \approx 0.412$. The viable step and its decay scale is calculated as for the Lorenz example in Section 12.4. 10 $\mathbf{q}^\dagger(t_f)$ are chosen from the sample for τ_λ. The corresponding viable step $\delta\Theta(t_i, t_f)$ is estimated via (12.30) and shown in Figure 13.2 (b). Its decay time scale is estimated per (12.36) to be $\tau_\phi \approx 0.339$.

13.1.4 Control

The objective functional is

$$\mathcal{J} = \frac{1}{t_f - t_i} \int_{t_i}^{t_f} \mathcal{I}(t)\, dt, \tag{13.5a}$$

where $t_i = 0$, $t_f = 5$, and $\mathcal{I}(t)$ is

$$\mathcal{I} = \int_0^{2\pi} \frac{1}{2} |u(x,t) - U_c|^2 W_\Omega(x)\, dx, \tag{13.5b}$$

with $W_\Omega(x)$

$$W_\Omega(x) = \begin{cases} 0.5 + 0.5 \sin\left(\pi \frac{x - 0.4L}{0.1L}\right) & x \in [0.35L, 0.45L] \\ 1 & x \in [0.45L, 0.55L] \\ 0.5 - 0.5 \sin\left(\pi \frac{x - 0.6L}{0.1L}\right) & x \in [0.55L, 0.65L] \\ 0 & \text{otherwise,} \end{cases} \tag{13.6}$$

shown in Figure 13.3 along with

$$W_\Gamma(x) = \begin{cases} 0.5 - 0.5 \sin\left(\pi \frac{x - 0.2L}{0.1L}\right) & x \in [0.15L, 0.25L] \\ 0 & x \in [0.25L, 0.75L] \\ 0.5 + 0.5 \sin\left(\pi \frac{x - 0.8L}{0.1L}\right) & x \in [0.75L, 0.85L] \\ 1 & \text{otherwise.} \end{cases} \tag{13.7}$$

These forms were selected to be both $\mathcal{C}^3$-continuous and exactly compact.

This optimization problem is defined to be similar to the Lorenz example, in that an effective control is anticipated to exist, yet finding it would be challenging due to chaos. This is done so by considering an estimate of the time for control effects to reach the target region,

$$\tau_A \approx \frac{\min \|\mathbf{x}_\Omega - \mathbf{x}_\Gamma\|}{U_c} \approx 0.25, \tag{13.8}$$

where $\min \|\mathbf{x}_\Omega - \mathbf{x}_\Gamma\| = 0.4\pi$ is the minimum distance between the target region Ω and the control region Γ,

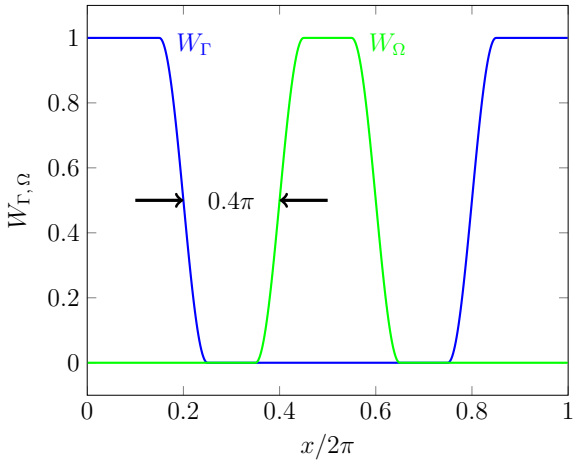


Figure 13.3: Mollifying supports for the control region and target region.

and $U_c = 5$ is the characteristic advection or wave propagation speed. This 'control propagation' time is set shorter than e-folding time $\tau_\lambda \approx 0.412$, so the control efforts can reach the target region before significantly affected by the nonlinear chaotic dynamics. While this does not guarantee the controllability as we could for the Lorenz example (Appendix C), evidences from experiments and simulations empirically support that the controllability of a flow system is limited by long τ_A. Kim and Bewley [61] illustrated based on flow instability modes that even laminar flows can be nearly uncontrollable far away from wall-mounted actuators. It is also well understood that two-point correlation in turbulence decays both space and time [79, 147], suggesting that the flow far away from the actuator is likely to be independent of the control. By having $\tau_A < \tau_\lambda$ we anticipate to avoid, though not completely, such limitation in controllability.

However, chaos will still present a challenge since the simulation time $t_f - t_i = 5$ is much longer than $\tau_\lambda \approx 0.412$. Figure 13.2 (a) shows that $\frac{\partial \mathcal{J}}{\partial f(t)}$ grows at the time scale of τ_λ, up to a factor of $\exp[(t_f - t_i)/\tau_\lambda] \approx 10^3$ during this optimization time period. Its associated viable step $\|\delta \mathbf{f}(t)\|$ per (12.30) also decays with the time scale τ_ϕ. In Figure 13.2 (b), $\|\delta \mathbf{f}(t)\|$ matches $\delta\Theta$ in both magnitude and decaying time scale τ_ϕ. Hence, it is anticipated from these time scales that the optimization will be impacted significantly by chaos, despite the potential for an effective control.

13.1.5 Optimization result

Figure 13.4 (a) shows that standard gradient-based optimization reduces $\mathcal{J}$ by 22.4%. The step sizes in Figure 13.4 (b) shows that most steps do not significantly reduce $\mathcal{J}$. Steps are also small with $\delta\Theta \lesssim 10^{-5}$, which is much smaller than the overall right-hand side of the equation, which yield $\|\frac{\partial u}{\partial t}\| \approx 1$ at its minimum and typically $\approx 10^2$. This suggests that the optimization stalls in narrow features of $\mathcal{J}$ optimization space.

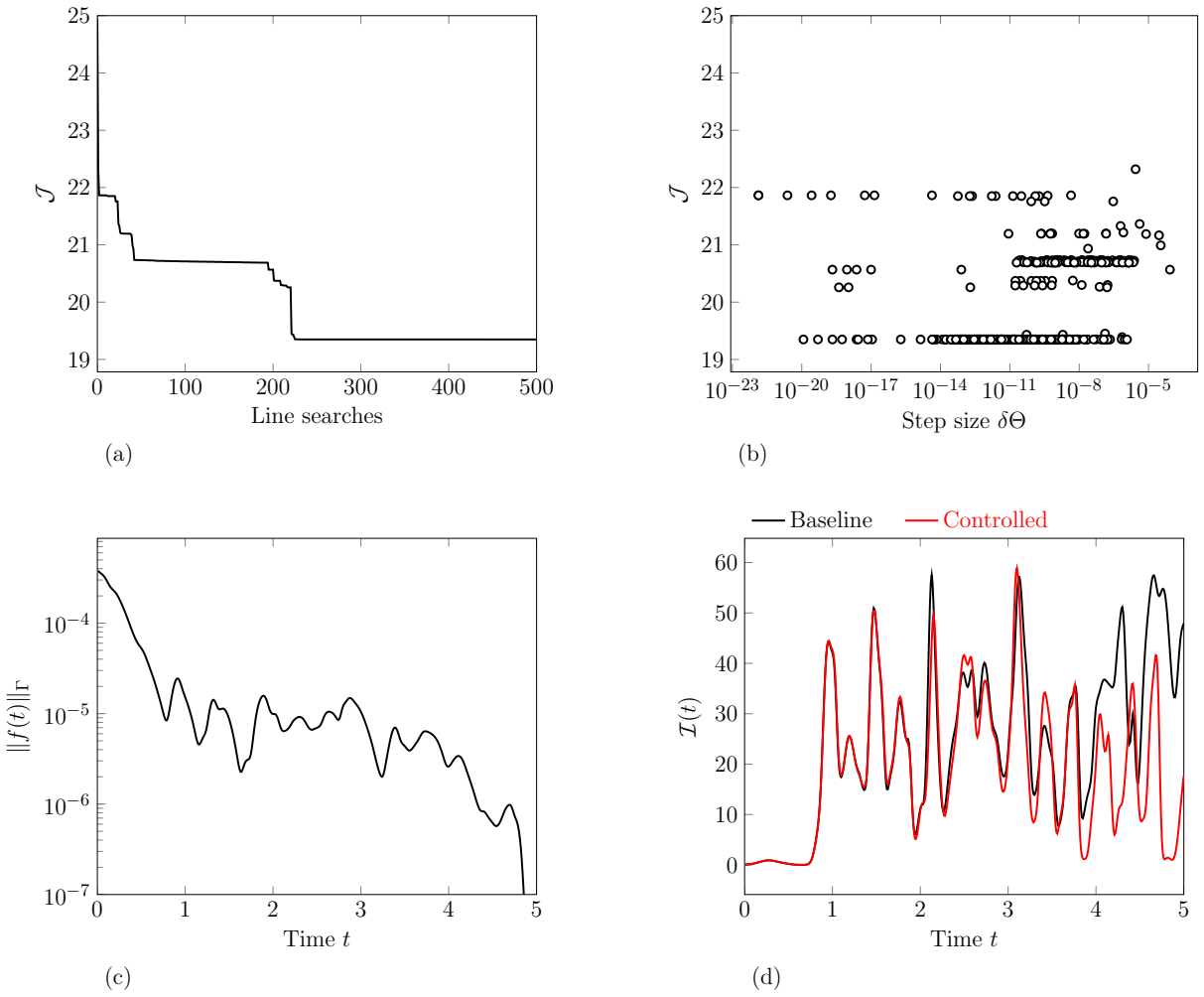


Figure 13.4: Results for the K–S configuration: (a) reduction of $\mathcal{J}$ (13.5) using standard gradient-based optimization; (b) step sizes taken in the optimization; (c) the control strength for the optimized control; and (d) the instantaneous functional $\mathcal{I}$ (13.5b) of the controlled solution compared the baseline solution.

Occasionally $\mathcal{J}$ is reduced by a few line search steps, implying that the optimization escapes ridge-like features. However, even in these cases, steps are still limited at $\delta\Theta \approx 10^{-5}$, consistent in Figure 13.2 (b). Even if the optimization happens to circumvent a ridge of such narrow feature, it encounters another similar restriction. Figure 13.4 (c) shows that the control is biased to early times and with a limited amplitude $\|f\|_\Gamma \approx 10^{-4}$, as a result of these small steps. While it is impressive that such a small control can achieve 22% reduction, there is no reason to pursue such control, since the control amplitude is not penalized. Figure 13.4 (d) confirms that the control effort is focused on late times, with most of the simulation time unaffected. Overall, the behavior is similar to the Lorenz system we analyzed in detail, with qualitatively similar optimization challenges.

13.2 Two-dimensional Kolmogorov flow

13.2.1 Configuration

We consider two-dimensional compressible flow with a similarly challenging optimal control problem. A sinusoidal body force is added to the flow equation (11.41),

$$\frac{\partial \mathbf{q}}{\partial t} - \mathcal{R}[\mathbf{q}, \mathbf{f}] = \begin{pmatrix} 0 & f_K(\mathbf{x}) & 0 & 0 & 0 \end{pmatrix}^T, \tag{13.9}$$

with x_1-direction forcing f_K

$$f_K(\mathbf{x}, t) = \chi \sin\left(2\pi n \frac{x_2}{L_t}\right), \tag{13.10}$$

on the two-dimensional periodic domain $(x_1, x_2) \in [0, L_x] \times [0, L_t]$. For simplicity, we assume an energy sink that exactly cancels out the work by the body force to preserve stationary temperature.

The domain and the external forcing parameters in (13.10) are chosen to be close to the values that are reported to exhibit strong chaos [123–125]. The Reynolds number based on the forcing strength χ is

$$\mathrm{Re}_\chi \equiv \frac{\sqrt{\chi}}{\mu_0}\left(\frac{L_t}{2\pi}\right)^{\frac{3}{2}} \approx 284, \tag{13.11}$$

with dynamic viscosity μ_0 constant. This is higher than the reported values (at largest $\mathrm{Re}_\chi = 200$), at which a chaotic flow state becomes the global attractor (so any initial flow leads to a chaotic state) [123–125]. The aspect ratio $\alpha \equiv \frac{L_t}{L_x} \approx 1.11$ is chosen close to the $\alpha = 1.1$ value at which intensely chaotic oscillations of vortices are observed by Lucas and Kerswell [125]. Lastly, the widely-used wavenumber $n = 4$ is used in (13.10) for the body force [123–125]. Time and velocity scales are characterized by L_t and χ with constant

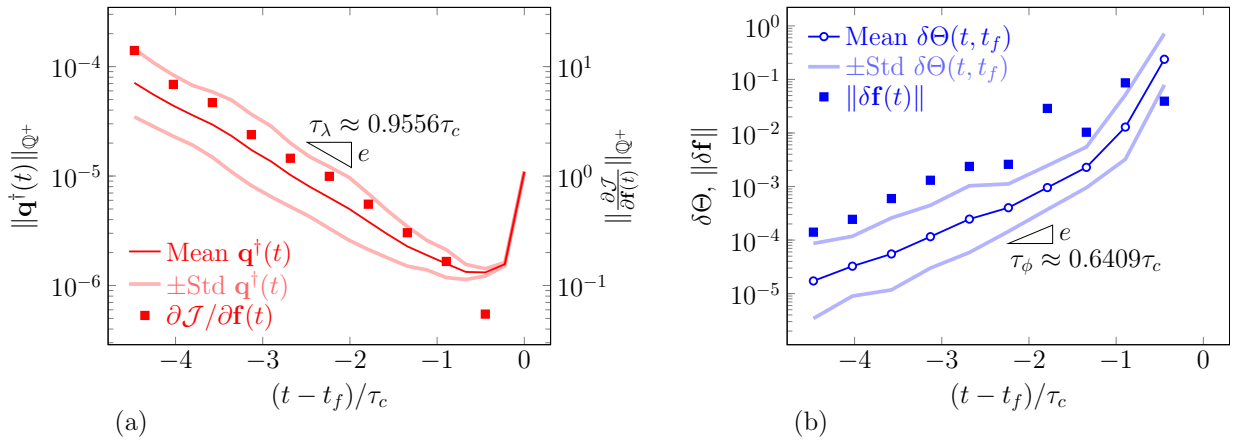


Figure 13.5: Two-dimensional Kolmogorov flow: (a) The ensemble average of the adjoint $\mathbf{q}^\dagger(t)$ and the standard deviation around the average. The gradient of $\mathcal{J}$ (13.14) with respect to control forcing $f(t)$ is also plotted for comparison. (b) The ensemble average of the viable step $\delta\Theta(t, t_f)$ and the standard deviation around the average. The viable step $\|\delta\mathbf{f}(t)\|$ associated with $\mathcal{J}$ is also plotted for comparison.

density ρ_0 at ambient temperature,

$$\tau_c = \sqrt{\frac{\rho_0 L_t}{\chi}} \qquad\qquad u_c = \sqrt{\frac{L_t \chi}{\rho_0}}. \tag{13.12}$$

The initial condition is,

$$u_1(\mathbf{x}) = U_0 + \Delta U \cos\left[2\pi\left(\frac{x_1}{L_x} - 0.1\sin\frac{2\pi x_2}{L_t}\right)\right], \tag{13.13}$$

with $U_0 = \sqrt{5}u_c$, $\Delta U = \frac{\sqrt{5}}{10}u_c$, and $u_2(\mathbf{x}) = u_3(\mathbf{x}) = 0$.

Using the SBP 3-6 scheme of Section 11.4.2, the domain is discretized with 256^2 uniform mesh points. Mild artificial dissipation is included per (11.42) with $\sigma_{\text{diss}} = 0.005$. A standard explicit RK4 scheme is used for time integration with time step $\Delta t = 2.236 \times 10^{-4}\tau_c$. The dual-consistent, discrete-exact adjoint solver developed by Vishnampet [132] is used to compute gradients. Results are confirmed to be independent of resolution.

13.2.2 Quantification of chaos

To calculate τ_λ and τ_ϕ, $N = 10$ adjoint final states $\mathbf{q}^\dagger(t_f)$ are taken to have random values for all state variables at mesh points, with the uniform random distribution $\mathcal{U}[-10^{-5}, 10^{-5}]$. Sample times t_f are chosen to be $t_f = (44.72+4.47n)\tau_c$ for $n = 1, 2, \ldots, N$, and each $\mathbf{q}^\dagger(t_f)$ is evolved for $t_f - t_i = 4.47\tau_c$. Figure 13.5 (a) shows the ensemble average of the adjoint states in time, which grows exponentially in reverse time for

$t - t_f \lesssim -\tau_c$. The e-folding time based on this is $\tau_\lambda \approx 0.9556\tau_c$ and viable step decay time is $\tau_\phi \approx 0.6409\tau_c$ as shown in Figure 13.5 (b), which is somewhat faster than τ_λ.

13.2.3 Control formulation

The objective functional is

$$\mathcal{J} = \int_{t_i}^{t_f} \mathcal{I}(t)\, dt, \tag{13.14a}$$

where $t_i = 44.72\tau_c$, $t_f = 49.19\tau_c$, and the instantaneous functional $\mathcal{I}(t)$ is,

$$\mathcal{I}(t) = \int_\Omega (p - p_0)^2 W_\Omega(\mathbf{x})\, d^2\mathbf{x}, \tag{13.14b}$$

with constant ambient pressure $p_0 = 14.286 L_t \chi$ and

$$W_\Omega = 0.5\left\{\tanh[4(x_1^\Omega - 0.075)] - \tanh[4(x_1^\Omega - 0.925)]\right\}, \tag{13.15a}$$

where

$$x_1^\Omega = \begin{cases} \frac{1}{0.4}\left(\frac{x_1}{L_x} - 0.6\right) & \left|\frac{x_1}{L_x} - 0.8\right| \le 0.2 \\ 0 & \text{otherwise.} \end{cases}$$

For the target region

$$W_\Gamma = 0.5\left\{\tanh[4(x_1^\Gamma - 0.075)] - \tanh[4(x_1^\Gamma - 0.925)]\right\}, \tag{13.15b}$$

where

$$x_1^\Gamma = \begin{cases} \frac{1}{0.4}\left(\frac{x_1}{L_x} - 0.1\right) & \left|\frac{x_1}{L_x} - 0.3\right| \le 0.2 \\ 0 & \text{otherwise.} \end{cases}$$

For this control problem we grant controllability on full-state (ρ, $\rho\mathbf{u}$, ρE) at the discrete points in the control region. Figure 13.6 shows an evolution of the initial condition with the control region and target region defined above.

As for the K–S equation, this control problem is set up so that an effective control is expected to exist. The minimum distance between half maximum of W_Ω and W_Γ is $\Delta x_{\Gamma-\Omega} = 0.198 L_t$, so the control propagation time (13.8) based on average advection speed $U_0 = \sqrt{5}u_c$ is $\tau_A = 0.0885\tau_c$, significantly shorter than e-folding time $\tau_\lambda \approx 0.9556\tau_c$. From this, as for K–S example, we anticipate that control efforts can reach the target region before being overwhelmed by chaos. On the other hand, the optimization is challenging for

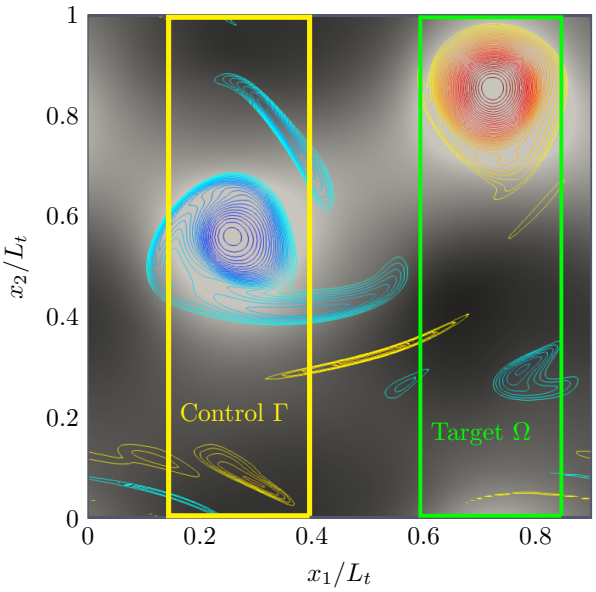


Figure 13.6: Two-dimensional Kolmogorov flow pressure $p/p_0 \in [0.9, 1.1]$ (grayscale) and vorticity $\omega\tau_c \in [-44.72, 44.72]$ (contours) at $t = 45.62\tau_c$ with the control and target regions indicated.

the time $t_f - t_i = 4.472\tau_c$, for which a gradient is amplified by a factor of $\exp[(t_f - t_i)/\tau_\lambda] \approx 107.78$ and a viable step decays by $\exp[-(t_f - t_i)/\tau_\phi] \approx 10^{-3}$.

13.2.4 Optimization result

Figure 13.7 (a) shows that the standard gradient-based optimization achieved 43.8% $\mathcal{J}$-reduction. The optimization is not stalled so significantly as in Figure 13.4 (a), presumably because the Kolmogorov flow has longer τ_λ and τ_ϕ compared to $t_f - t_i$ than K–S system. However, the optimization is still limited by its chaotic dynamics. Figure 13.7 (b) shows that most line search steps are scattered with $\delta\Theta \approx 10^{-5}$, which is consistent with $\delta\Theta(t, t_f)$ at $t - t_f = -4.47\tau_c$ shown in Figure 13.5 (b). This implies that the optimization is impeded by similarly sized features in $\mathcal{J}[\boldsymbol{\Theta}]$. As for other systems, Figure 13.7 (c) and (d) show that the control is biased to short early times and mainly influences late times.

13.3 Advection + Kuramoto–Sivashinsky (Adv+KS) model

13.3.1 Governing equation

A model system is constructed to illustrate how a relatively chaotic portion of a turbulent flow might impede optimization even for a case that only requires control of a relatively deterministic (presumably

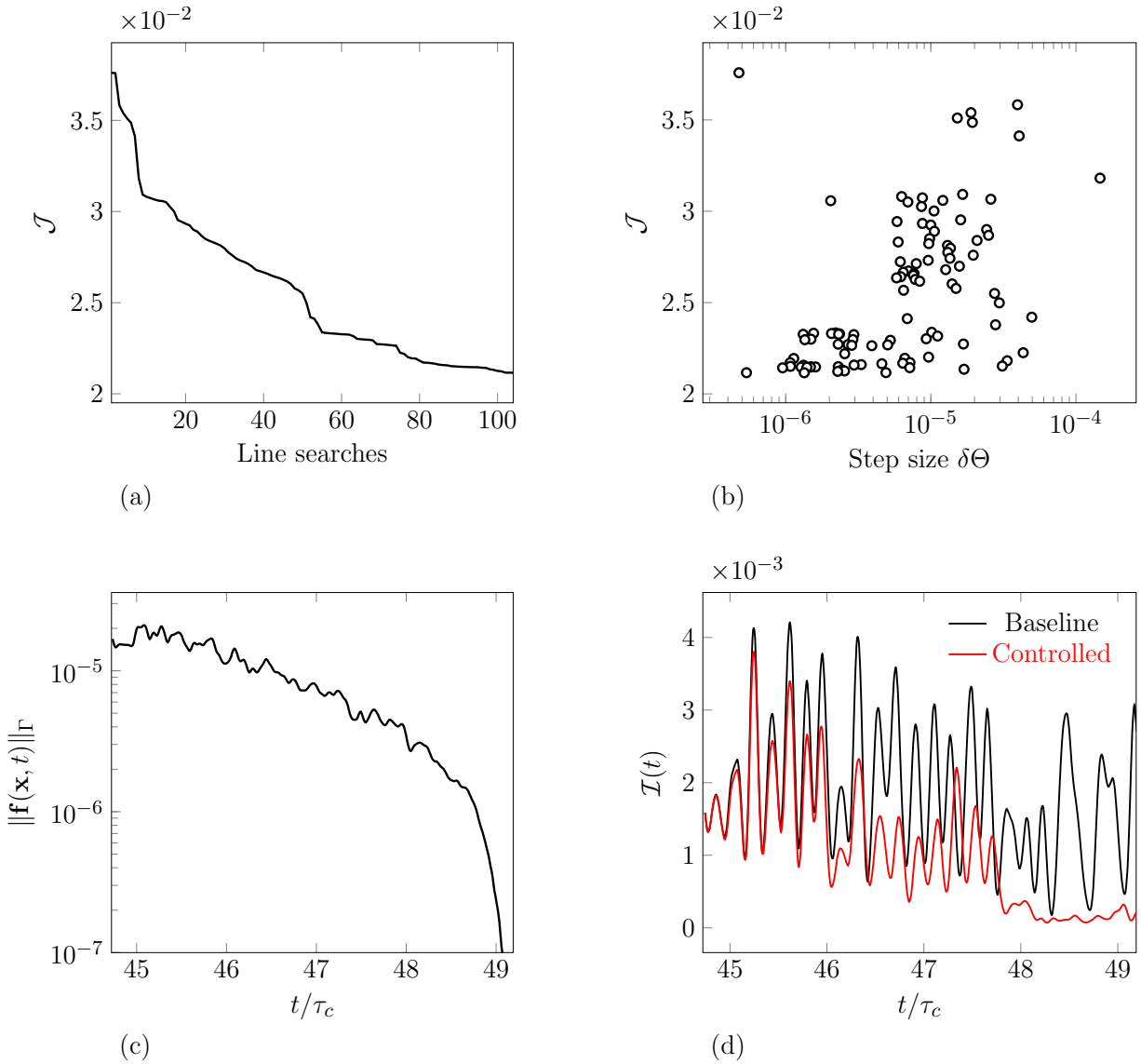


Figure 13.7: Two-dimensional Kolmogorov flow control: (a) reduction of $\mathcal{J}$ (13.14) over line searches, (b) step sizes taken in the optimization, (c) the control strength of the optimized control, and (d) the instantaneous functional $\mathcal{I}$ (13.14b) of the controlled solution compared the baseline solution.

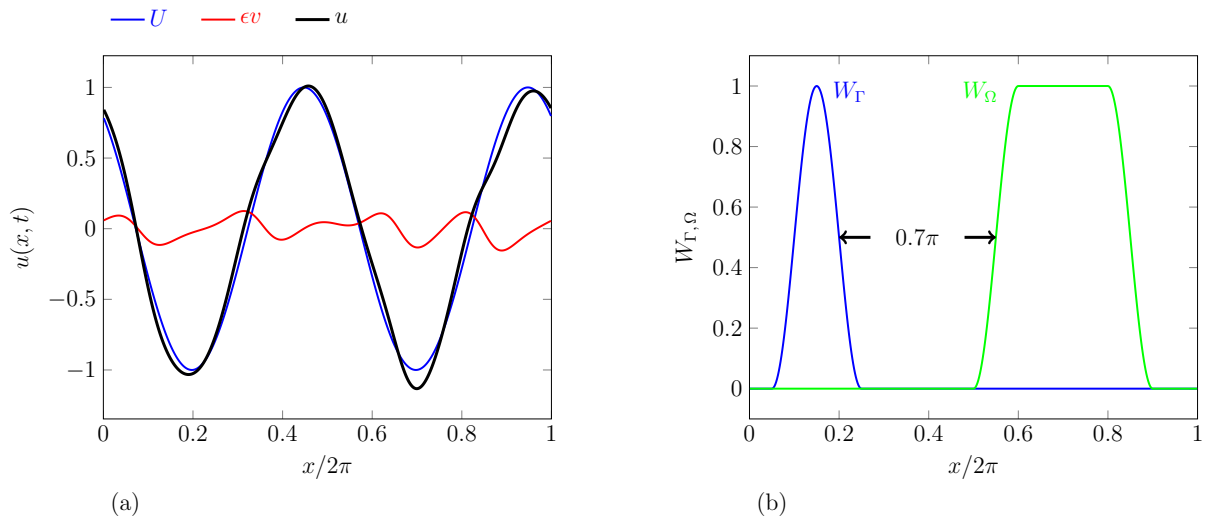


Figure 13.8: (a) The evolution of the wave $u = U + \epsilon v$. (b) Mollifying supports for the control region Γ and the target region Ω.

larger) component. A one-dimensional periodic solution $u(x,t)$ is taken to be composed of two parts:

$$u(x,t) = U(x,t) + \epsilon v(x,t), \tag{13.16}$$

where $\epsilon = 0.01$. These two parts phenomenologically represent a scale separation, and their independence is explicit for clarifying this demonstration. However, as for turbulence, we assume that we neither know their independent dynamics nor that we can infer a separation schema.

Each part is governed by a distinct equation. The deterministic large scales $U(x,t) = U(x - U_0 t, 0)$ are governed simply by

$$\frac{\partial U}{\partial t} + U_0 \frac{\partial U}{\partial x} = 0, \tag{13.17a}$$

so the initial wave simply advects with a constant U_0. The $v(x,t)$ is governed by the K–S equation (13.1),

$$\frac{\partial v}{\partial t} + \frac{1}{2}\frac{\partial v^2}{\partial x} + \alpha_1 \frac{\partial^2 v}{\partial x^2} + \alpha_2 \frac{\partial^4 v}{\partial x^4} = 0, \tag{13.17b}$$

with $u_0 = 2$, $\alpha_1 = 1$, and $\alpha_2 = 0.029910$, and so is chaotic. The initial conditions are

$$U(x,0) = \sin 2x \qquad \text{and} \qquad v(x,0) = \sin 5x. \tag{13.18}$$

Figure 13.8 (a) shows how U and v constitute the solution u.

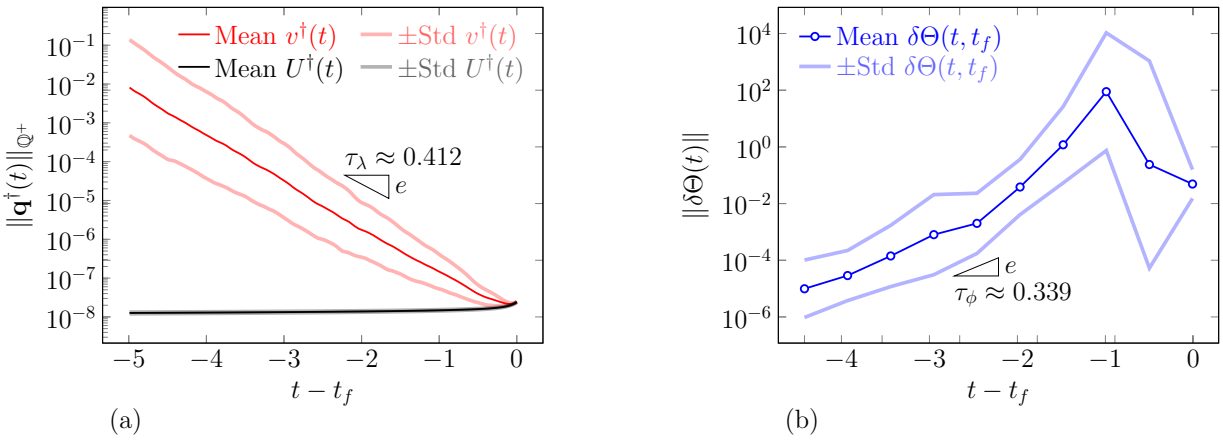


Figure 13.9: The Adv+KS model (13.17): (a) The ensemble average of the adjoint $v^\dagger(t)$ and the standard deviation around the average. For comparison, the adjoint $U^\dagger(t)$ for large wave (13.17a) is also plotted with its ensemble average and standard deviation around the average. (b) The ensemble average of the viable step $\delta\Theta(t, t_f)$ for v and the standard deviation around the average. For U, $\delta\Theta \to \infty$ due to its linear dynamics and thus it is not plotted.

13.3.2 Quantification of chaos

Due to the governing equation (13.17b), only v has the impact of chaos on optimization. We quantify this with τ_λ and τ_ϕ for U and v separately. To converge statistics, $N = 100$ adjoint final states $U^\dagger(t_f)$ and $v^\dagger(t_f)$ are respectively taken to have pseudo-random numbers at mesh points, with the uniform distribution $\mathcal{U}[-10^{-5}, 10^{-5}]$. The sample time intervals are chosen the same as for the K–S example in Section 13.1.3. Figures 13.9 (a) and (b) respectively show the adjoint state for v and its associated viable step, for which time scales are the same as the K–S example. For U, the adjoint state simply advects with small numerical dissipation as shown in Figures 13.9 (a), indicating that U is completely deterministic.

13.3.3 Optimization problem

The unified control objective is to suppress the wave in Ω,

$$\mathcal{J} = \int_{t_0}^{t_f} \int_0^L |u(x,t)|^2 W_\Omega(x,t)\, dxdt, \tag{13.19}$$

where

$$W_\Omega(x) = \begin{cases} 0.5 + 0.5\sin\left(\frac{x-1.1\pi}{0.2}\right) & \frac{x}{2\pi} \in [0.5, 0.6] \\ 1 & \frac{x}{2\pi} \in [0.6, 0.8] \\ 0.5 - 0.5\sin\left(\frac{x-1.7\pi}{0.2}\right) & \frac{x}{2\pi} \in [0.8, 0.9] \\ 0 & \text{otherwise,} \end{cases} \tag{13.20}$$

is $\mathcal{C}^3$-continuous and exactly compact (Figure 13.8 b). The scale factor $\epsilon = 0.01$ in (13.16) causes most contribution to $\mathcal{J}$ to come from U.

We force the entire u in the Γ control region, so the full-dynamics (13.17) has a nominally combined form with a forcing term,

$$\frac{\partial u}{\partial t} + \mathcal{R}[u] = (1+\epsilon) W_\Gamma(x) f(x,t), \tag{13.21}$$

where the left-hand side implicitly represents (13.17). For this demonstration, we assume the forcing is distributed to both U and v in proportion to their amplitude factor 1 and ϵ, so

$$\frac{\partial U}{\partial t} + U_0 \frac{\partial U}{\partial x} = W_\Gamma(x) f(x,t) \tag{13.22a}$$

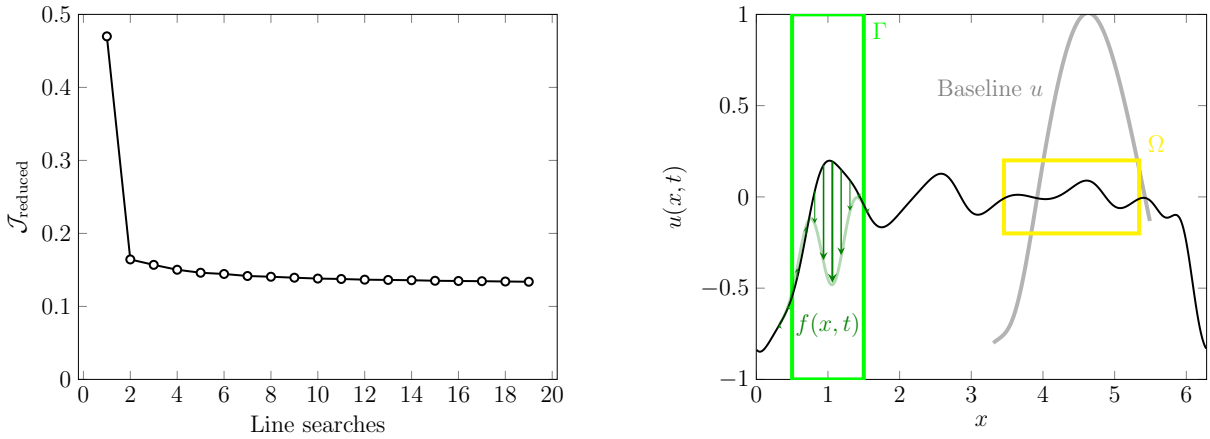
$$\frac{\partial v}{\partial t} + \frac{1}{2}\frac{\partial}{\partial x}\left(v^2\right) + \alpha_1 \frac{\partial^2 v}{\partial x^2} + \alpha_2 \frac{\partial^4 v}{\partial x^4} = W_\Gamma(x) f(x,t), \tag{13.22b}$$

where the control mollifying support $W_\Gamma(x)$ is shown in Figure 13.8 (b),

$$W_\Gamma(x) = \begin{cases} 0.5 + 0.5\sin\left(\frac{x-0.2\pi}{0.2}\right) & \frac{x}{2\pi} \in [0.05, 0.25] \\ 0 & \text{otherwise.} \end{cases} \tag{13.23}$$

The control propagation time τ_A (13.8) for U is $\tau_A = 0.175$, and due to the simple characteristics of (13.22a) its controllability is guaranteed. However, the characteristic advection speed of v is zero due to its zero mean value, hence we might say $\tau_A \to \infty$. So there is no direct path by which control effort in v can propagate to the target region, and most of control effort is continually impeded by chaotic dynamics of (13.22b). In this situation, as discussed in Section 10.3, if our knowledge and technologies permit, it is reasonable to extract the large-U dynamics (13.22a) to construct a reduced model with it.

We first consider such an ideal case, utilizing the explicit separability of $u = U + \epsilon v$. The optimization



(a) (b)

Figure 13.10: The optimization result from the model-reduction approach. (a) Minimization of the reduced-order model objective functional $\mathcal{J}_{\text{reduced}}$. (b) The optimized control from the reduced model is successfully applied to the full $U + \epsilon v$ dynamics (13.22). Green arrows indicate the control forcing $f(x,t)$ (scaled with a factor of 0.3).

can be pursued with a corresponding reduced objective from (13.19),

$$\mathcal{J}_{\text{reduced}} = \int_{t_0}^{t_f} \int_0^L |U(x,t)|^2 W_\Omega(x,t)\, dxdt. \tag{13.24}$$

where the U dynamics (13.22a) is recast as reduced model

$$\frac{\partial U}{\partial t} + U_0 \frac{\partial U}{\partial x} = W_\Gamma(x) f(x,t). \tag{13.25}$$

With $\mathcal{J}_{\text{reduced}}$ and (13.25), standard gradient-based optimization identifies an effective control with a few line searches, as shown in Figure 13.10 (a). For this demonstration, obviously the reduced model represents the full dynamics accurately, so the control found from the reduced model is similarly effective for the full dynamics (13.22), as shown in Figure 13.10 (b). When applied to the full dynamics, the control reduces $\mathcal{J}$ slightly more than the model $\mathcal{J}_{\text{reduced}}$ ($\mathcal{J}_{\text{reduced}} = 0.1338$ to $\mathcal{J} = 0.1314$).

Of course, extracting the analogous U-dynamics for turbulent flows is challenging, so optimization is exposed to the full dynamics (13.22). However, this is problematic. In this example, once v is included, the optimization with the standard gradient method fails, as the useful gradient associated with the large-scale U is masked by the exploding gradient of the v component (Figure 13.11 a). Figure 13.11 (b) further shows that the obtained control is limited in its magnitude compared to the one from the reduced-order model. Figure 13.11 (c) and (d) suggest that this optimization is also trapped.

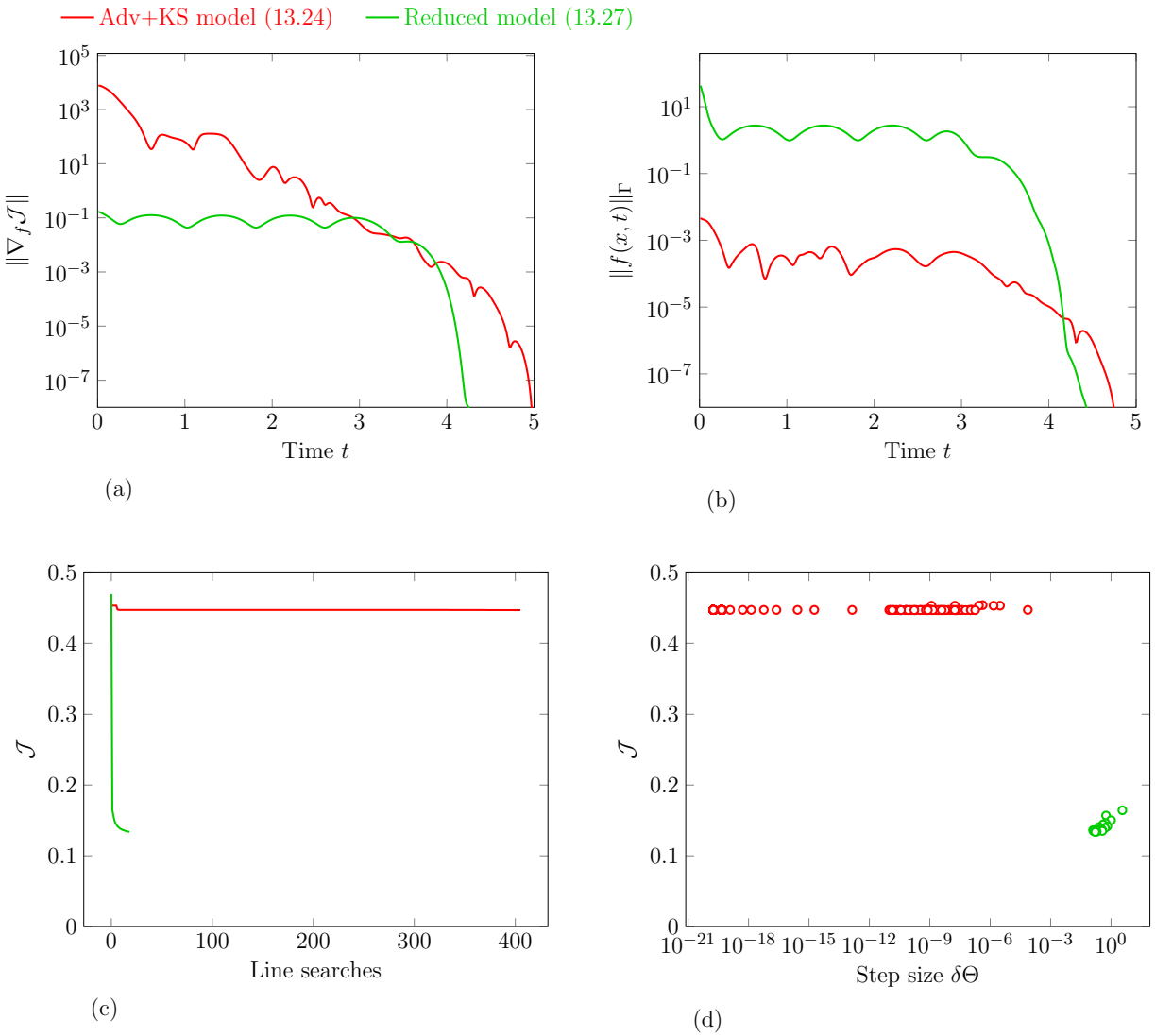
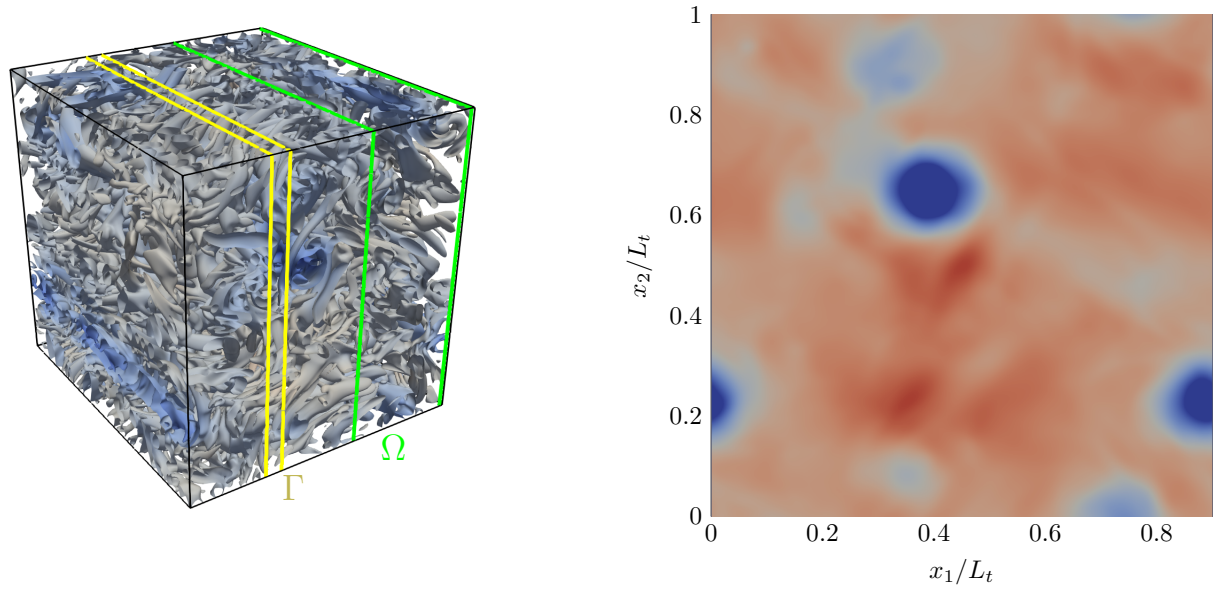


Figure 13.11: The Adv+KS model: (a) gradient of $\mathcal{J}$ (13.19) to $f(x,t)$, from the full dynamics and the reduced-order model; (b) the control strength of the optimized controls from the full dynamics and the reduced model; (c) $\mathcal{J}$ minimization using standard adjoint method, which is applied to the full dynamics and the reduced-order model; Since an effective solution is already found for the reduced-order model, further line searches are not needed and the optimization is stopped; and (d) The step sizes taken in optimization with the full dynamics and the reduced-order model



(a) (b)

Figure 13.12: The baseline simulation of the three-dimensional Kolmogorov flow. (a) Isosurfaces of Q-criterion ($Q = 20\tau_c^{-2}$) colored by the pressure $p/p_0 \in [0.95, 1.05]$ at $t = 13.42\tau_c$. (b) Pressure $p/p_0 \in [0.973, 1.008]$ averaged along x_3 direction.

13.4 Three-dimensional Kolmogorov flow

Three-dimensional Kolmogorov flow is turbulent. The chaotic small scales are expected to shorten the time scales τ_λ and τ_ϕ.

13.4.1 Configuration and discretization

The compressible flow equations are solved on the triply periodic domain $(x_1, x_2, x_3) \in [0, L_x] \times [0, L_t] \times [0, L_t]$. The same aspect ratio $\alpha = \frac{L_t}{L_x} \approx 1.11$ and body force wave number $n = 4$ are used with the same non-dimensional parameters (13.11) as for the two-dimensional case. The initial condition is

$$u_1(\mathbf{x}) = U_0 \tag{13.26a}$$

$$u_2(\mathbf{x}) = \Delta U \cos\left[2\pi\left(\frac{x_1}{L_x} - 0.1\sin\frac{2\pi x_2}{L_t} - 0.1\sin\frac{2\pi x_3}{L_t}\right)\right], \tag{13.26b}$$

with $U_0 = \sqrt{5}u_c$, $\Delta U = \frac{\sqrt{5}}{50}u_c$, and $u_3(\mathbf{x}) = 0$.

The domain is discretized with 256^3 uniform mesh points, and the time step is $\Delta t = 2.236 \times 10^{-4}\tau_c$. The same SBP 3-6 scheme from Section 13.2 is used in all three directions.

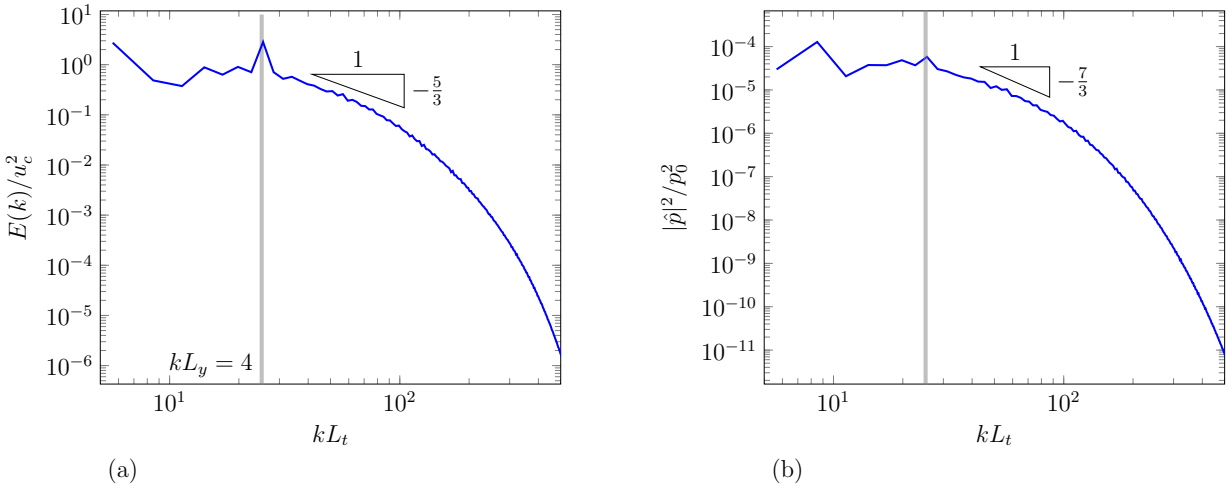


Figure 13.13: Spectra of the three-dimensional Kolmogorov flow simulation for (a) turbulence energy, and (b) pressure fluctuation. Gray line indicates the external forcing wavenumber.

Figure 13.12 (a) visualizes the flow, showing the presence of small-scale turbulence. At the same time its x_3-averaged pressure, shown in Figure 13.12 (b), has a structure similar to the two-dimensional counterpart (Figure 13.6). This two-dimensional structure is expected in this flow [148], and loosely corresponds the class of potential control targets discussed in Section 10.3 in which a deterministic component underlies more chaotic turbulence. Turbulence statistics are collected at 200 intervals during $t/\tau_c \in [13.42, 17.89]$. Energy spectra in Figure 13.13 shows that the turbulence is broadbanded. However, unlike the Adv+KS example, no prior knowledge is given concerning the dynamics between the larger—seemingly more deterministic—structure and smaller—seemingly more chaotic—scales of turbulence. In general, a reduced model to accurately separate the dynamics is not available.

13.4.2 Quantification of chaos

The e-folding time (12.28) is computed as two-dimensions. Figure 13.14 (a) shows the ensemble average of 10 adjoint state samples in time, which starts to grow in reverse time for $t - t_f \lesssim -0.22\tau_c$. The e-folding time based on this growth rate is $\tau_\lambda \approx 0.49\tau_c$. Figure 13.14 (b) shows the decay time scale $\tau_\phi \approx 0.30\tau_c$ of the corresponding viable steps. These two time scales are shorter than the two-dimensional case, presumably due to turbulence.

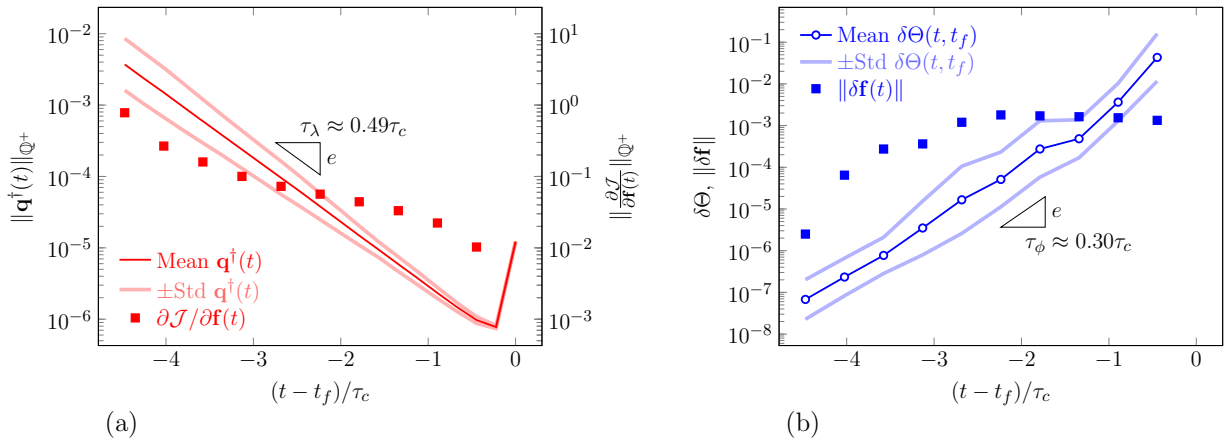


Figure 13.14: The three-dimensional Kolmogorov flow: (a) The ensemble average of the adjoint $\mathbf{q}^\dagger(t)$ and the standard deviation around the average. The gradient of $\mathcal{J}$ (13.14) with respect to control forcing $f(t)$ is plotted together for comparison. (b) The ensemble average of the viable step $\delta\Theta(t, t_f)$ and the standard deviation around the average. The inferred viable step $\|\delta\mathbf{f}(t)\|$ (12.30) associated with $\mathcal{J}$ is also plotted for comparison.

13.4.3 Control problem

The same objective functional (13.14) as for two dimensions is targeted between $t_i = 13.42\tau_c$ and $t_f = 17.89\tau_c$, with the same spatial support W_Ω in (13.15b). The control region is taken smaller than the two-dimensional case with

$$W_\Gamma \sim B_{0,3}(4x_1^\Gamma - 2), \tag{13.27}$$

where $B_{0,3}(x)$ is the cubic B-spline basis function, and

$$x_1^\Gamma = \frac{1}{0.1L_x}(x_1 - 0.25L_x).$$

The support is normalized so that $\max W_\Gamma = 1$. The controller actuates thermal energy ρE in this region. Figure 13.12 (a) visualized the flow.

As for the previous examples, the configuration is set up so the control propagation time τ_A is shorter than $\tau_\lambda \approx 0.49\tau_c$. The τ_A we assume is again based on the minimum distance $\Delta x_{\Gamma-\Omega} \lesssim 0.279L_t$ between half maximum of W_Ω and W_Γ, so with average advection speed $U_0 = \sqrt{5}u_c$, $\tau_A = 0.125\tau_c$. The simulation time $t_f - t_i = 4.47\tau_c$ is the same as the two-dimensional case. The gradient and associated viable step for the actual $\mathcal{J}$ (13.14) are shown in Figure 13.14. Though the gradient seem to be not affected by the chaos for $t - t_f \gtrsim -2.24\tau_c$, it eventually explodes with the time scale of τ_λ. Similarly, in Figure 13.14 (b) its viable step also starts to decay with the time scale close to τ_ϕ for $t - t_f < -2.24\tau_c$.

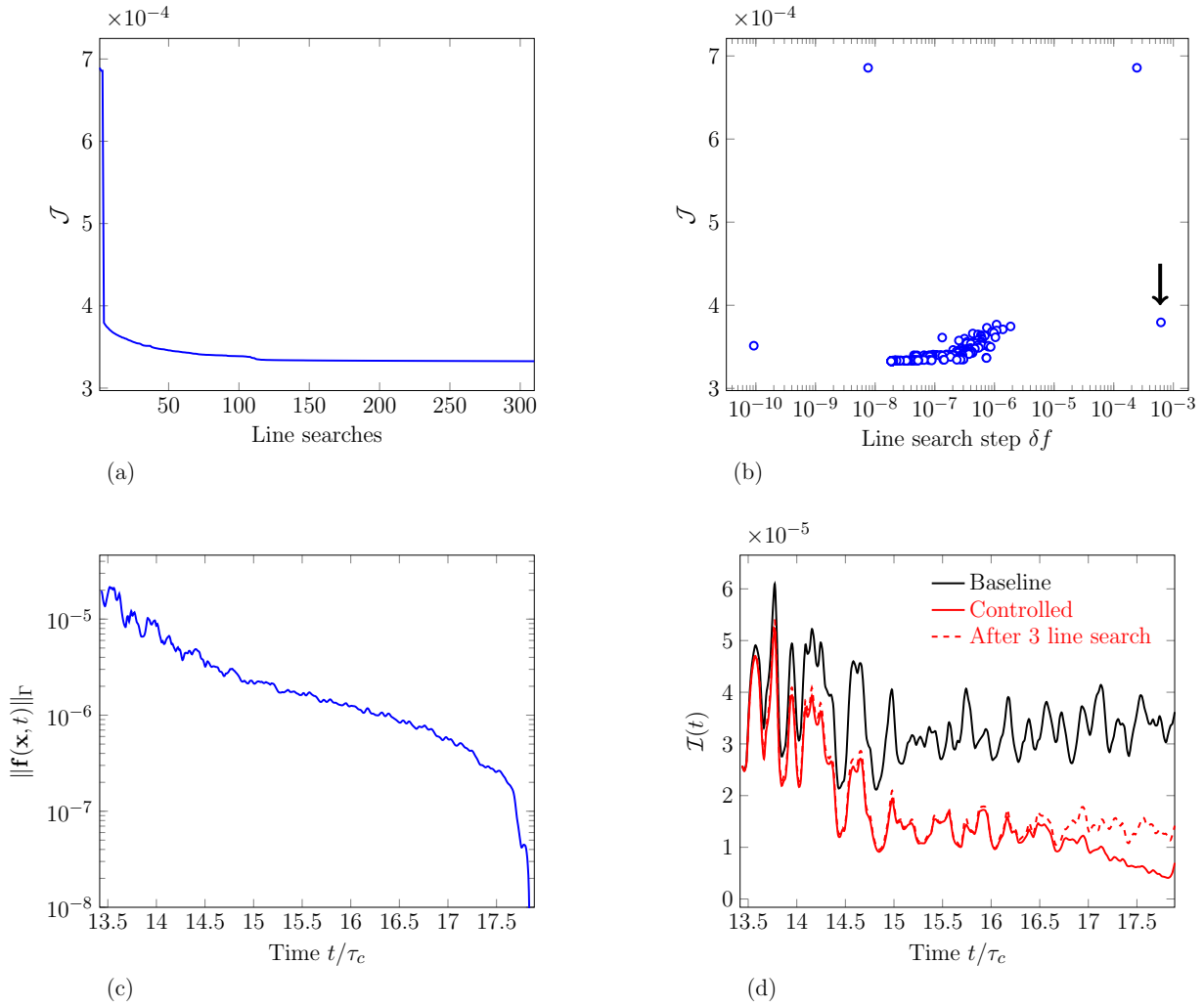


Figure 13.15: The three-dimensional Kolmogorov flow control: (a) reduction of $\mathcal{J}$ (13.14), (b) step sizes taken in the optimization, (c) the control strength of the optimized control, and (d) the instantaneous functional $\mathcal{I}$ (13.14b) of the controlled solution compared the baseline solution.

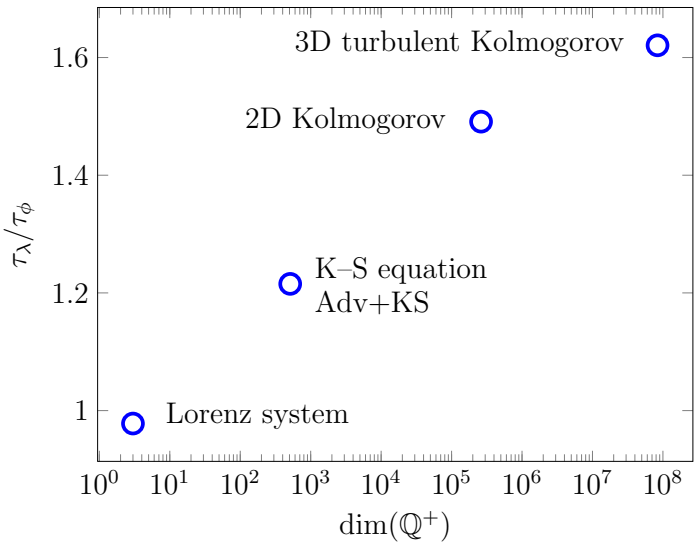


Figure 13.16: The ratio between e-folding time τ_λ and the decay time scale of viable step τ_ϕ for the flow systems in this study, plotted with the dimension of discretized state.

13.4.4 Optimization result

The optimization of turbulent Kolmogorov flow is also impacted by the chaotic dynamics as other flows. Figure 13.15 (a) shows that the standard adjoint method achieved 51.5% reduction of $\mathcal{J}$. Though perhaps significant, most of the reduction is in one single step of the third line search. Figure 13.15 (b) shows the step sizes taken in the optimization, where most of them are consistent with the smallest $\delta\Theta \sim 10^{-7}$ in Figure 13.14 (b), except the third line search. While this seems to be a fortunate escape of ridges in $\mathcal{J}[\boldsymbol{\Theta}]$ space, the control pursued by the optimization is still biased, distributed exponentially in time, as shown in Figure 13.15 (c). Figure 13.15 (d) shows the reduced instantaneous objective functional, where most of the reduction is achieved in the third line search. Further reduction after the third search is still biased to late times.

13.5 Summary on the quantification of chaos time scales

We summarize the τ_λ and τ_ϕ time scales for all systems considered. Figure 13.16 shows τ_λ/τ_ϕ. While the Lorenz system has $\tau_\lambda \approx \tau_\phi$, all the other systems have $\tau_\phi < \tau_\lambda$. This suggests that for these systems non-convexity is the greater concern than the gradient growth.

Chapter 14

Multi-step penalty method

In this chapter, we propose an optimization framework that can skirt the non-convexity of $\mathcal{J}$ described in Chapter 12. It is based on the framework for the equality-constrained optimization introduced in Chapter 11, which includes many variants [57], from which we focus on two: the quadratic penalty method and the augmented Lagrangian method [57, 58]. To motivate our formulation, the new framework is developed in Sections 14.1 through 14.3 and applied to the logistics map example of Section 12.2.2, showing its effectiveness in face of horseshoe mapping. It is then demonstrated for the Lorenz example in Section 14.4. Chapter 15 will cover its application to the increasingly challenging applications of Chapter 13.

14.1 Equality-constrained optimization with a penalty method

We recall the generic equality-constrained optimization problem (11.3)

$$\text{minimize } \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] \in \mathbb{R} \qquad \text{such that} \quad \mathcal{N}[\mathbf{q}; \boldsymbol{\Theta}] = \mathbf{0}.$$

While the gradient-based optimization framework in Chapter 11 strictly enforces the equality, some optimization methods pursue a local solution to (11.3) via a sequence of unconstrained optimizations. In these, the state $\mathbf{q}$ is not strictly constrained by $\mathcal{N} = 0$, but the violation is penalized with increasing strength [57, 58]. The penalization is imposed by augmenting $\mathcal{J}$ in each subproblem

$$\text{minimize } \mathcal{J}_A[\mathbf{q}, \boldsymbol{\Theta}, \mu] \equiv \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] + \mathcal{P}[\mathcal{N}[\mathbf{q}, \boldsymbol{\Theta}], \mu] \in \mathbb{R}, \tag{14.1}$$

where the functional $\mathcal{P}$ penalizes the violation of $\mathcal{N}[\mathbf{q}, \boldsymbol{\Theta}] = 0$ with a penalty strength $\mu > 0$. Since $\mathcal{N} = 0$ is not strictly constrained, the gradient is directly obtained without adjoint variables,

$$\nabla_{\mathbf{q}} \mathcal{J}_A = \nabla_{\mathbf{q}} \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] + \nabla_{\mathbf{q}} \mathcal{P}[\mathcal{N}, \mu] \tag{14.2a}$$

$$\nabla_{\boldsymbol{\Theta}} \mathcal{J}_A = \nabla_{\boldsymbol{\Theta}} \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] + \nabla_{\boldsymbol{\Theta}} \mathcal{P}[\mathcal{N}, \mu]. \tag{14.2b}$$

The subproblem (14.1) is then solved with (14.2) via standard gradient-based methods. Algorithm 3 summarizes this procedure. This is different from Algorithm 2 in three aspects: instead of solving $\mathcal{N}[\mathbf{q}, \boldsymbol{\Theta}] = 0$ for

Algorithm 3 Minimization of $\mathcal{J}_A$ (14.1)

Given: initial guess $(\mathbf{q}_0, \boldsymbol{\Theta}_0)$, tolerance ε, maximum search limit $K_{\max}$
Result: $(\mathbf{q}^*, \boldsymbol{\Theta}^*) = \operatorname{argmin} \mathcal{J}_A[\mathbf{q}, \boldsymbol{\Theta}, \mu]$
Evaluate the residual $\mathcal{N}_0 = \mathcal{N}[\mathbf{q}_0; \boldsymbol{\Theta}_0]$
$\mathcal{J}_A = \mathcal{J}[\mathbf{q}_0, \boldsymbol{\Theta}_0] + \mathcal{P}[\mathcal{N}_0, \mu]$
Compute $(\nabla_{\mathbf{q}} \mathcal{J}_{A,1}, \nabla_{\boldsymbol{\Theta}} \mathcal{J}_{A,1})$ via (14.2)
for $k = 1, \ldots, K_{\max}$ **do**
  $(\delta\mathbf{q}_k, \delta\boldsymbol{\Theta}_k) = \text{direction}(\nabla_{\mathbf{q}} \mathcal{J}_{A,1}, \nabla_{\boldsymbol{\Theta}} \mathcal{J}_{A,1}, \ldots, \nabla_{\mathbf{q}} \mathcal{J}_{A,k}, \nabla_{\boldsymbol{\Theta}} \mathcal{J}_{A,k})$ ▷ Eq. (11.12)
  $\tau_k = \operatorname{argmin} \mathcal{J}_A[\mathbf{q}_{k-1} + \tau\delta\mathbf{q}_k, \boldsymbol{\Theta}_{k-1} + \tau\delta\boldsymbol{\Theta}_k]$ ▷ Eq. (11.13)
  $(\mathbf{q}_k, \boldsymbol{\Theta}_k) = (\mathbf{q}_{k-1} + \tau_k\delta\mathbf{q}_k, \boldsymbol{\Theta}_{k-1} + \tau_k\delta\boldsymbol{\Theta}_k)$ ▷ Determine $(\mathbf{q}_k, \boldsymbol{\Theta}_k)$
  Evaluate the residual $\mathcal{N}_k = \mathcal{N}[\mathbf{q}_k; \boldsymbol{\Theta}_k]$
  $\mathcal{J}_A = \mathcal{J}[\mathbf{q}_k, \boldsymbol{\Theta}_k] + \mathcal{P}[\mathcal{N}_k, \mu]$
  Compute $(\nabla_{\mathbf{q}} \mathcal{J}_{A,k+1}, \nabla_{\boldsymbol{\Theta}} \mathcal{J}_{A,k+1})$ via (14.2)
  if $\|\nabla_{\mathbf{q}} \mathcal{J}_{A,k+1}\|, \|\nabla_{\boldsymbol{\Theta}} \mathcal{J}_{A,k+1}\| < \varepsilon$ **then**
    $(\mathbf{q}^*, \boldsymbol{\Theta}^*) = (\mathbf{q}_k, \boldsymbol{\Theta}_k)$
    Exit

$\mathbf{q}$, the residual $\mathcal{N}$ is evaluated; while Algorithm 2 only computes the gradient with respect to $\boldsymbol{\Theta}$, Algorithm 3 also computes the gradient with respect to $\mathbf{q}$; and the adjoint variable is not used to compute gradients.

A local minimum for the original problem (11.3) is then pursued by increasing μ, wherein each subproblem is solved with Algorithm 3. Algorithm 4 summarizes this procedure.

Algorithm 4 Penalty-based unconstrained optimization

Given: ${}_0\mathbf{q}$, ${}_0\boldsymbol{\Theta}$, ${}_1\mu$ ▷ The prescript ${}_i(\cdot)$ indicates i-th subproblem
Result: $(\mathbf{q}^*, \boldsymbol{\Theta}^*) = \operatorname{argmin} \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}]$ such that $\mathcal{N}[\mathbf{q}, \boldsymbol{\Theta}] = 0$
for $i = 1, \ldots$ **do**
  $({}_i\mathbf{q}, {}_i\boldsymbol{\Theta}) = \operatorname{argmin} \mathcal{J}_A[\mathbf{q}, \boldsymbol{\Theta}, {}_i\mu]$ with initial guess $({}_{i-1}\mathbf{q}, {}_{i-1}\boldsymbol{\Theta})$ ▷ Use Algorithm 3 with tolerance ${}_i\epsilon$
  if convergence test **then**
    $(\mathbf{q}^*, \boldsymbol{\Theta}^*) = ({}_i\mathbf{q}, {}_i\boldsymbol{\Theta})$
    Exit
  Choose ${}_{i+1}\mu > {}_i\mu$

Many variants have been proposed and studied for the penalty functional $\mathcal{P}$ [57, 149]. Comparing them and proposing the best functional for our formulation is beyond the scope of this study. Instead, we introduce the quadratic penalty method, which is widely used due to its simplicity and intuitive appeal [57]. The augmented Lagrangian method, as its extension, is also used in combination with it. Our framework will be formulated with a consideration for other compatible functionals that may be used in the future.

14.1.1 Quadratic penalty method

This method uses a squared norm reflecting the equality constraint [57, 58, 126],

$$\mathcal{P}[\mathcal{N}[\mathbf{q}, \mathbf{\Theta}], \mu] = \frac{\mu}{2} \|\mathcal{N}[\mathbf{q}, \mathbf{\Theta}]\|^2, \tag{14.3}$$

with $\|\cdot\|$ defined for $\mathbb{N} \equiv \mathbb{Q}$, so in an implementation it would be the residual of the governing equation. For a real vector space (11.7), it is shown that the sequence from Algorithm 4 with the quadratic penalty converges to a local optimum as $\mu \to \infty$ [57, 58].

Theorem 14.1 (Convergence of quadratic penalty method [57])**.** *Suppose* $\lim_{i\to\infty} {}_i\mu \to \infty$ *and* $\lim_{i\to\infty} {}_i\epsilon = 0$ *in Algorithm 4. If a limit solution* $(\mathbf{q}^*, \mathbf{\Theta}^*) = \lim_{i\to\infty} ({}_i\mathbf{q}, {}_i\mathbf{\Theta})$ *has linearly independent gradients* $(\nabla_{\mathbf{q}}\mathcal{N}_l, \nabla_{\mathbf{\Theta}}\mathcal{N}_l)$ *with* $l = 1, \ldots, n$*, then* $(\mathbf{q}^*, \mathbf{\Theta}^*)$ *satisfies the first-order optimality condition in Theorem 11.2 with an adjoint variable* $\mathbf{q}^{\dagger *}$

$$\mathbf{q}^{\dagger *} = -\lim_{i\to\infty} {}_i\mu \mathcal{N}[{}_i\mathbf{q}, {}_i\mathbf{\Theta}]. \tag{14.4}$$

Proof. See Nocedal and Wright [57, Theorem 17.2]. □

There is no set rule for choosing the penalty strength ${}_{i+1}\mu > {}_i\mu$ for subsequent subproblems. A typical choice is ${}_{i+1}\mu = \alpha\, {}_i\mu$ with $\alpha = 4$ and to adjust within $\alpha \in [1.5, 10]$ depending on the reduction achieved from the previous subproblem [57]; if a local minimum of $\mathcal{J}_A$ is found within a few search steps, larger $\alpha > 4$ is used, and otherwise smaller $\alpha < 4$. For our examples, we compare the number of search steps with the previous subproblem and adjust α accordingly: $\alpha = 2$ if more steps are taken, and $\alpha = 10$ otherwise.

Note (14.4) that the residual decreases with μ in the following form,

$$\mathcal{N}[{}_i\mathbf{q}, {}_i\mathbf{\Theta}] \approx -\frac{\mathbf{q}^{\dagger *}}{{}_i\mu}. \tag{14.5}$$

This suggests that, while $\mathcal{N} \to 0$ as μ increases, the residual always remains non-zero with a finite ${}_i\mu$. While this is often effective enough to find a local optimum, the residual can decrease faster by revising $\mathcal{P}$. This is the motivation of the Augmented Lagrangian method, which is introduced subsequently.

14.1.2 Augmented Lagrangian method

The augmented Lagrangian method augments the quadratic penalty (14.3) with the Lagrange multiplier (adjoint variable) [57, 58],

$$\mathcal{P}[\mathcal{N}[\mathbf{q},\boldsymbol{\Theta}];\mu,\mathbf{q}^\dagger] = \frac{\mu}{2}\|\mathcal{N}[\mathbf{q},\boldsymbol{\Theta}]\|^2 - \left\langle \mathbf{q}^\dagger, \mathcal{N}[\mathbf{q},\boldsymbol{\Theta}]\right\rangle. \tag{14.6}$$

Since the equality $\mathcal{N}[\mathbf{q},\boldsymbol{\Theta}] = 0$ is not constrained, there is no corresponding adjoint equation to solve for $\mathbf{q}^\dagger$ as in Chapter 11. Instead, we utilize (14.4) that the residual approximates the optimal adjoint variable. We consider a stationary point $({}_i\mathbf{q}, {}_i\mathbf{q}^\dagger, {}_i\boldsymbol{\Theta})$ for $\mathcal{J}_A$, which gradient (14.2) is zero,

$$\nabla_{\mathbf{q}}\mathcal{J}_A = \nabla_{\mathbf{q}}\mathcal{J}[{}_i\mathbf{q}, {}_i\boldsymbol{\Theta}] - (\nabla_{\mathbf{q}}\mathcal{N})^T({}_i\mathbf{q}^\dagger - {}_i\mu\,\mathcal{N}[{}_i\mathbf{q}, {}_i\boldsymbol{\Theta}]) = 0 \tag{14.7a}$$

$$\nabla_{\boldsymbol{\Theta}}\mathcal{J}_A = \nabla_{\boldsymbol{\Theta}}\mathcal{J}[{}_i\mathbf{q}, {}_i\boldsymbol{\Theta}] - (\nabla_{\boldsymbol{\Theta}}\mathcal{N})^T({}_i\mathbf{q}^\dagger - {}_i\mu\,\mathcal{N}[{}_i\mathbf{q}, {}_i\boldsymbol{\Theta}]) = 0, \tag{14.7b}$$

which is expressed on a real vector space (11.7) for simplicity. The goal is that $({}_i\mathbf{q}, {}_i\mathbf{q}^\dagger, {}_i\boldsymbol{\Theta})$ would approximate the first-order optimality condition in Theorem 11.2 after solving a sufficient number of subproblem. Comparing (14.7) with (11.9a-b), we deduce that the optimal adjoint variable should be approximated as

$$\mathbf{q}^{\dagger *} \approx {}_i\mathbf{q}^\dagger - {}_i\mu\,\mathcal{N}[{}_i\mathbf{q}, {}_i\boldsymbol{\Theta}]. \tag{14.8}$$

This immediately motivates an update formula for $\mathbf{q}^\dagger$

$${}_{i+1}\mathbf{q}^\dagger = {}_i\mathbf{q}^\dagger - {}_i\mu\,\mathcal{N}[{}_i\mathbf{q}, {}_i\boldsymbol{\Theta}], \tag{14.9}$$

after solving each subproblem in Algorithm 4. Algorithm 5 shows the extended procedure with (14.9). In

Algorithm 5 Augmented Lagrangian method

Given: ${}_0\mathbf{q}$, ${}_0\boldsymbol{\Theta}$, ${}_1\mathbf{q}^\dagger$, ${}_1\mu$ ▷ The prescript ${}_i(\cdot)$ indicates i-th subproblem
Result: $(\mathbf{q}^*, \mathbf{q}^{\dagger *}, \boldsymbol{\Theta}^*)$ = stationary point of $\mathcal{L}[\mathbf{q},\mathbf{q}^\dagger,\boldsymbol{\Theta}]$
for $i = 1, \ldots$ **do**
  $({}_i\mathbf{q}, {}_i\boldsymbol{\Theta}) = \operatorname{argmin} \mathcal{J}_A[\mathbf{q}, {}_i\mathbf{q}^\dagger, \boldsymbol{\Theta}, {}_i\mu]$ with initial guess $({}_{i-1}\mathbf{q}, {}_{i-1}\boldsymbol{\Theta})$ ▷ Use Algorithm 3
  if convergence test **then**
    $(\mathbf{q}^*, \mathbf{q}^{\dagger *}, \boldsymbol{\Theta}^*) = ({}_i\mathbf{q}, {}_i\mathbf{q}^\dagger, {}_i\boldsymbol{\Theta})$
    Exit
  ${}_{i+1}\mathbf{q}^\dagger = {}_i\mathbf{q}^\dagger - {}_i\mu\,\mathcal{N}[{}_i\mathbf{q}, {}_i\boldsymbol{\Theta}]$
  Choose ${}_{i+1}\mu \geq {}_i\mu$

this method, not only the optimal state $\mathbf{q}^*$ and control $\mathbf{\Theta}^*$ are pursued, but also the associated Lagrange multiplier $\mathbf{q}^{\dagger *}$ as in Theorem 11.2 is sought.

Now (14.8) suggests that the residual will have the form

$$\mathcal{N}[{}_i\mathbf{q}, {}_i\mathbf{\Theta}] \approx -\frac{\mathbf{q}^{\dagger *} - {}_i\mathbf{q}^{\dagger}}{{}_i\mu}, \tag{14.10}$$

after solving each subproblem. Thus, compared with (14.5) for quadratic penalty, the residual can also decrease as ${}_i\mathbf{q}^{\dagger} \to \mathbf{q}^{\dagger *}$, aside from increasing μ. The convergence and some salient properties of the augmented Lagrangian method are proven by Bertsekas [58]. This method is also attractive in that it can be linked with the adjoint method, which will be further formulated in detail in Section 14.3.

A condition for the convergence is to have a penalty strength larger than a threshold ${}_i\mu > \overline{\mu}$ [58, Proposition 2.7]. In essence, the Hessian of $\mathcal{J}_A$ has only positive eigenvalues with $\mu > \overline{\mu}$. While this threshold is difficult to evaluate, for our examples we simply solve a first few subproblems with the quadratic penalty (14.3) via Algorithm 4, so that the solution may enter a convex set for $\mathcal{J}_A$. Then Algorithm 5 takes over afterward for faster convergence to $\mathcal{N} = 0$. This transition is simple. The quadratic penalty method is identical to the augmented Lagrangian method with ${}_i\mathbf{q}^{\dagger} = \mathbf{0}$ not updated. So in the transition, the solution from the quadratic method is used as the initial guess ${}_0\mathbf{q}$, ${}_0\mathbf{\Theta}$ with ${}_1\mathbf{q}^{\dagger} = \mathbf{0}$ for Algorithm 5.

Standard penalty method relaxes the entire equation $\mathcal{N}[\mathbf{q}, \mathbf{\Theta}] = 0$, and includes $\mathbf{q}$ in the optimization space as an independent variable from $\mathbf{\Theta}$. However, doing it for large-scale flow simulations will significantly increase the optimization space dimension, leading to slower convergence. On the other hand, the multi-step penalty method relaxes only key equality constraints which are the root cause of non-convexity, which is illustrated subsequently.

14.2 Application to the logistics map of Chapter 12

The logistics map example (12.11) of Section 12.2.3 is used to show how the proposed method should be able to navigate the horseshoe mapping of more challenging systems:

$$q_{n+1} = q(q_n) \equiv (2q_n - 1)^2 \qquad q_n \in [0, 1],$$

with the objective functional (12.13),

$$\mathcal{J} = q_2 + \frac{1}{2}q_0.$$

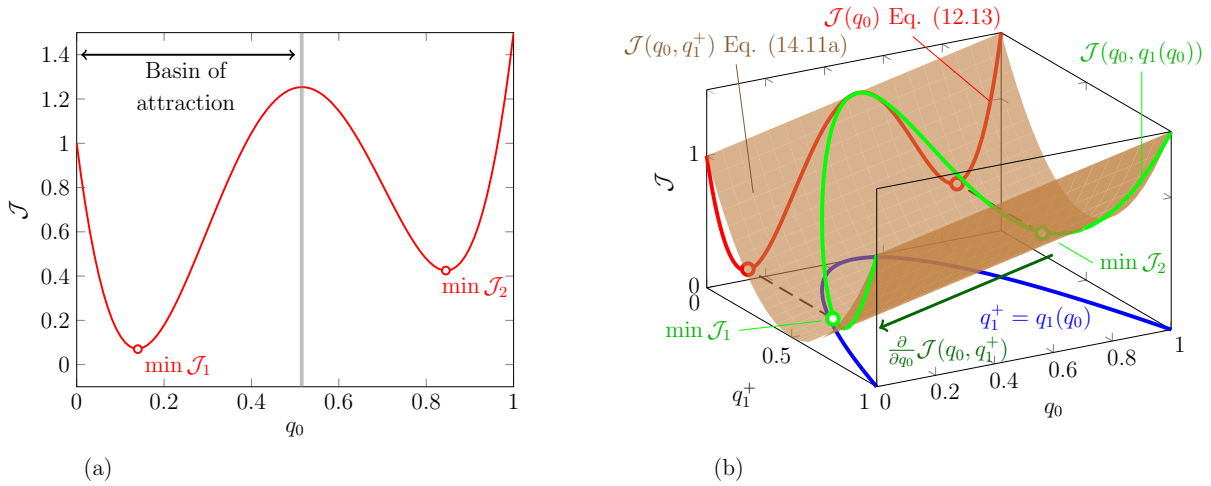


Figure 14.1: (a) The basin of attraction for the global minimum of $\mathcal{J}$ (12.13). (b) The objective $\mathcal{J}$ (14.11a) in (q_0, q_1^+)-space.

Figure 14.1 (a) shows the basin of attraction for the global minimum $\min \mathcal{J}_1$: gradient-based optimization for any initial q_0 outside this basin $q_0 \in [0, 0.515]$ converges to the other local minimum $\min \mathcal{J}_2$. This is emblematic of the local minima landscape that the horseshoe mapping creates (Section 12.2.3).

To expose a path that reaches $\min \mathcal{J}_1$ for any q_0, we introduce an intermediate state q_1^+, which corresponds to a possible q_1 though not one that is strictly tied to the current q_0. Instead of viewing $q_2 \equiv q(q(q_0))$ as a direct function of q_0, we replace $q_1 \equiv q(q_0)$ with q_1^+ and consider $\mathcal{J}$ as it depends on q_0 and q_1^+,

$$\mathcal{J}(q_0, q_1^+) = q_2(q_1^+) + \frac{1}{2}q_0 = (2q_1^+ - 1)^2 + \frac{1}{2}q_0, \tag{14.11a}$$

with an additional equality constraint for the intermediate state

$$q_1^+ = q_1(q_0). \tag{14.11b}$$

This is an exact rearrangement of the original problem, however without the intermediate constraint (14.11b), the two-dimensional objective functional $\mathcal{J}(q_0, q_1^+)$ has the unique global minimum at $(q_0, q_1^+) = (0, 0.5)$, as shown in Figure 14.1 (b). The basin of attraction for the global minimum is thus all of the (q_0, q_1^+)-domain. Only the constraint $q_1^+ = q_1(q_0)$ limits the basin of attraction. With this constraint, $\mathcal{J}(q_0, q_1(q_0))$ is a curve through the two-dimensional space, shown as the green line in Figure 14.1 (b). Three extrema (two minima) are created in the green line $\mathcal{J}(q_0, q_1(q_0))$, as the blue curve $q_1^+ = q_1(q_0)$ is projected and folded onto the surface $\mathcal{J}(q_0, q_1)$. The result is the $\mathcal{J}(q_0)$ of Figure 14.1. This illustrates how the convexity per Definition 11.2 breaks down: while $\mathcal{J}$ may be defined to be convex on $\mathbb{Q}$ (q_0, q_1^+), the set

dom($\mathcal{N}$) (constrained by $q_1^+ = q_1(q_0)$) becomes non-convex and thus a stationary point only becomes the minimum of a smaller subset.

The strategy is therefore to relax this intermediate constraint (14.11b) to expand the basin of attraction for objective functionals, such as $\mathcal{J}(q_0, q_1^+)$ in Figure 14.1 (b), avoiding the confinement introduced by the horseshoe mapping. This is compatible with the optimization framework of Section 14.1. The objective functional is augmented with the penalty,

$$\mathcal{J}_A = (2q_1^+ - 1)^2 + \frac{1}{2}q_0 + \frac{\mu}{2}\{q_1^+ - (2q_0 - 1)^2\}^2, \tag{14.12}$$

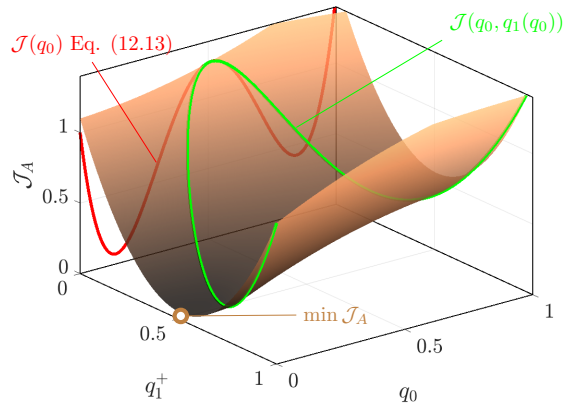
which is minimized over $(\mathbf{q}, \boldsymbol{\Theta}) = (q_1^+, q_0)$ by Algorithm 4. The intermediate state q_1^+ is optimized together with q_0, until $q_1^+ \to q_1 \equiv (2q_0 - 1)^2$ as μ increases. This circumvents the non-convexity $\mathcal{J}$.

Figure 14.2 (a) shows that $\mathcal{J}_A$ with a weak $\mu = 10^{-1}$ has a nearly idential shape to $\mathcal{J}(q_0, q_1^+)$ (14.11a) in Figure 14.1 (b). So the entire domain (q_0, q_1^+) is still the basin of attraction for its global minimum. As μ increases, $\mathcal{J}_A$ converges to the original $\mathcal{J}(q_0, q_1(q_0))$ as shown in Figure 14.2 (b). This $\mathcal{J}_A$ with a larger μ has the similarly limited basin of attraction as the original $\mathcal{J}(q_0)$. However, if used correctly, the optimized (q_0, q_1^+) will have already entered the $\mathcal{J}_1$ basin before the penalty strength μ is increased. Figure 14.3 shows how the local minimizer of $\mathcal{J}_A$ converges to the global minimizer of $\mathcal{J}$ in (12.13) as μ increases. Any initial guess (q_0, q_1^+) converges to this optimum.

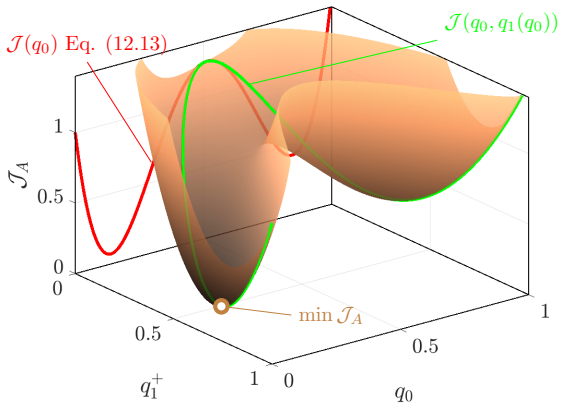
The augmented Lagrangian method can accelerate enforcement of the intermediate constraint via the additional adjoint term. In this case, the augmented objective functional is

$$\begin{aligned}\mathcal{J}_A[q_0, q_1^+; q_1^{\dagger+}, \mu] = (2q_1^+ - 1)^2 + \frac{1}{2}q_0 \\ + \frac{\mu}{2}\{q_1^+ - (2q_0 - 1)^2\}^2 \underbrace{-q_1^{\dagger+}\{q_1^+ - (2q_0 - 1)^2\}}_{\mathcal{J}_{A,\mathrm{adj}}},\end{aligned} \tag{14.13}$$

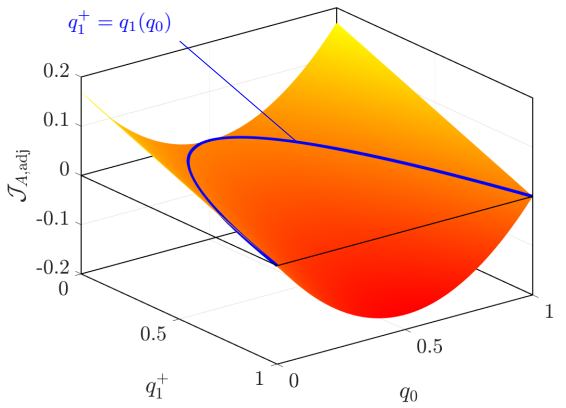
which is minimized over $(\mathbf{q}, \mathbf{q}^\dagger, \boldsymbol{\Theta}) = (q_1^+, q_1^{\dagger+}, q_0)$ by Algorithm 5. As the adjoint $q_1^{\dagger+}$ is updated with (14.9), the solution converges to the optimum even with constant $\mu = 10^{-1}$, as shown in Figure 14.3. Figure 14.2 (c) shows its adjoint-associated term $\mathcal{J}_{A,\mathrm{adj}} = -q_1^{\dagger+}\{q_1^+ - q_1(q_0)\}$, which guides the minimum of $\mathcal{J}_A$ toward the intermediate constraint. This adjoint term shifts $\min \mathcal{J}_A$ of quadratic penalty with same $\mu = 10^{-1}$ (Figure 14.2 a) toward the feasible optimum, as shown in Figure 14.2 (d). This demonstrates that the augmented Lagrangian method can enforce the equality constraint faster than the quadratic penalty method at smaller μ [57, 58].



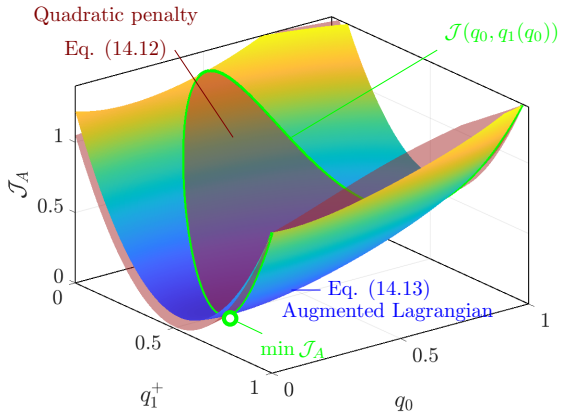
(a) Quadratic penalty $\mu = 10^{-1}$



(b) Quadratic penalty $\mu = 10^{1}$



(c) Augmentative adjoint term $\mathcal{J}_{A,\text{adj}}$



(d) Augmented Lagrangian $\mu = 10^{-1}$

Figure 14.2: Logistics map demonstration objective functionals: $\mathcal{J}_A$ (14.12) for the quadratic penalty method with (a) $\mu = 10^{-1}$ and (b) $\mu = 10^{2}$; (c) additional adjoint term $\mathcal{J}_{A,\text{adj}} = -q_1^{\dagger+}\{q_1^+ - q_1(q_0)\}$ for the augmented Lagrangian method (converged), and (d) $\mathcal{J}_A$ (14.13) for the augmented Lagrangian method with $\mu = 10^{-1}$ compared with the quadratic penalty method.

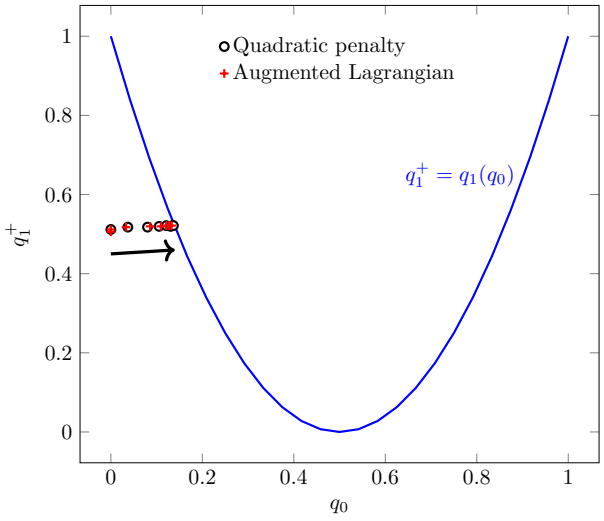


Figure 14.3: Local minimizer of $\mathcal{J}_A$ with the quadratic penalty method and the augmented Lagrangian method. The black arrow indicates the converging direction. For the quadratic penalty method, the penalty strength is increased from $\mu = 10^{-1}$ to $\mu = 10^2$. For the augmented Lagrangian method, the adjoint variable $q_1^{\dagger+}$ is updated with constant $\mu = 10^{-1}$.

14.3 Formulation for time-continuous dynamical systems

Penalty-based optimization is extended to time-continuous dynamical systems by redefining the governing equation $\mathcal{N}$ and the flow state $\mathbf{q}$ with auxiliary intermediate conditions, resulting in a modified inner product for the adjoint formulation. The multi-step penalty-based optimization framework is formulated subsequently.

14.3.1 Modified governing equation with intermediate constraints

We consider modifications needed for the general governing equation (11.17),

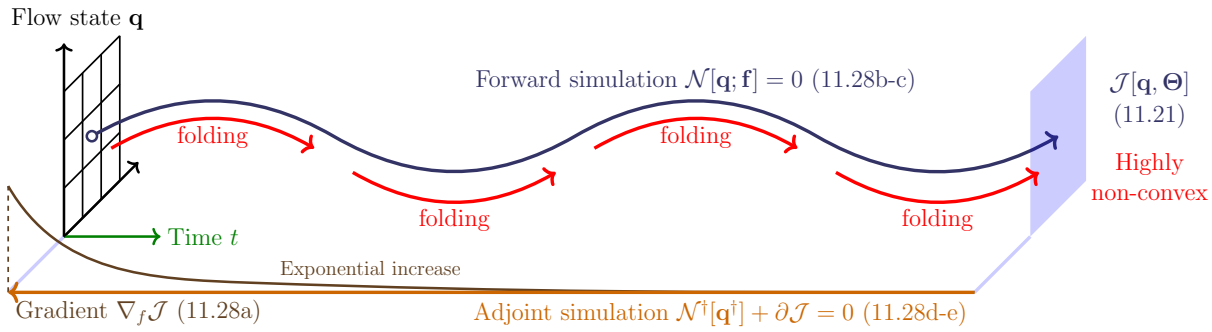
$$\frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \boldsymbol{\Theta}] = \mathbf{0}$$

with

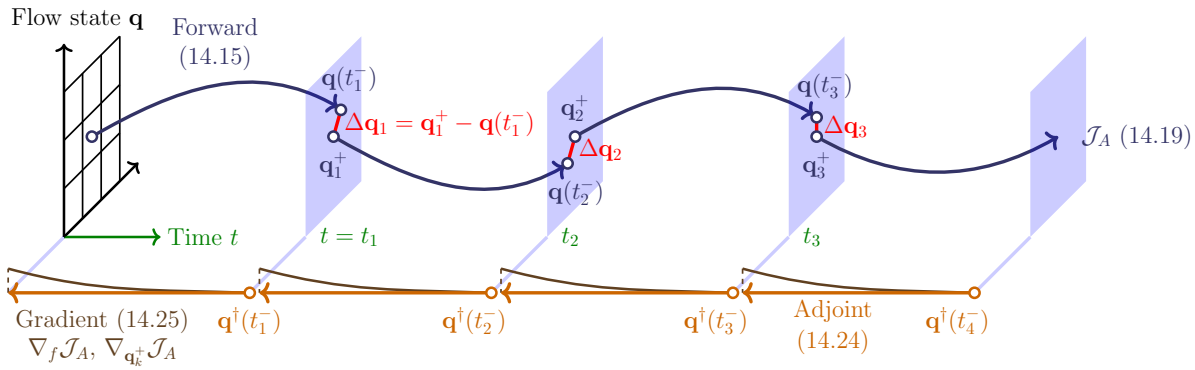
$$\mathbf{q}(t_i) = \mathbf{q}_0,$$

and the objective functional (11.21)

$$\mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] = \phi[\mathbf{q}(t_f)] + \int_{t_i}^{t_f} \mathcal{I}[\mathbf{q}(t), \boldsymbol{\Theta}(t)]\, dt.$$



(a) Standard gradient-based method



(b) Multi-step penalty-based method

Figure 14.4: Schematic of (a) standard gradient-based-based optimization and (b) multi-step penalty-based optimization with three intermediate conditions.

Figure 14.4 (a) shows a schematic of the standard optimization with (11.17) and (11.21). We recall from (12.10) that the interval T can be considered a discrete mapping, and introduce intermediate conditions $\mathbf{q}_k^+$ for each time $t_k = t_i + kT$,

$$\frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q};\boldsymbol{\Theta}] = \mathbf{0} \qquad \text{for } t \in [t_k, t_{k+1}) \tag{14.15a}$$

$$\mathbf{q}(t_k) = \mathbf{q}_k^+ \qquad \text{for } k = 0, \ldots, N-1. \tag{14.15b}$$

The intermediate constraints analogous to (14.11b) are

$$\Delta\mathbf{q}_k \equiv \mathbf{q}_k^+ - \mathbf{q}(t_k^-) = 0 \qquad \text{for } k = 1, \ldots, N-1, \tag{14.15c}$$

where $\mathbf{q}(t_k^-)$ is the terminal state of the previous interval $t \in [t_{k-1}, t_k)$,

$$\mathbf{q}(t_k^-) = \mathbf{q}_{k-1}^+ + \int_{t_{k-1}}^{t_k^-} \mathcal{R}[\mathbf{q};\boldsymbol{\Theta}]\, dt. \tag{14.16}$$

Figure 14.4 (b) shows the modified system schematically. Only the (14.15c) constraint is relaxed, and (14.15a) and (14.15b) are strictly enforced within the intervals. The state trajectory $\mathbf{q} \in \mathbb{Q}$ is piecewise-continuous with intermediate discontinuities $\Delta\mathbf{q}_k$ at $t = t_k$.

For the adjoint formulation, the inner-product for the state space $\mathbb{Q}$ is modified with sub-inner-products for piecewise-continuous trajectories,

$$\langle \mathbf{p}, \mathbf{q} \rangle_{\mathbb{Q}} = \sum_{k=1}^{N} \langle \mathbf{p}, \mathbf{q} \rangle_{\mathbb{Q}_k}, \tag{14.17}$$

where the subspaces $\mathbb{Q}_k \subset \mathbb{Q}$ cover time intervals $t \in [t_{k-1}, t_k)$,

$$\langle \mathbf{p}, \mathbf{q} \rangle_{\mathbb{Q}_k} = \int_{t_{k-1}}^{t_k^-} \langle \mathbf{p}(t), \mathbf{q}(t) \rangle_{\mathbb{Q}^+}\, dt. \tag{14.18}$$

In our examples, (14.15) is integrated in time with the standard four-step Runge–Kutta method, and (11.43) is used to approximate time integral (14.18).

14.3.2 Penalty-based optimization

We first formulate the augmented objective functional $\mathcal{J}_A$ (14.1), for which Algorithm 4 can be applied, with a generic penalty form for the intermediate constraints (14.15c),

$$\mathcal{J}_A[\mathbf{q}, \boldsymbol{\Theta}; \{\mathbf{q}_k^+\}, \mu] = \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] + \mathcal{P}[\{\Delta\mathbf{q}_k\}, \mu], \tag{14.19}$$

with $\{\mathbf{q}_k^+\} = (\mathbf{q}_1^+, \mathbf{q}_2^+, \ldots, \mathbf{q}_{N-1}^+)$ and $\{\Delta\mathbf{q}_k\} = (\Delta\mathbf{q}_1, \Delta\mathbf{q}_2, \ldots, \Delta\mathbf{q}_{N-1})$. The subproblem for Algorithm 4 is then,

$$\big({}_i\{\mathbf{q}_k^+\}, {}_i\boldsymbol{\Theta}\big) = \underset{\{\mathbf{q}_k^+\}, \boldsymbol{\Theta}}{\operatorname{argmin}} \mathcal{J}_A[\mathbf{q}, \boldsymbol{\Theta}; \{\mathbf{q}_k^+\}, {}_i\mu], \qquad \text{with (14.15a) and (14.15b)}, \tag{14.20}$$

which forms Algorithm 6 with the subproblem (14.20). The subproblem (14.20) is solved using Algorithm 2,

Algorithm 6 Multi-point penalty-based method

Given: ${}_0\{\mathbf{q}_k^+\}$, ${}_0\boldsymbol{\Theta}$, ${}_1\mu$
for $i = 1, \ldots$ **do**
  Solve subproblem (14.20) with ${}_{i-1}\{\mathbf{q}_k^+\}$, ${}_{i-1}\boldsymbol{\Theta}$ ▷ Algorithm 2 with gradient (14.25)
  if convergence test **then**
    Exit
  Choose ${}_{i+1}\mu > {}_i\mu$

which requires the gradient of $\mathcal{J}_A$ (14.19) to the control parameter $\boldsymbol{\Theta}$ and the intermediate conditions $\{\mathbf{q}_k^+\}$. This is formulated subsequently by the standard gradient-based method.

14.3.3 Adjoint-based gradient for the subproblem (14.20)

The Lagrangian associated with $\mathcal{J}_A$ (14.19) is

$$\begin{aligned}\mathcal{L}_A[\mathbf{q}, \mathbf{q}^\dagger, \boldsymbol{\Theta}] &= \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] + \mathcal{P}[\{\Delta\mathbf{q}_k\}, \mu] - \left\langle \mathbf{q}^\dagger, \frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \boldsymbol{\Theta}] \right\rangle_{\mathbb{Q}} \\ &= \mathcal{J}[\mathbf{q}, \boldsymbol{\Theta}] + \mathcal{P}[\{\Delta\mathbf{q}_k\}, \mu] - \sum_{k=1}^{N} \left\langle \mathbf{q}^\dagger, \frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \boldsymbol{\Theta}] \right\rangle_{\mathbb{Q}_k}.\end{aligned} \tag{14.21}$$

Linearizing it and formulating its adjoint as in Section 11.3 yields

$$\delta\mathcal{L}_A[\mathbf{q}, \mathbf{q}^\dagger, \boldsymbol{\Theta}] =$$

$$
\left.\begin{aligned}
&-\sum_{k=1}^{N}\left\langle \delta\mathbf{q}^\dagger, \frac{d\mathbf{q}}{dt}-\mathcal{R}[\mathbf{q};\boldsymbol{\Theta}]\right\rangle_{\mathbb{Q}_k}\\
&-\left\langle \mathbf{q}^\dagger(t_i), \delta\mathbf{q}(t_i)\right\rangle_{\mathbb{Q}^+}
\end{aligned}\right\}\ \text{Governing equation}
$$

$$
\left.\begin{aligned}
&-\left\langle \mathbf{q}^\dagger(t_f)-\frac{\partial\phi}{\partial\mathbf{q}(t_f)}, \delta\mathbf{q}(t_f)\right\rangle_{\mathbb{Q}^+}\\
&-\sum_{k=1}^{N-1}\left\langle \mathbf{q}^\dagger(t_k^-)-\frac{\partial\mathcal{P}}{\partial\mathbf{q}(t_k^-)}, \delta\mathbf{q}(t_k^-)\right\rangle_{\mathbb{Q}^+}\\
&+\sum_{k=1}^{N}\left\langle \frac{d\mathbf{q}^\dagger}{dt}+\frac{\partial\mathcal{R}}{\partial\mathbf{q}}^{\dagger}\mathbf{q}^\dagger+\frac{\partial\mathcal{I}}{\partial\mathbf{q}}, \delta\mathbf{q}\right\rangle_{\mathbb{Q}_k}
\end{aligned}\right\}\ \text{Adjoint equation}
$$

$$
\left.\begin{aligned}
&+\left\langle \frac{\partial\mathcal{R}}{\partial\boldsymbol{\Theta}}^{\dagger}\mathbf{q}^\dagger+\frac{\partial\mathcal{I}}{\partial\boldsymbol{\Theta}}, \delta\boldsymbol{\Theta}\right\rangle_{\mathbb{T}}\\
&+\sum_{k=1}^{N-1}\left\langle \mathbf{q}^\dagger(t_k)+\frac{\partial\mathcal{P}}{\partial\mathbf{q}_k^+}, \delta\mathbf{q}_k^+\right\rangle_{\mathbb{Q}^+},
\end{aligned}\right\}\ \text{Gradient}
\tag{14.22}
$$

where $\frac{\partial\mathcal{P}}{\partial\mathbf{q}_k^+}$ and $\frac{\partial\mathcal{P}}{\partial\mathbf{q}(t_k^-)}$ are weak-form gradients of $\mathcal{P}$, so its variation is

$$
\delta\mathcal{P}=\sum_{k=1}^{N-1}\left[\left\langle \frac{\partial\mathcal{P}}{\partial\mathbf{q}_k^+}, \delta\mathbf{q}_k^+\right\rangle_{\mathbb{Q}^+}+\left\langle \frac{\partial\mathcal{P}}{\partial\mathbf{q}(t_k^-)}, \delta\mathbf{q}(t_k^-)\right\rangle_{\mathbb{Q}^+}\right].
\tag{14.23}
$$

In (14.22), the first two inner products vanish because the governing equation (14.15a) is enforced and $\delta\mathbf{q}(t_0)=\mathbf{0}$ since the initial condition in (14.15b) is fixed. The next three inner products are related to the adjoint solution. For each time interval $t\in[t_k,t_{k+1})$ we solve the adjoint equation

$$
-\frac{d\mathbf{q}^\dagger}{dt}=\frac{\partial\mathcal{R}}{\partial\mathbf{q}}^{\dagger}\mathbf{q}^\dagger+\frac{\partial\mathcal{I}}{\partial\mathbf{q}},
\tag{14.24a}
$$

in reverse time with conditions

$$
\mathbf{q}^\dagger(t_f)=\frac{\partial\Phi}{\partial\mathbf{q}(t_f)}\qquad\text{for } k=N-1
\tag{14.24b}
$$

$$
\mathbf{q}^\dagger(t_{k+1}^-)=\frac{\partial\mathcal{P}}{\partial\mathbf{q}(t_{k+1}^-)}\qquad\text{for } k=1,\ldots,N-2.
\tag{14.24c}
$$

Each interval starts from intermediate condition $\mathbf{q}^\dagger(t_{k+1}^-)$, then progresses in reverse time to $\mathbf{q}^\dagger(t_k)$. Then the adjoint simulation for the previous interval $t\in[t_{k-1},t_k)$ starts with its own distinct intermediate condition $\mathbf{q}^\dagger(t_k^-)$. This procedure is illustrated schematically in Figure 14.4 (b). The last two inner products in (14.22)

provide the gradient for updating the control and the intermediate conditions,

$$\frac{\partial \mathcal{L}_A}{\partial \mathbf{q}_k^+} = \mathbf{q}^\dagger(t_k) + \frac{\partial \mathcal{P}}{\partial \mathbf{q}_k^+} \tag{14.25a}$$

$$\frac{\partial \mathcal{L}_A}{\partial \mathbf{\Theta}} = \frac{\partial \mathcal{R}}{\partial \mathbf{\Theta}}^\dagger \mathbf{q}^\dagger + \frac{\partial \mathcal{I}}{\partial \mathbf{\Theta}}, \tag{14.25b}$$

which are used in Algorithm 6.

For the quadratic penalty method,

$$\mathcal{P}[\{\Delta \mathbf{q}_k\}, \mu] = \frac{\mu}{2} \sum_{k=1}^{N-1} \|\mathbf{q}_k^+ - \mathbf{q}(t_k^-)\|_{\mathbb{Q}^+}^2, \tag{14.26}$$

with gradients for (14.24) and (14.25),

$$\frac{\partial \mathcal{P}}{\partial \mathbf{q}_k^+} = \mu \left[\mathbf{q}_k^+ - \mathbf{q}(t_k^-)\right] \tag{14.27a}$$

$$\frac{\partial \mathcal{P}}{\partial \mathbf{q}(t_k^-)} = -\mu \left[\mathbf{q}_k^+ - \mathbf{q}(t_k^-)\right]. \tag{14.27b}$$

Similarly, for the augmented Lagrangian method,

$$\mathcal{P}[\{\Delta \mathbf{q}_k\}, \mu] = \frac{\mu}{2} \sum_{k=1}^{N-1} \|\mathbf{q}_k^+ - \mathbf{q}(t_k^-)\|_{\mathbb{Q}^+}^2 - \sum_{k=1}^{N-1} \left\langle \mathbf{q}_k^{\dagger +}, \mathbf{q}_k^+ - \mathbf{q}(t_k^-) \right\rangle_{\mathbb{Q}^+}, \tag{14.28}$$

with gradients for (14.24) and (14.25),

$$\frac{\partial \mathcal{P}}{\partial \mathbf{q}_k^+} = -\mathbf{q}_k^{\dagger +} + \mu \left[\mathbf{q}_k^+ - \mathbf{q}(t_k^-)\right] \tag{14.29a}$$

$$\frac{\partial \mathcal{P}}{\partial \mathbf{q}(t_k^-)} = \mathbf{q}_k^{\dagger +} - \mu \left[\mathbf{q}_k^+ - \mathbf{q}(t_k^-)\right]. \tag{14.29b}$$

As in Algorithm 5, the augmented adjoint variables $\{\mathbf{q}_k^{\dagger +}\}$ are updated after solving each subproblem (14.20),

$${}_{i+1}\mathbf{q}_k^{\dagger +} = {}_i\mathbf{q}_k^{\dagger +} - {}_i\mu \left[{}_i\mathbf{q}_k^+ - {}_i\mathbf{q}(t_k^-)\right], \qquad \text{for } k = 1, \ldots, N-1, \tag{14.30}$$

for which Algorithm 5 is recast as Algorithm 7.

As discussed previously, we recommend using Algorithm 6 with quadratic penalty in early stages then

Algorithm 7 Multi-point Augmented Lagrangian method

Given: ${}_0\{\mathbf{q}_k^+\}$, ${}_0\boldsymbol{\Theta}$, ${}_1\{\mathbf{q}_k^{\dagger+}\}$, ${}_1\mu$
for $i = 1, \ldots$ **do**
  Subproblem (14.20) with initial guess ${}_{i-1}\{\mathbf{q}_k^+\}$, ${}_{i-1}\boldsymbol{\Theta}$ ▷ Use Algorithm 2 with the gradient (14.25)
  if convergence test **then**
    Exit
  Update ${}_{i+1}\{\mathbf{q}_k^{\dagger+}\}$ with (14.30)
  Choose ${}_{i+1}\mu \geq {}_i\mu$

switch to Algorithm 7 with the augmented Lagrangian method. When switching to the augmented Lagrangian method, the solution from the quadratic method is simply used as the initial guess ${}_0\{\mathbf{q}_k^+\}$, ${}_0\boldsymbol{\Theta}$ with ${}_1\{\mathbf{q}_k^{\dagger+}\} = \{\mathbf{0}\}$ for Algorithm 7.

14.4 Demonstration on the Lorenz example

Because it is so easily dissected, the Lorenz example is used here to illustrate the full method for the numerical solution of a time-continuous system. More challenging examples, including turbulence, are considered in the following chapter. The optimization period $t_f - t_0 = 20$ is split evenly into $n = 200$ intervals of period $T = 0.1$. Based on e-folding time of this system $t_\lambda \approx 1.11$ from Figure 12.8, the sensitivity is anticipated to be amplified by only about a factor of 1.09 within intervals. The optimization uses only the quadratic penalty (14.26), and Algorithm 6 is applied for 6 iterations of subproblem (14.20). The minimization of each subproblem is deemed sufficient when

$$\left\| \frac{\partial \mathcal{L}_A}{\partial \boldsymbol{\Theta}} \right\|_{\mathbb{T}}^2 + \sum_{k=1}^{N-1} \left\| \frac{\partial \mathcal{L}_A}{\partial \mathbf{q}_k^+} \right\|_{\mathbb{Q}^+}^2 < 10^{-8}. \tag{14.31}$$

The penalty strength μ is increased by a factor of 10 at each subproblem, starting from ${}_1\mu = 10^{-5}$.

Figure 14.5 (a) compares the result to the standard gradient-based method from Figure 10.3. The multi-point method achieves a 99.99% reduction of $\mathcal{J}$ versus the 44.6% for the standard gradient-based method. The intermediate discontinuities are decreased to $\|\Delta\mathbf{q}_k\|_{\mathbb{Q}^+} < 10^{-13}$. There is no need for the augmented Lagrangian method to accelerate the optimization. Figure 14.5 (b) compares the step sizes taken in line searches to the standard gradient-based method. Clearly, the multi-point method searches a much larger region, only decreasing the step size with increasing μ. This enables identification of the effective control shown in Figure 14.5 (c) and (d). It is not concentrated toward early times, unlike the optimized control with standard gradient-based method. Similarly, without exception it suppresses all the peaks of $\mathcal{I}(t)$. Figure 14.6 further shows that it is more regular compared to the nonlinear feedback control of Appendix C.

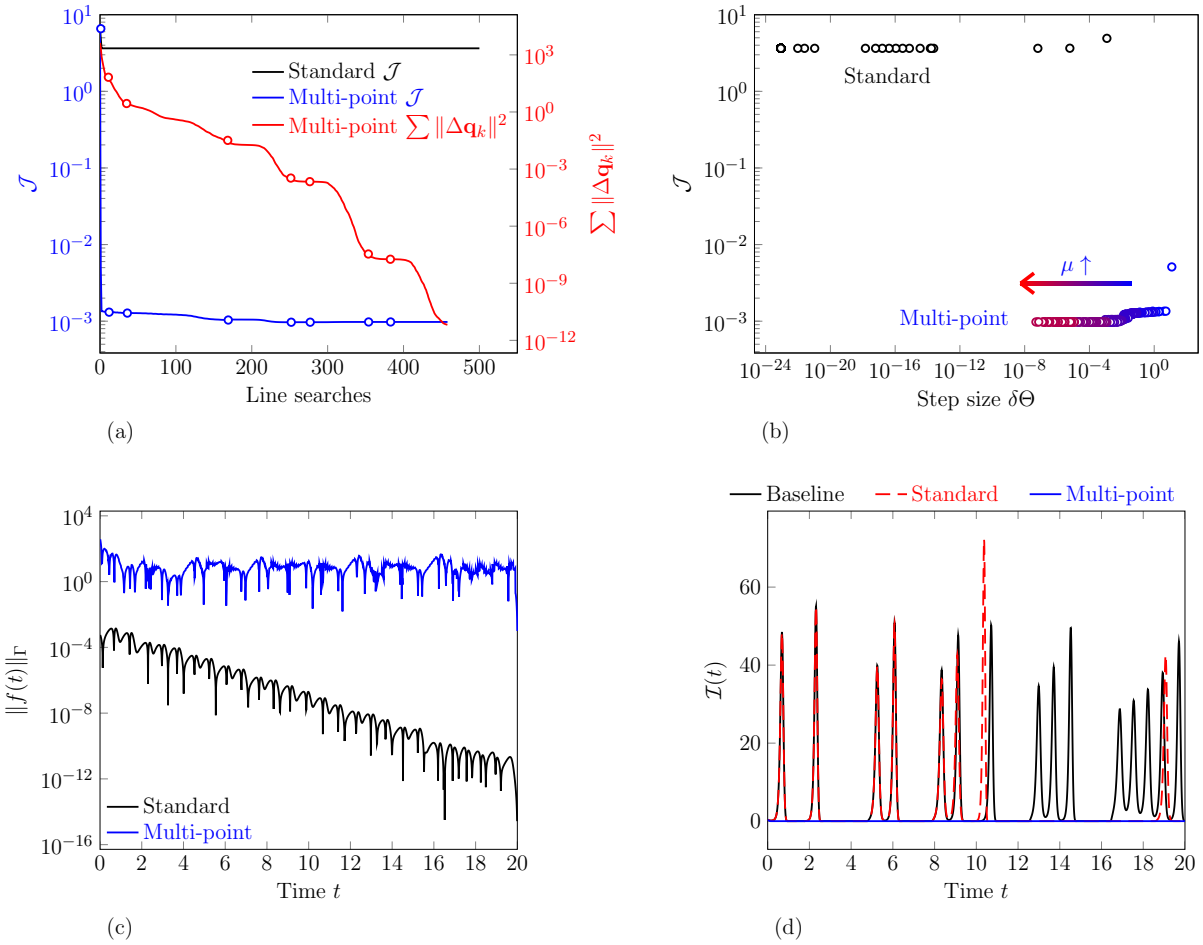


Figure 14.5: Multi-point penalty-based method applied to the Lorenz example of Section 12.1. (a) Reduction of $\mathcal{J}$ (12.3) and the intermediate discontinuities of the multi-point method. Markers indicate the updates of μ in Algorithm 6. (b) Step sizes taken in the optimizations. For the multi-point method, color changing from blue to red indicates the increase of μ. (c) The control strength $f(t)$ of the optimized controls. (d) The instantaneous objective functional $\mathcal{I}(t)$ (12.4) of the baseline solution and the controlled solutions.

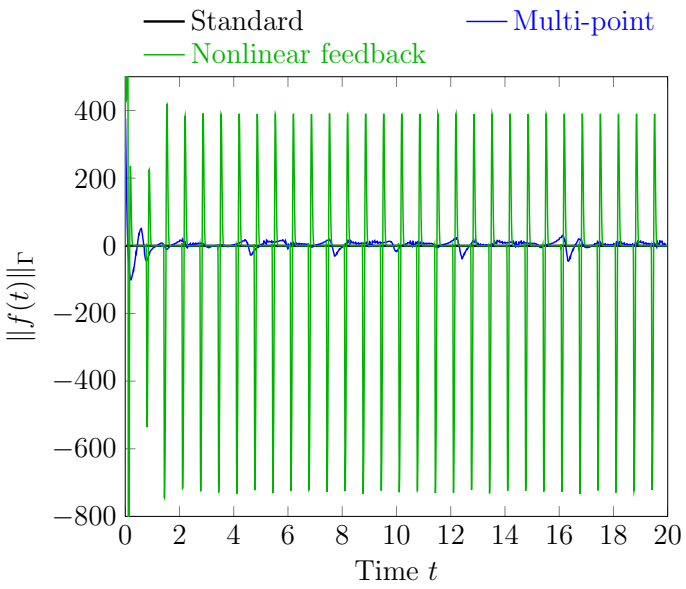


Figure 14.6: The control $f(t)$ in (10.1), optimized by the standard gradient-based method, the nonlinear feedback control from Appendix C, and the multi-point method.

The optimized state trajectory was shown in Figure 10.4.

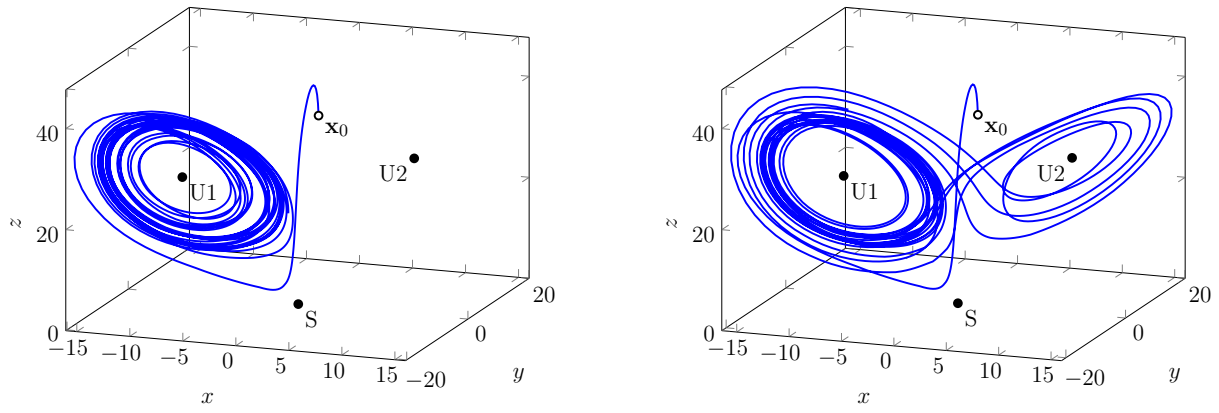
The control is successful, but the pathological sensitivity to initial condition still raises questions about the utility of the optimized solution: Does it approximate a realistic trajectory? Can it be used for the actual control? These are discussed in following sections.

14.5 Approximation in a shadowing sense

Although intermediate discontinuities $\Delta\mathbf{q}_k$ of the optimized solution (Figure 14.5 a) are small, and the optimized state trajectory in Figure 10.4 seems smooth and continuous in time, we consider the details of the procedure, recognizing that $\sum \|\Delta\mathbf{q}_k\| \to 0$ only for $\mu \to \infty$ [57, 58]. Because chaotic dynamics amplify any small deviation, a question remains whether or not the result can be considered physical, reflecting the true dynamics of the actual system.

We can indeed confirm that the small $\Delta\mathbf{q}_k$ does affect the optimized solution. Figure 14.7 (a) shows the piecewise-continuous trajectory with the optimized control forcing $\boldsymbol{\Theta} = f(t)$ and small but finite $\{\Delta\mathbf{q}_k\}$. Applying the same $\boldsymbol{\Theta}$ with $\mathbf{q}(t_k) = \mathbf{q}_k^+ = \mathbf{q}(t_k^-)$ so $\{\Delta\mathbf{q}_k\} = \mathbf{0}$ leads to the continuous trajectory shown in Figure 14.7 (b), which eventually deviates and return to orbit $U2$.

Though this is expected, this same question underlies any numerical solution of a chaotic dynamical system. Even without $\Delta\mathbf{q}_k$, a numerical solution is still subject to errors from its discretization and finite-precision arithmetics, so its true fidelity is likewise questionable. Similarly, no a real-world actuator can



(a) with discontinuity

(b) without discontinuity

Figure 14.7: (a) The optimized state trajectory of the Lorenz system (10.1) that is piecewise continuous. (b) The state trajectory with the optimized control forcing and with the intermediate constraints strictly enforced.

apply the optimized $\boldsymbol{\Theta}$ without error, nor can a state-estimator exactly identify the flow state $\mathbf{q}$.

This error susceptibility of numerical solutions of chaotic dynamical systems is well recognized [13, 39, 41, 42], and it has been extensively studied concerning in what sense they approximate the true solution [150–153]. While numerical solutions do deviate from the true trajectory with the same initial condition, in many cases there is thought to exist a different true trajectory with slightly different initial conditions that closely tracks the computed solution for a long time [152], lending credence to the numerical solution. That is, there is a true trajectory that *shadows* the numerical solution.

Definition 14.1 (shadowing [39, 151])**.** *Let $\mathbf{q}^* \in \mathbb{Q}$ be the exact trajectory of (11.17), so*

$$\left\| \frac{\partial \mathbf{q}^*(t)}{\partial t} - \mathcal{R}[\mathbf{q}^*(t)] \right\|_{\mathbb{Q}^+} = 0 \qquad \forall t > 0, \tag{14.32}$$

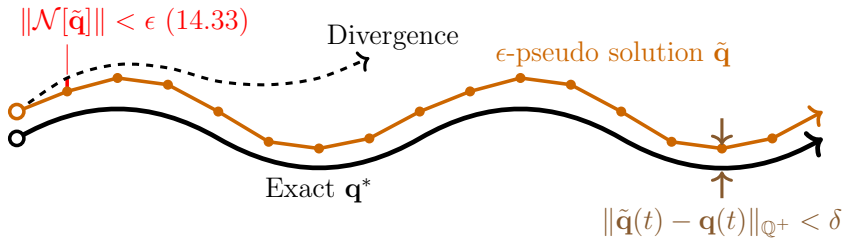
and let $\tilde{\mathbf{q}} \in \mathbb{Q}$ be a ϵ-pseudo trajectory of (11.17) which is subject to error

$$\left\| \frac{\partial \tilde{\mathbf{q}}(t)}{\partial t} - \mathcal{R}[\tilde{\mathbf{q}}(t)] \right\|_{\mathbb{Q}^+} < \epsilon \qquad \forall t > 0. \tag{14.33}$$

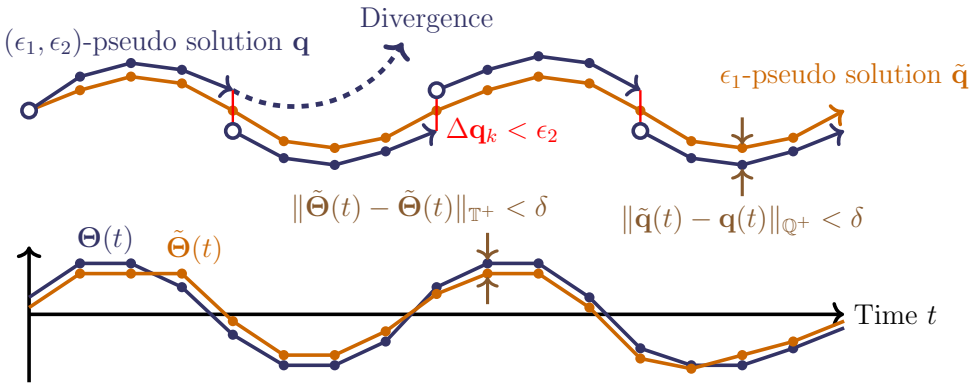
The exact trajectory $\mathbf{q}^*$ δ-shadows the numerical solution $\tilde{\mathbf{q}}$ for $t \in [t_i, t_f]$ if

$$\|\mathbf{q}^*(t) - \tilde{\mathbf{q}}(t)\|_{\mathbb{Q}^+} < \delta \qquad \forall t \in [t_i, t_f]. \tag{14.34}$$

Figure 14.8 (a) illustrates an exact solution $\mathbf{q}^*$ that shadows an ϵ-pseudo solution. There exists a δ-shadowing



(a) Definition 14.1



(b) Definition 14.2

Figure 14.8: Schematics of shadowing: (a) Exact solution $\mathbf{q}^*$ shadowing ϵ-pseudo solution $\tilde{\mathbf{q}}$, (b) ϵ_1-pseudo solution $\tilde{\mathbf{q}}$ shadowing multi-point (ϵ_1, ϵ_2)-pseudo solution $\mathbf{q}$, with their respective controls.

$\mathbf{q}^*$ for any ϵ-pseudo trajectory $\tilde{\mathbf{q}}$ and any time interval if the dynamical system (11.17) is hyperbolic [150, 151]. While we do not know whether the examples of Chapter 13 are hyperbolic in the required strict sense, it has been supported by a chaotic hypothesis that systems with more state-space complexity may behave more hyperbolically [8, 154, 155]. Even for non-hyperbolic systems, a shadowing trajectory may exist for a finite time [39, 152, 153].

So a full validation, of course, depends on the existence of $\mathbf{q}^*$. However, it is likely impossible to present such a solution, and its existence has been proven only implicitly [39, 152, 153]. Thus we restrict the question with respect to the 'numerically continuous' solution: Does a numerical solution with $\|\Delta\mathbf{q}_k\| = 0$ (ϵ-pseudo solution in Definition 14.1) shadow the multi-point optimized solution $\mathbf{q}$ and $\mathbf{\Theta}$? For this, we re-define shadowing in Definition 14.1 for the multi-point optimized solution.

Definition 14.2 (shadowing for multi-point optimized solution)**.** *Let $\tilde{\mathbf{q}} \in \mathbb{Q}$ and $\tilde{\mathbf{\Theta}} \in \mathbb{T}$ be an ϵ_1-pseudo solution of (11.17), so*

$$\left\| \frac{\partial \tilde{\mathbf{q}}(t)}{\partial t} - \mathcal{R}[\tilde{\mathbf{q}}(t), \tilde{\mathbf{\Theta}}(t)] \right\|_{\mathbb{Q}^+} < \epsilon_1 \qquad \forall t \in [t_i, t_f], \tag{14.35}$$

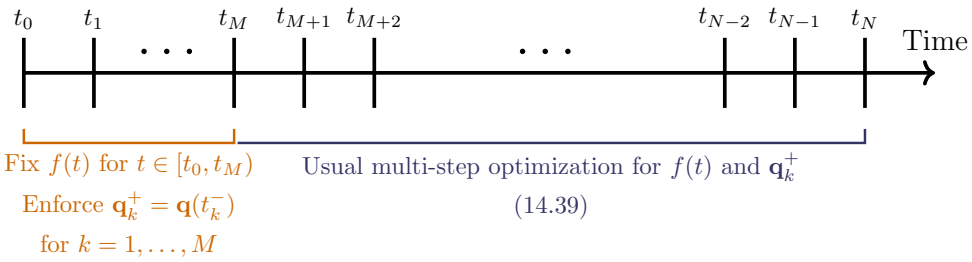


Figure 14.9: Schematic diagram of the subproblem (14.39) for the shadowing trajectory construction (Algorithm 8).

and let $\mathbf{q} \in \mathbb{Q}$ *and* $\boldsymbol{\Theta} \in \mathbb{T}$ *be a multi-point* (ϵ_1, ϵ_2)*-pseudo solution of (11.17), so*

$$\left\|\mathbf{q}_k^+ - \mathbf{q}(t_k^-)\right\|_{\mathbb{Q}^+} < \epsilon_2 \qquad \forall t \in T_{\text{int}} \equiv \{t_1, t_2, \ldots, t_{n-1}\} \tag{14.36a}$$

and

$$\left\|\frac{\partial \mathbf{q}(t)}{\partial t} - \mathcal{R}[\mathbf{q}(t)]\right\|_{\mathbb{Q}^+} < \epsilon_1 \qquad \forall t \in [t_i, t_f]/T_{\text{int}}. \tag{14.36b}$$

Both $\tilde{\mathbf{q}}$ *and* $\mathbf{q}$ *have the same initial condition, so* $\tilde{\mathbf{q}}(t_i) = \mathbf{q}(t_i)$*. The* ϵ_1*-pseudo solution* $\tilde{\mathbf{q}}$ *and* $\tilde{\boldsymbol{\Theta}}$ *δ-shadows the multi-point solution* $\mathbf{q}$ *and* $\boldsymbol{\Theta}$ *for* $t \in [t_i, t_f]$ *if*

$$\|\tilde{\mathbf{q}}(t) - \mathbf{q}(t)\|_{\mathbb{Q}^+} < \delta \qquad \left\|\tilde{\boldsymbol{\Theta}}(t) - \boldsymbol{\Theta}(t)\right\|_{\mathbb{T}^+} < \delta \qquad \forall t \in [t_i, t_f]. \tag{14.37}$$

For our present considerations, ϵ_1 in Definition 14.2 represents corresponds to discretization and finite-precision error. Similarly, ϵ_2 corresponds to the intermediate discontinuities $\{\Delta\mathbf{q}_k\}$. Figure 14.8 (b) illustrates a ϵ_1-pseudo solution $\tilde{\mathbf{q}}$ that shadows the optimized (ϵ_1, ϵ_2)-pseudo solution $\mathbf{q}$. Like Definition 14.1, it is likely impossible to prove its existence analytically, though it is possible to numerically construct a viable $\tilde{\mathbf{q}}$ and $\tilde{\boldsymbol{\Theta}}$. The existence of $\tilde{\mathbf{q}}$ then suggests that the multi-point solution $\mathbf{q}$ approximates an exact trajectory $\mathbf{q}^*$, just as $\tilde{\mathbf{q}}$ does. We introduce a systematic procedure for this. This is in some sense the refinement procedure of Grebogi *et al.* [152], though the specific steps are different.

To find a shadowing $\tilde{\mathbf{q}}$ with $\Delta\mathbf{q}_k = 0$, we continue the optimization with the framework introduced in Section 14.3, but sequentially enforces the intermediate constraint (14.15c) at increasingly many times t_k. Figure 14.9 shows a schematic for this sequential enforcement. The procedure starts from the optimized solution, for which all $\{\Delta\mathbf{q}_k\}$ are small but finite. The optimization period is shortened by strictly enforcing

(14.15c) for the first $M \leq N-1$ time points, so the governing equation (14.15) becomes

$$\frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \boldsymbol{\Theta}] = \mathbf{0} \qquad \text{for } t \in [t_i, t_M) \tag{14.38a}$$

$$\mathbf{q}(t_i) = \mathbf{q}_0, \tag{14.38b}$$

with $\mathbf{q}_k^+$ used for $t_k \geq t_{M+1}$,

$$\frac{d\mathbf{q}}{dt} - \mathcal{R}[\mathbf{q}; \boldsymbol{\Theta}] = \mathbf{0} \qquad \text{for } t \in [t_k, t_{k+1}) \tag{14.38c}$$

$$\mathbf{q}(t_M) = \mathbf{q}(t_M^-) \tag{14.38d}$$

$$\mathbf{q}(t_k) = \mathbf{q}_k^+ \qquad \text{for } k = M+1, \ldots, N-1. \tag{14.38e}$$

Hence, the violation of (14.15c) is considered only for $t_k \geq t_{M+1}$,

$$\Delta\mathbf{q}_k \equiv \mathbf{q}_k^+ - \mathbf{q}(t_k^-) = 0 \qquad \text{for } k = M+1, \ldots, N-1. \tag{14.38f}$$

We continue to use Algorithms 6 and 7 to minimize $\mathcal{J}_A$ (14.19), but exclude all optimization variables for $t < t_M$, so the subproblem (14.20) is modified as

$$\left({}_i\{\mathbf{q}_k^+\}_{M+1}^{N-1}, {}_i\boldsymbol{\Theta}(t_M \leq t \leq t_f)\right) = \operatorname{argmin} \mathcal{J}_A[\mathbf{q}, \boldsymbol{\Theta}; \{\mathbf{q}_k^+\}_{M+1}^{N-1}, {}_i\mu], \qquad \text{with (14.38c-e)}, \tag{14.39}$$

where the solution for $t \in [t_i, t_M)$ under (14.38a) does not change in the optimization. For $t \in [t_M, t_f]$, the standard multi-step optimization is applied. Repeating the optimization of (14.39) from $M = 1$ to $N-1$ enforces (14.15c) for all $k = 1, 2, \ldots, N-1$, which constructs a shadowing trajectory $\tilde{\mathbf{q}}$ in Definition 14.2. Algorithm 8 summarizes the overall construction procedure.

Algorithm 8 Shadowing trajectory construction

Given: the multi-point optimized solution ${}_0\{\mathbf{q}_k^+\}$, ${}_0\boldsymbol{\Theta}$, ${}_1\mu$
for $M = 1, 2, \ldots, N-1$ **do**
  for $k = 1, \ldots, M$ **do**
    ${}_M\mathbf{q}_k^+ = {}_{M-1}\mathbf{q}(t_k^-)$
  Subproblem (14.39) with initial guess ${}_{M-1}\{\mathbf{q}_k^+\}$, ${}_{M-1}\boldsymbol{\Theta}$ ▷ Optimize only within $t \in [t_M, t_N]$
  ${}_{M+1}\mu = {}_M\mu$ ▷ Needs not increase

A shadowing solution $\tilde{\mathbf{q}}_1$ is constructed from the optimized $\mathbf{q}$ of Section 14.4. To start the procedure, an initial solution $\mathbf{q}_1$ is taken to be the optimized solution after 460 searches, as shown in Figure 14.10 (a). $\tilde{\mathbf{q}}_1$ is

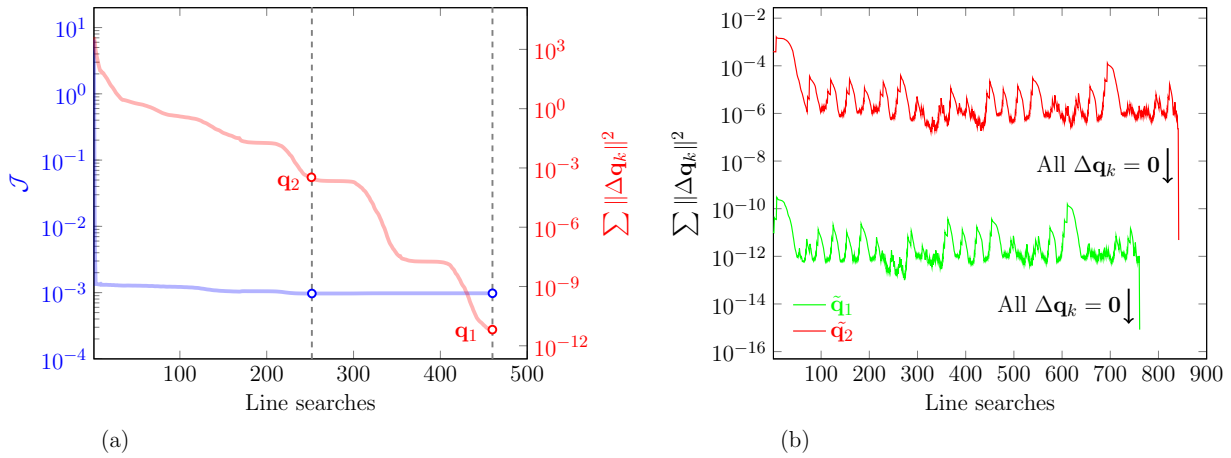
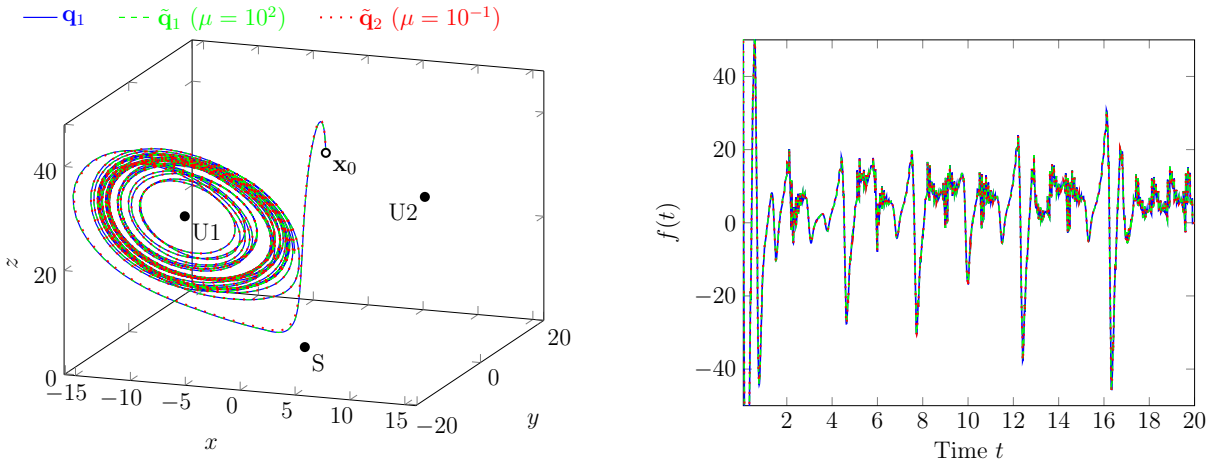


Figure 14.10: Construction of ϵ_1-pseudo trajectories $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$ of the Lorenz system (10.1) from the multi-point optimized solutions. (a) Initial solutions for Algorithm 8 chosen from the multi-point optimization. (b) Total discontinuity throughout the procedure of Algorithm 8. In the end, the discontinuity strictly becomes zero, as all the intermediate constraints are enforced.

constructed via Algorithm 8 with the penalty strength $\mu = 10^2$. Figure 14.10 (b) shows the total discontinuity of $\tilde{\mathbf{q}}_1$ under the construction procedure, in which $\sum \|\Delta \mathbf{q}_k\|^2$ remains at $\approx 10^{-11}$ until $\mathbf{q}_k^+ = \mathbf{q}(t_k^-)$ for all k, then the discontinuity is reduced directly to zero. $\tilde{\mathbf{q}}_1$ has almost the same $\mathcal{J} = 9.7722578 \times 10^{-4}$ to $\mathcal{J} = 9.7722579 \times 10^{-4}$ of the optimized solution.

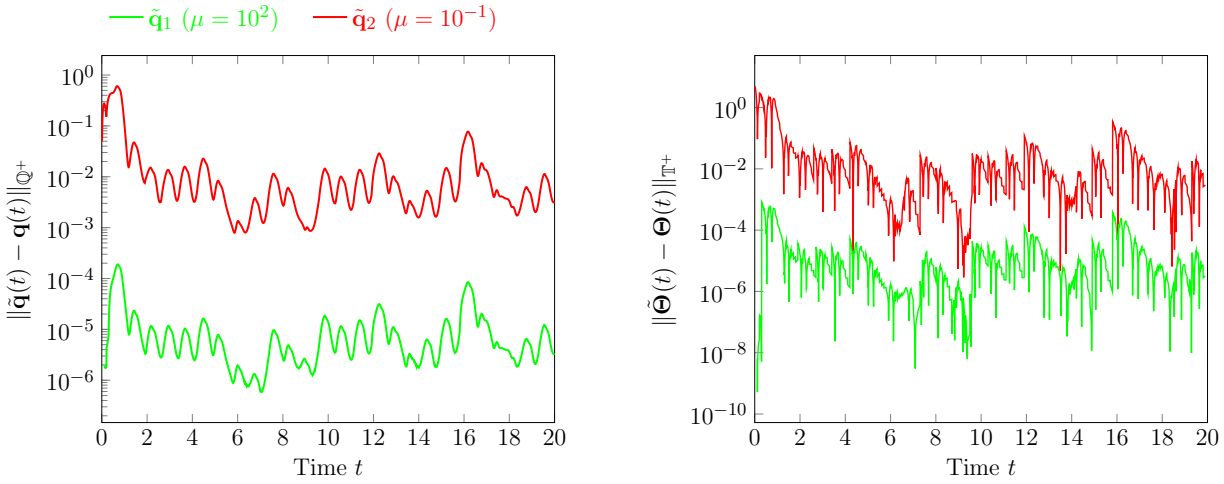
To see whether the construction procedure is sensitive to the discontinuity of initial solution, we constructed another shadowing solution $\tilde{\mathbf{q}}_2$ with a less optimized solution. An initial solution $\mathbf{q}_2$ is taken to be the optimized solution after 252 searches in Figure 14.10 (a). It has a larger discontinuity than $\mathbf{q}_1$ by about a factor of 10^8. $\tilde{\mathbf{q}}_2$ is constructed with weaker $\mu = 10^{-1}$. During the procedure, the total discontinuity of $\tilde{\mathbf{q}}_2$ remains at $\approx 10^{-5}$ larger than $\tilde{\mathbf{q}}_1$, though it is also reduced directly to zero at the end. Despite a larger discontinuity, $\tilde{\mathbf{q}}_2$ still has $\mathcal{J} = 9.695 \times 10^{-4}$ similar to the optimized solution. Shadowing solutions $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$ are compared with the optimized solution $\mathbf{q}_1$ in their trajectory $\mathbf{q}(t)$ and forcing $f(t)$ in Figure 14.11. All of them appear to overlay each other, for both $\mathbf{q}$ and $\mathbf{f}$. The shadowing distances of $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$ in (14.37), quantified in Figure 14.12, are bounded at all times, not diverging as in Figure 14.7 (b). With its μ and smaller $\{\Delta \mathbf{q}_k\}$ (Figure 14.10 b), $\tilde{\mathbf{q}}_1$ shadows closer (smaller δ in Definition 14.2). These results suggest that a multi-point solution, even with a moderately reduced discontinuity, can approximate a (numerically) continuous solution.



(a) State trajectory

(b) Control forcing

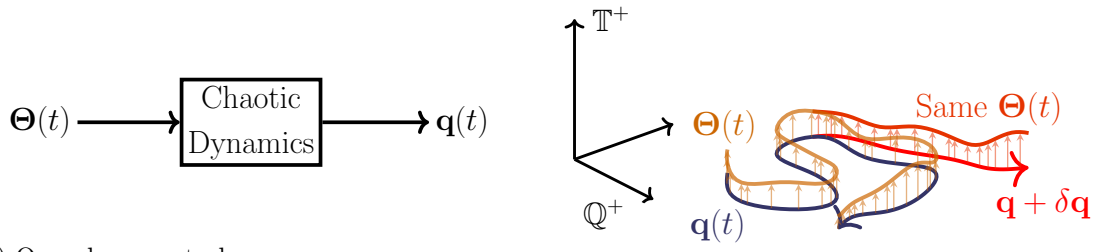
Figure 14.11: Comparison of $\mathbf{q}_1$ with $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$ in (a) their state trajectories, and (b) their controls.



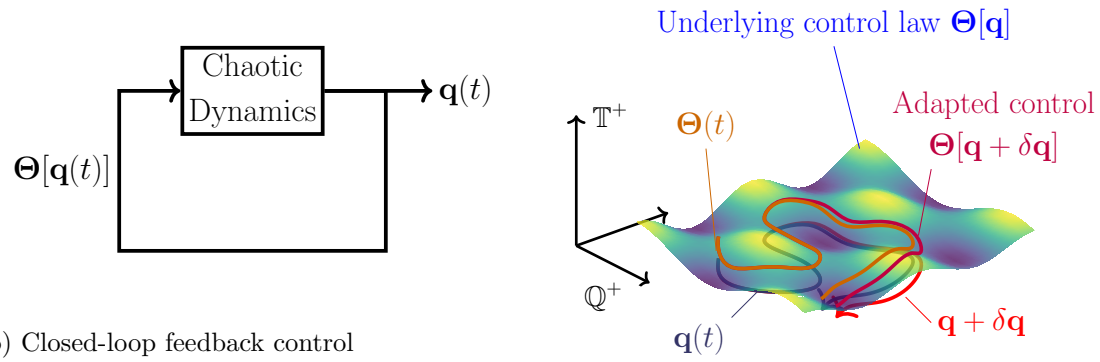
(a)

(b)

Figure 14.12: The shadowing distance of $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$ with respect to $\mathbf{q}_1$ in (a) their states, and (b) their controls.



(a) Open-loop control



(b) Closed-loop feedback control

Figure 14.13: A schematic illustration for the stability of the optimized control depending on its format: (a) for an open-loop control (left), the solution is susceptible to deviations (right); (b) for a closed-loop control (left), the control can adapt to deviations (right).

14.6 Utility of the optimized solution and the burden of interpretation

The results in Section 14.5 may seem contradictory in that the optimized solution is sensitive to small discontinuities yet at the same time shadows an actual continuous trajectory. One may even question the utility of the optimized solution: if it requires such extremely fine resolution to have the expected performance, is the control usefully robust? or even, is the system controllable in practice? As discussed in the previous section, this concern underlies any numerical simulations of chaotic dynamical systems, making the optimized solution useful only for some tasks (e.g, computation of turbulence statistics) but not all (e.g, exactly tracking dynamics).

There is, however, another perspective, which exposes additional utility. From the control system theory viewpoint, the optimized $\mathbf{q}(t)$ and $\boldsymbol{\Theta}(t)$ solution is given in a format of open-loop control, where there is no $\boldsymbol{\Theta}$ compensation regardless of how $\mathbf{q}$ deviates from the desired output [156, 157]. This is illustrated in Figure 14.13 (a). Many such controls are well understood to be highly sensitive to disturbances and errors, and only effective when the underlying dynamical system is stable [156].

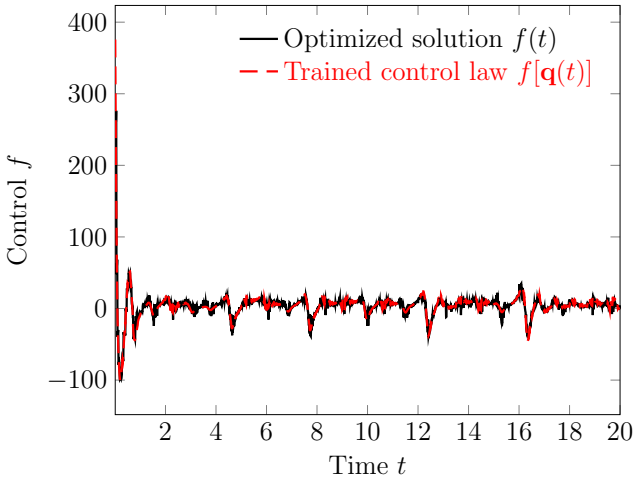


Figure 14.14: The time history of actuation f from the optimized solution and the trained regression tree.

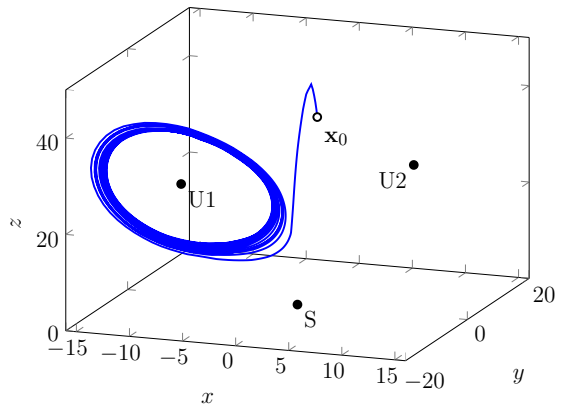
However, as one instance of the control law $\mathbf{\Theta}[\mathbf{q}]$, it retails utility to advance knowledge about an effective feedback control law $\mathbf{\Theta}(t) = \mathbf{\Theta}[\mathbf{q}(t)]$, as is illustrated in Figure 14.13 (b). In this sense, the failure shown in Figure 14.7 is only a consequence of passively applying the control in an open-loop fashion.

We demonstrate an approach for this Lorenz example by simply fitting $\mathbf{q}(t)$ and $\mathbf{\Theta}(t)$ to find a closed-loop control that is robust. The optimized solution $\mathbf{q}$ and $\mathbf{\Theta}$ in $t \in [0, 20]$ has 2000 time steps, which are used as 2000 training samples of $(\mathbf{q}(t), f(t))$. A regression tree, as implemented in the MATLAB Toolbox [158], is used as a model function with minimum leaf size 4, leading to a tree of 685 nodes. The trained regression tree has

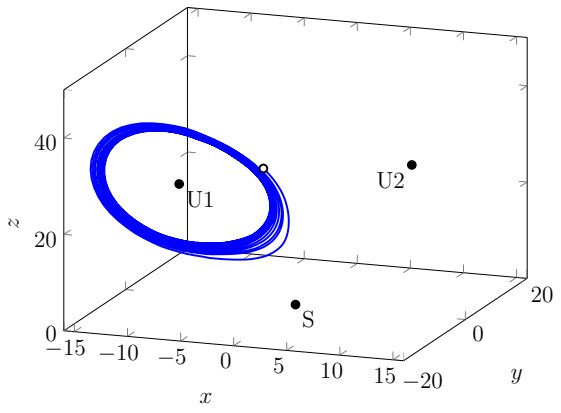
$$R^2 = 1 - \frac{\sum_{k=1}^{2000} [f_k - f(\mathbf{q}_k)]^2}{\sum_{k=1}^{2000} (f_k - \overline{f})^2} \approx 0.91. \tag{14.40}$$

Figure 14.14 (b) shows the time history of the actuation by the inferred control law, which agrees well with the optimized solution. Figure 14.15 shows the application of the inferred control law for 4 different initial conditions. Only one initial condition shown in Figure 14.15 (a) is used for the training, however all of them are well controlled, even for the initial conditions far from the training data (Figure 14.15 c and d).

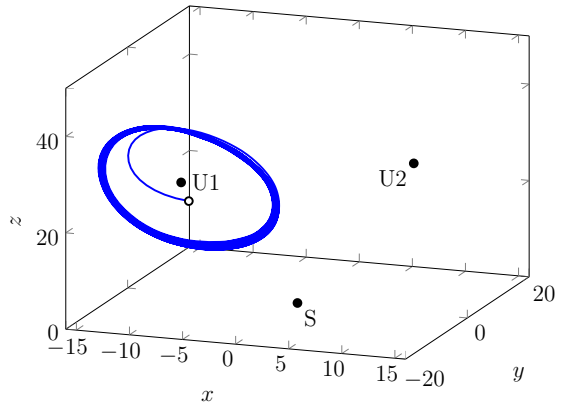
In this example, the optimization procedure is useful, not because it can be used directly for an actual control, but because it provides a valuable training data. Similarly, it is often not known whether a flow is controllable to the desired flow state. In such situations, the control found by the optimization also exposes the controllability of the flow system, in addition to providing a pathway to analyze and harness its mechanism. It is also anticipated that human-learning is possible such as in analysis done by Wei and Freund [78] for the optimized control of two-dimensional shear layer noise.



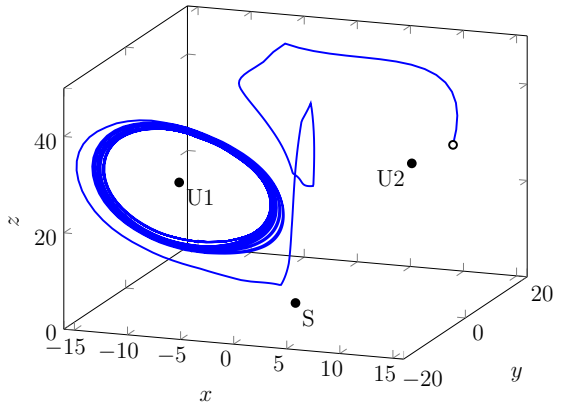
(a) Original initial condition $\mathbf{x}_0 = (1.49, 1.49, 37)^T$



(b) $\mathbf{x}_0 = (-3.67, 0.78, 28.03)^T$



(c) $\mathbf{x}_0 = (-3.49, -18.49, 28)^T$



(d) $\mathbf{x}_0 = (10, 15, 25)^T$

Figure 14.15: Controlled solutions by the inferred control law starting from 4 different initial conditions.

Chapter 15

Optimal control of chaotic advective systems and turbulent flow

The multi-point optimization framework proposed in Chapter 14 is applied to the control problems of Chapter 13: the Kuramoto–Sivashinsky Equation, two-dimensional Kolmogorov flow, the Adv+KS system, and three-dimensional Kolmogorov turbulence.

15.1 Kuramoto–Sivashinsky equation

To apply the multi-point method to this control problem as introduced in Section 13.1.4, the full simulation time $t_f - t_0 = 5$ is split into $n = 125$ intervals of $T = 0.04$. Based on the e-folding time of this system $t_\lambda \approx 0.412$ (Section 13.1), the sensitivity is therefore expected to be amplified a factor of only 1.1 within each interval. The quadratic penalty method with Algorithm 6 is applied for the first 6 subproblems, and the augmented Lagrangian method with Algorithm 7 is applied from the 7th subproblem and onward. A control is found to reduce $\mathcal{J}$ (13.5) by 99.7%, whereas the standard gradient-based approach from Section 13.1 only achieved a 22.4% reduction. The controlled solution is shown in Figure 15.1.

Figure 15.2 (a) illustrates how the multi-point optimization framework finds its minimum (13.5). In most subproblem minimizing $\mathcal{J}_A$, the actual objective $\mathcal{J}$ initially rises to a larger value before decreasing more gradually, eventually approaching a local optimum for the specified μ. The intermediate discontinuities $\|\Delta u_k\|$ decrease monotonically with each line search. Figure 15.2 (b) confirms, as anticipated by design of the method, that the multi-point framework takes larger $\boldsymbol{\Theta}$ steps, and that the search scale reduces as μ increases. Figure 15.2 (c) and (d) show that the control found by the multi-point method is no longer biased to early times due to the gradient amplification over the full time horizon. Similarly, the control amplitude is not restricted to low-amplitude as in Section 12.1.4. It is active at all times and reduces $\mathcal{I}(t)$ throughout the simulation except for times $t < \tau_A \approx 0.25$.

In this demonstration, the penalty strength μ is increased by a factor of 4 for each quadratic penalty subproblem, and by a factor of 2 for each augmented Lagrangian subproblem, as shown in Figure 15.3 (a).

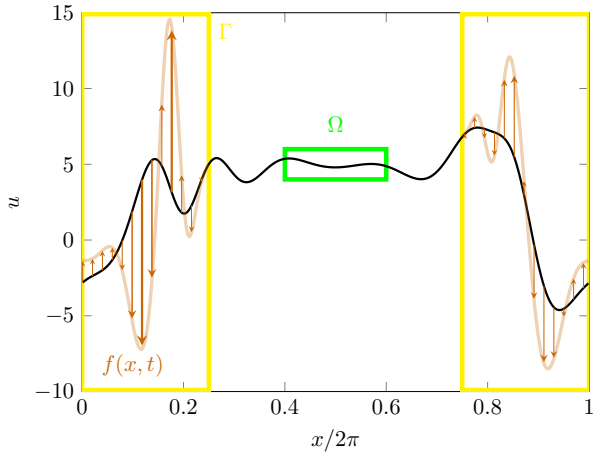


Figure 15.1: Controlled solution (black) at $t = 2.5$ of K–S equation. The control is visualized as orange arrows, with a scaled magnitude $0.05f(x,t)$.

Each subproblem is solved until the gradient magnitude decreases below a relative threshold,

$$\frac{\|\nabla \mathcal{L}_A\|^2}{\|\nabla \mathcal{L}_{A,1}\|^2} < {}_i\epsilon. \tag{15.1}$$

with $\|\nabla \mathcal{L}_A\|^2 = \|\nabla_{\mathbf{\Theta}} \mathcal{L}_A\|_{\mathbb{T}}^2 + \sum_{k=1}^{n-1} \|\nabla_{\mathbf{q}_k^+} \mathcal{L}_A\|_{\mathbb{Q}^+}^2$ and $\|\nabla \mathcal{L}_{A,1}\|^2$ the gradient for the first line search of the subproblem. The threshold is set ${}_i\epsilon = 0.1$ for $i \leq 8$ and ${}_i\epsilon = 0.01$ for $i > 8$. The gradient magnitude throughout the optimization is shown in Figure 15.3 (b). There is no formal criterion known yet to terminate a subproblem, only having a loose condition $\lim_{i\to\infty} {}_i\epsilon = 0$ [57, 58]. For our examples, keeping this relative threshold is sufficient and the results are not sensitive to ${}_i\epsilon$. Various precisions for the line search ($\alpha_3 - \alpha_1$ with values used in (11.16)) were used to decrease the gradient threshold, however the gradient magnitude was also insensitive to them.

For the final iterations shown in Figure 15.2 (a), we deem the optimization to be sufficient, because the $\|\Delta u_k\|_{\mathbb{Q}^+}$ keeps decreasing without significantly changing $\mathcal{J}$. We consider the validity of the optimized solution in the shadowing sense as we did for the Lorenz example in Section 14.5. Two shadowing solutions by Definition 14.2 are constructed by Algorithm 8. First shadowing solution $\tilde{\mathbf{q}}_1$ is constructed from the optimized solution $\mathbf{q}_1$ (Figure 15.4 a) via Algorithm 8 with $\mu = 800$. Another shadowing solution $\tilde{\mathbf{q}}_2$ is constructed from a less-optimized solution $\mathbf{q}_2$ (Figure 15.4 a) with $\mu = 25$, to confirm the insensitivity to Δu_k as we did for the Lorenz example. Both are constructed with augmented Lagrangian method, with μ and $\{u_k^{\dagger+}\}$ constant. Figure 15.4 (b) shows that the total discontinuity of $\tilde{\mathbf{q}}_2$ remains higher than $\tilde{\mathbf{q}}_1$, though both have $\|\Delta u_k\|_{\mathbb{Q}^+} = 0$ for all k at the end of the procedure. Though it takes many about 10^4 searches to finish the construction procedure, their $\mathcal{J}$'s only slightly change from the optimized solution $\mathbf{q}_1$:

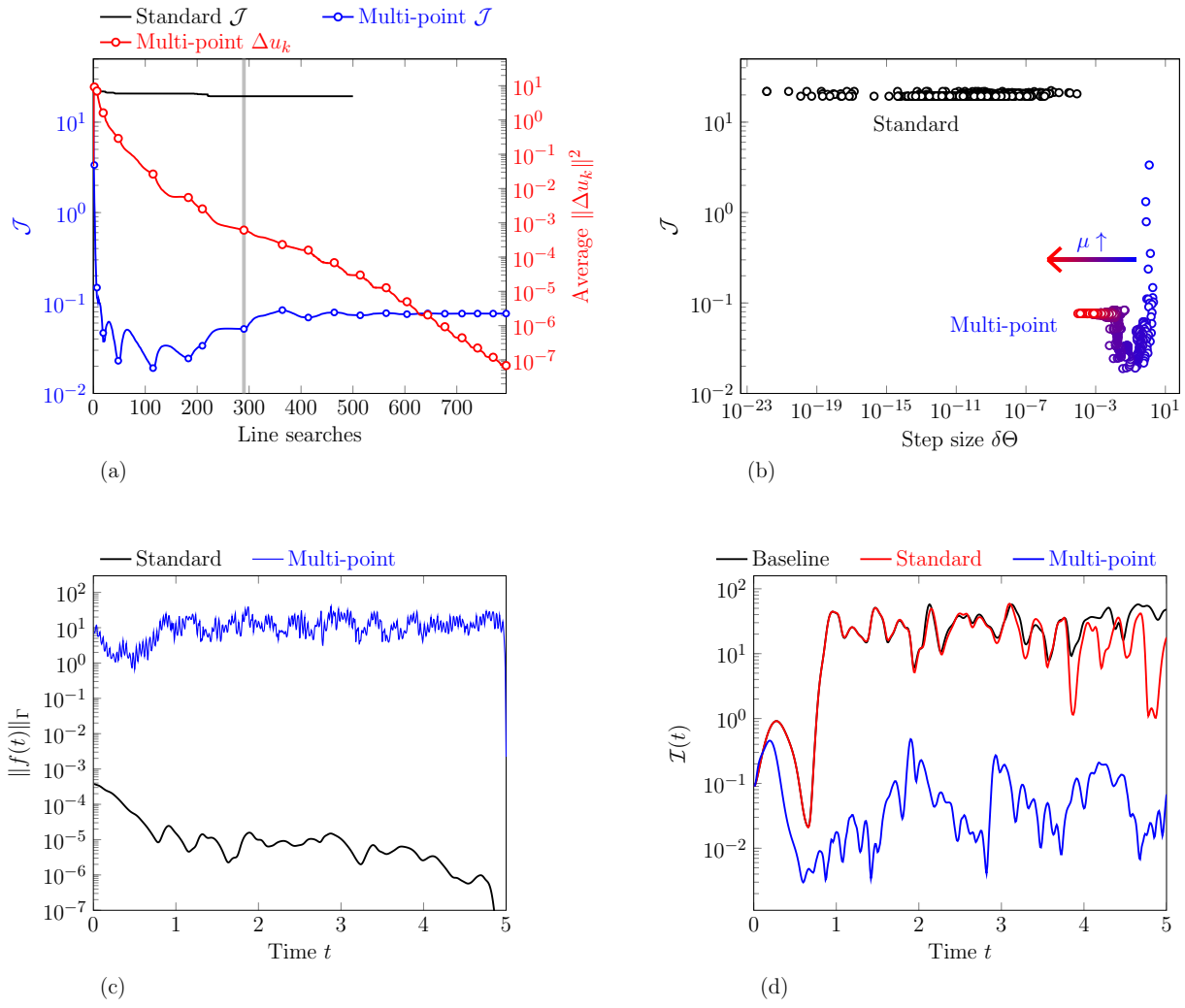


Figure 15.2: Optimization result for K–S equation. (a) Reduction of $\mathcal{J}$ (13.5) and average $\|\Delta u_k\|^2$, with markers denoting penalty strength updates in Algorithm 6 and 7. The transition from quadratic penalty method to the augmented Lagrangian method is marked with gray line. (b) The search step sizes in optimization procedure. For the multi-point method, steps are marked from blue to red with increasing μ. (c) The control strength of the optimized controls by standard gradient-based method and multi-point method. (d) The instantaneous objective functional $\mathcal{I}(t)$ (13.5b) of the baseline solution and the controlled solutions.

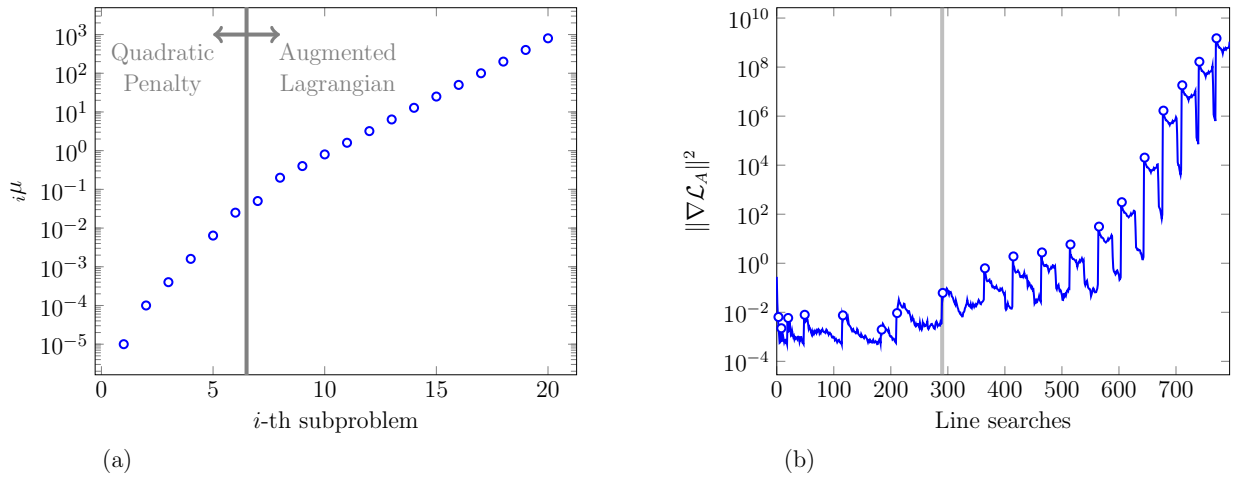


Figure 15.3: (a) Updated penalty strength $_i\mu$ at the i-th subproblem. (b) Gradient magnitude of augmented Lagrangians in Algorithm 6 and Algorithm 7, with markers denoting penalty strength updates. The gray line indicates the transition from the quadratic penalty method to the augmented Lagrangian method.

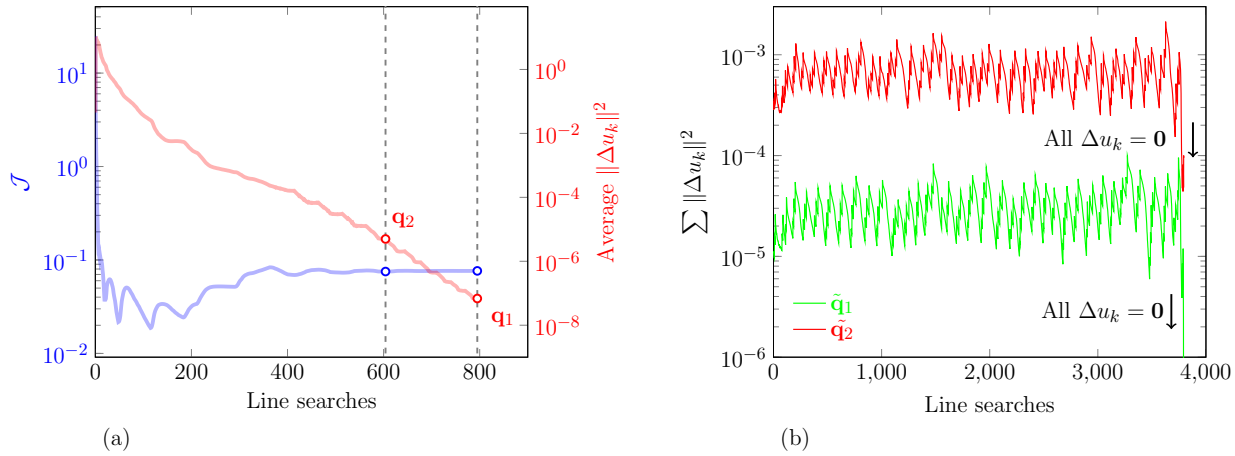


Figure 15.4: Construction of shadowing trajectories $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$ of the K–S equation from the multi-point optimized solutions. (a) Initial solutions for Algorithm 8 chosen from the multi-point optimization. (b) Total discontinuity throughout the procedure of Algorithm 8. In the end, $\|\Delta u_k\|_{\mathbb{Q}^+} = 0$, as all the intermediate constraints are enforced.

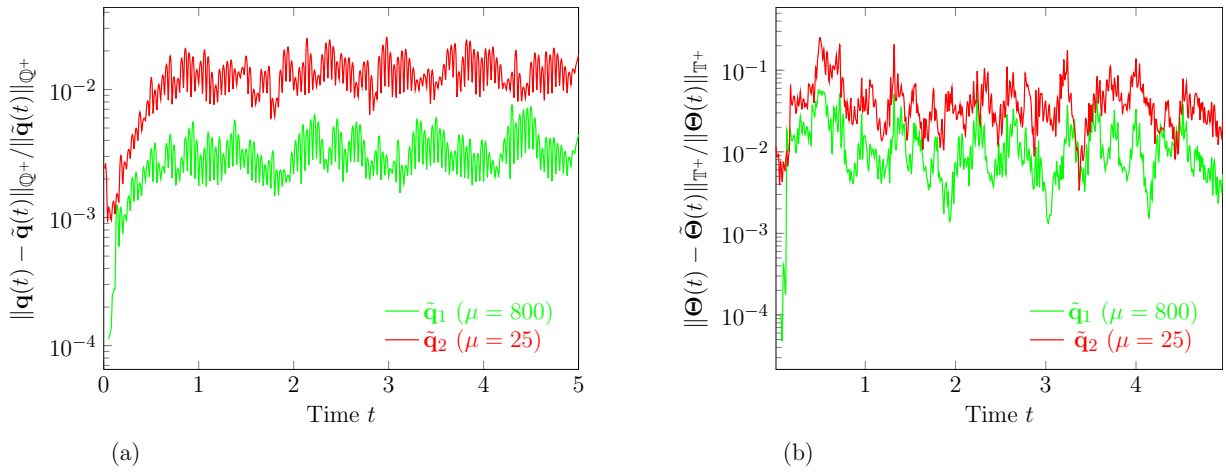


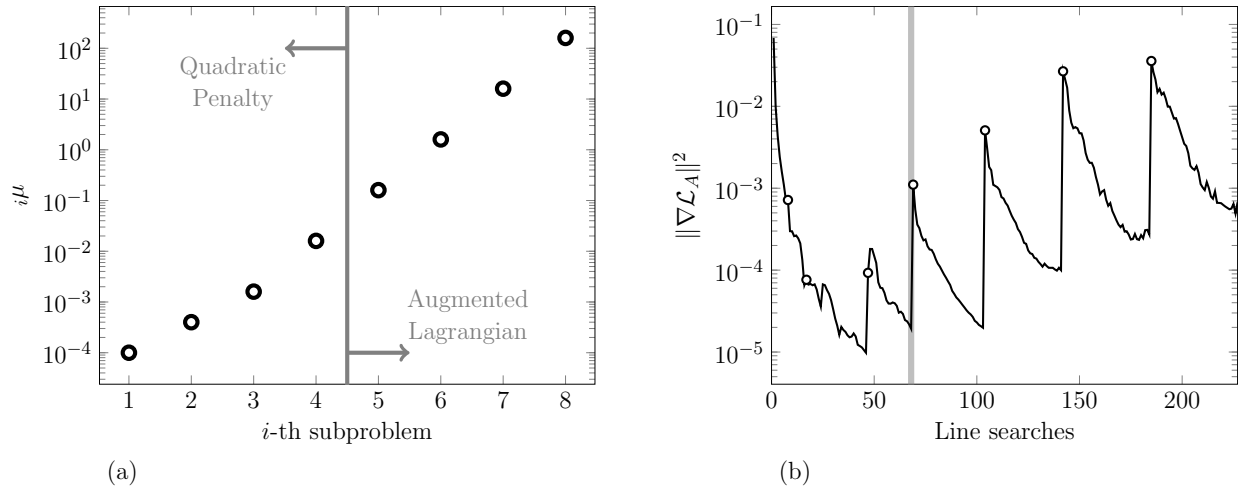
Figure 15.5: The shadowing distance of $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$ with respect to $\mathbf{q}_1$ in (a) their states, and (b) their control strengths.

$\mathcal{J} = 7.64 \times 10^{-2}$ for $\mathbf{q}_1$, $\mathcal{J} = 7.60 \times 10^{-2}$ for $\tilde{\mathbf{q}}_1$, and $\mathcal{J} = 8.13 \times 10^{-2}$ for $\tilde{\mathbf{q}}_2$. The shadowing distances of $\tilde{\mathbf{q}}_1$ and $\tilde{\mathbf{q}}_2$, quantified in Figure 15.5, are bounded at all times for both $\mathbf{q}$ and $\mathbf{\Theta}$. This again supports that the optimized solution, though it includes $\|\Delta u_k\|_{\mathbb{Q}^+} \lesssim 10^{-3}$, does approximate a continuous-in-time solution with similar $\mathcal{J}$.

Although this exhaustive approach we have used supports consistent convergence of the optimization for both the Lorenz system and the K–S equation, doing it for the following cases seem prohibitive, since these are significantly more costly simulations and are also anticipated to require still more line searches.

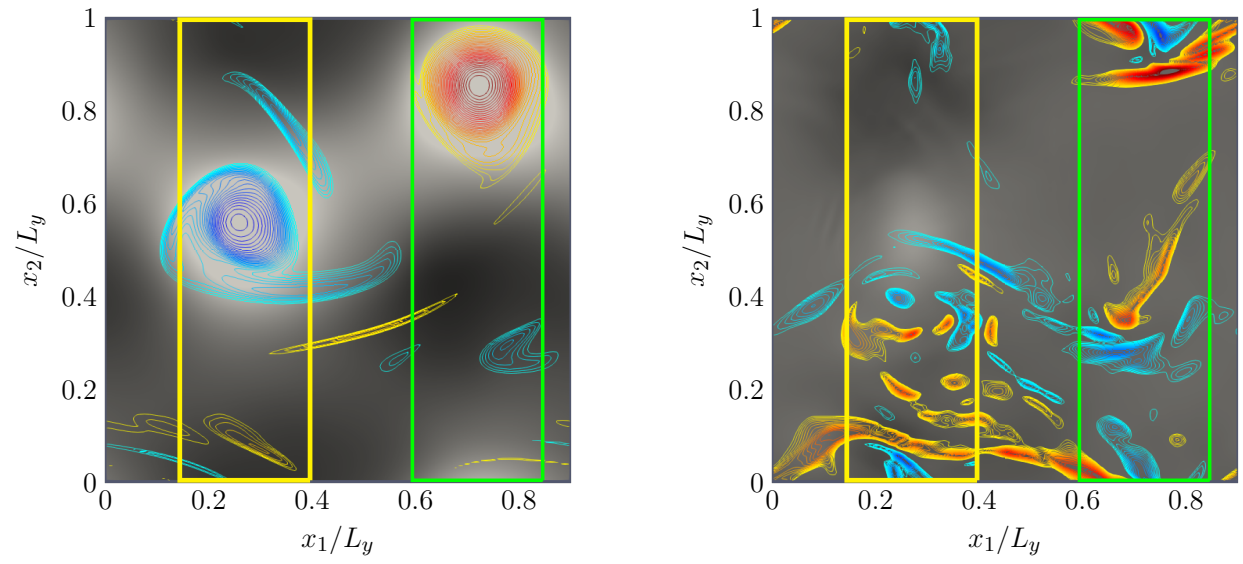
15.2 Two-dimensional Kolmogorov flow

For this flow, which was introduced in Section 13.2.3, the time $t_f - t_0 = 4.47\tau_c$ is split into $N = 40$ intervals of $T = 0.11\tau_c$. Based on the inferred e-folding time of this system, the sensitivity is anticipated to be amplified by a factor of 1.13 within each interval. For the results we consider, the quadratic penalty method with Algorithm 6 is applied for the first 4 subproblems, and the augmented Lagrangian method with Algorithm 7 is applied from the 5th subproblem and onward. The penalty strength μ is increased by a factor of 4 for the first 3 subproblems and then by a factor of 10 thereafter, as shown in Figure 15.6 (a). Similar to the K–S equation, each subproblem is solved until the relative gradient (15.1) decreases below a threshold: $\epsilon_i = 0.1$ for $i \leq 3$ and $\epsilon_i = 0.01$ for $i > 3$. The gradient magnitude throughout the optimization is shown in Figure 15.6 (b). Figure 15.7 shows that the control found significantly disrupts the vortices of the baseline simulation, preventing them from creating the pronounced dynamic pressure of the baseline flow.



(a) (b)

Figure 15.6: Two-dimensional Kolmogorov flow: (a) the penalty strength μ each update; and (b) gradient magnitude of the augmented Lagrangian. Circles denote the starting points of the subproblems.



(a) Baseline (b) Controlled

Figure 15.7: Pressure $p/p_0 \in [0.9, 1.1]$ (grayscale) and vorticity contours (colors) $\omega\tau_c \in [-44.7, 44.7]$ at $t = 45.6\tau_c$ for (a) the baseline flow and (b) the controlled flow by the multi-point method.

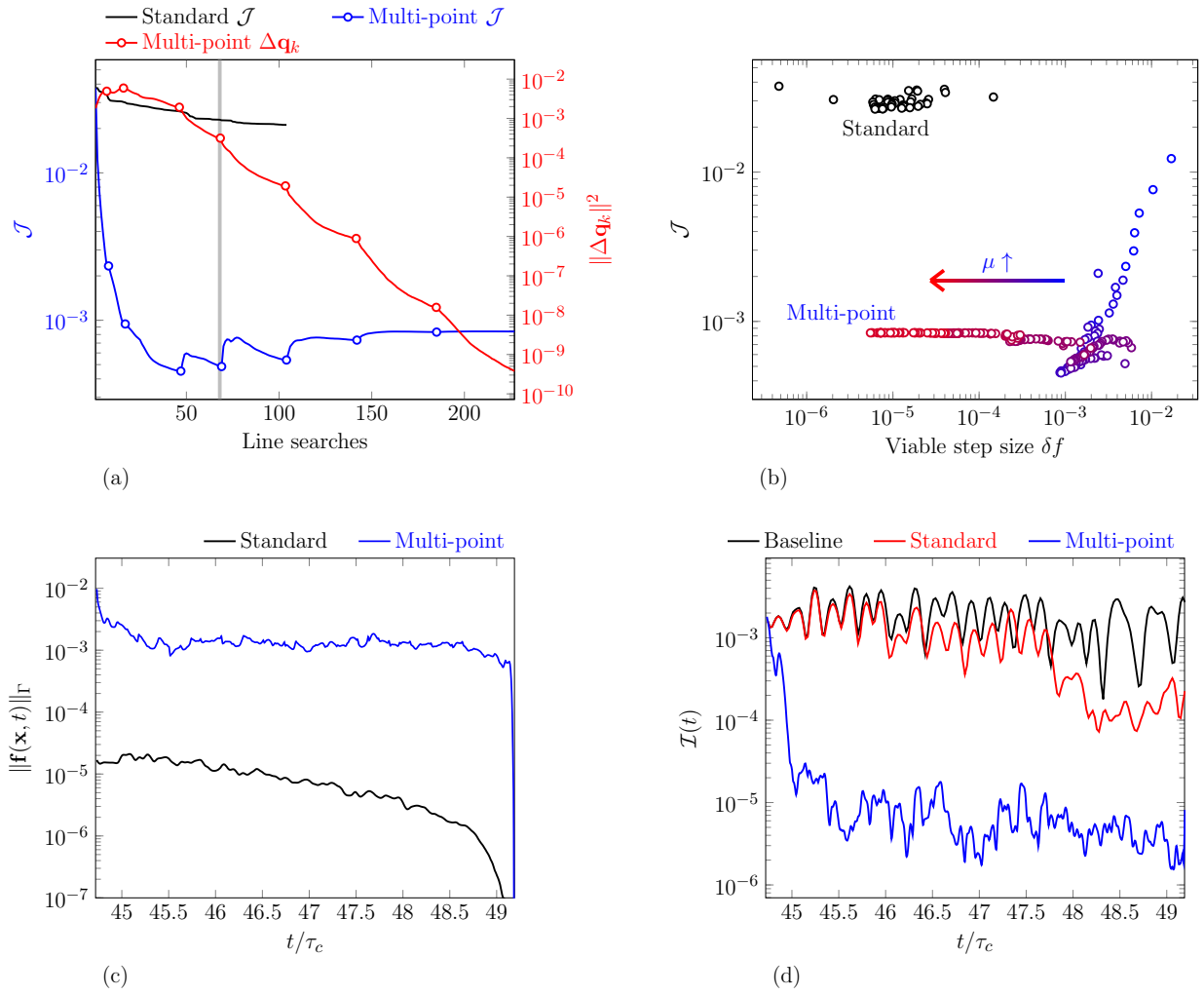


Figure 15.8: Optimization result for the two-dimensional Kolmogorov flow. (a) Reduction of $\mathcal{J}$ (13.14) and average $\|\Delta\mathbf{q}_k\|^2$, with markers denoting penalty strength updates in Algorithm 6 and 7. The transition from quadratic penalty method to the augmented Lagrangian method is marked with gray line. (b) The search steps in optimization procedure. For the multi-point method, steps are marked from blue to red with increasing μ. (c) The control strength of the optimized controls. (d) $\mathcal{I}(t)$ (13.14b) of the baseline solution and the controlled solutions.

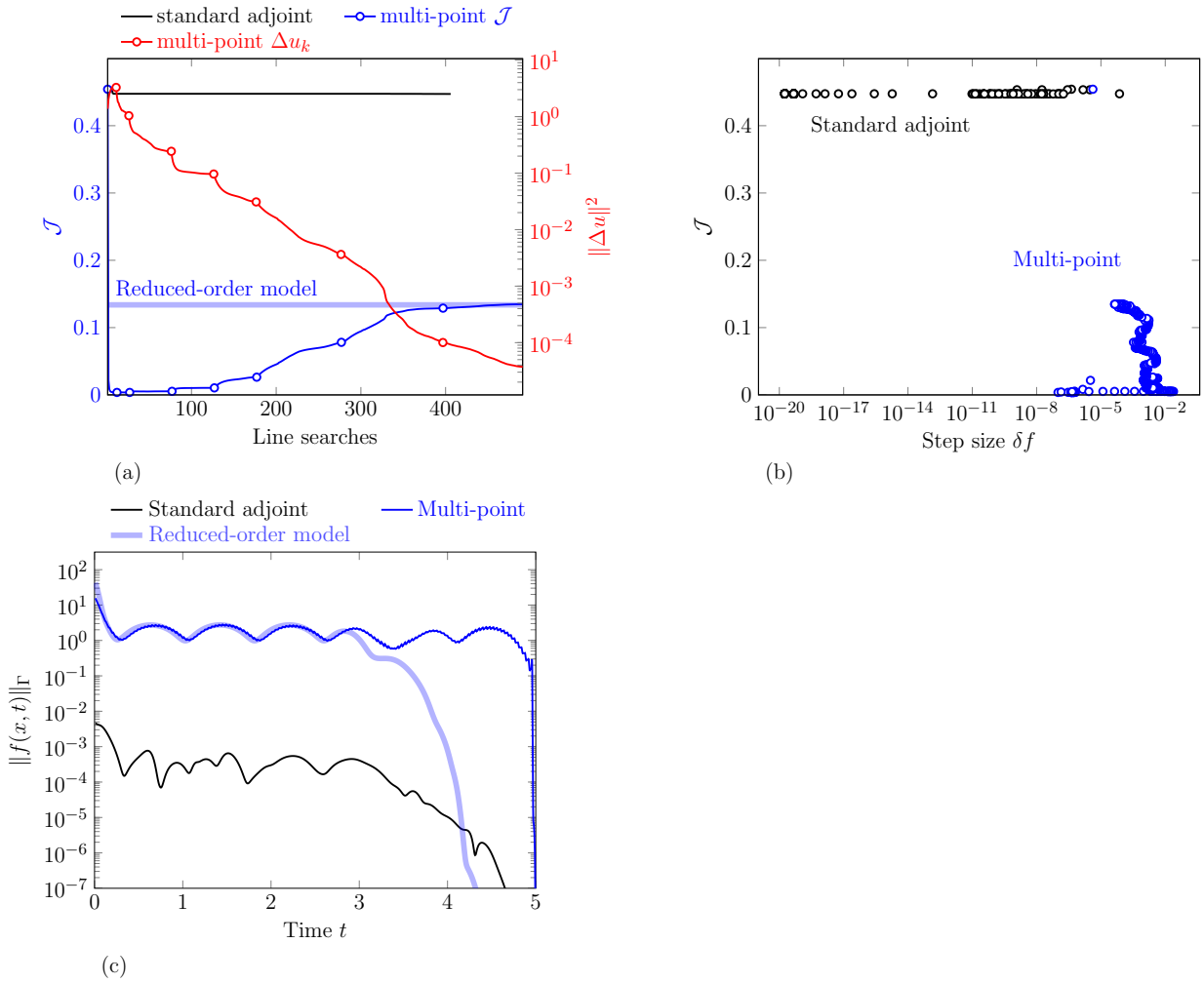


Figure 15.9: The optimization result of the Adv+KS model of Section 13.3. (a) Reduction of $\mathcal{J}$ (13.19) and the discontinuity. Standard gradient-based result from the full dynamics is included for a reference. Circle markers indicate the penalty strength updates. (b) $\mathcal{J}$ and step size of $f(x,t)$ taken in optimization. (c) The control strength of the optimized controls. The control from the reduced model is included for a reference.

Figure 15.8 (a) shows the optimization of the objective functional and intermediate discontinuities, compared with the result by the standard gradient-based method. That method stagnates after a 43.8% $\mathcal{J}$ reduction, whereas the proposed method achieves 97.8% reduction. As for the K–S example, Figure 15.8 (b) suggests that the multi-point framework explores a larger region of $\mathbb{T}$ space. Likewise, the control is active throughout the entire simulation time, as shown in Figure 15.8 (c). Figure 15.8 (d) confirms with $\mathcal{I}(t)$ that the solution is converged to a uniformly effective trajectory.

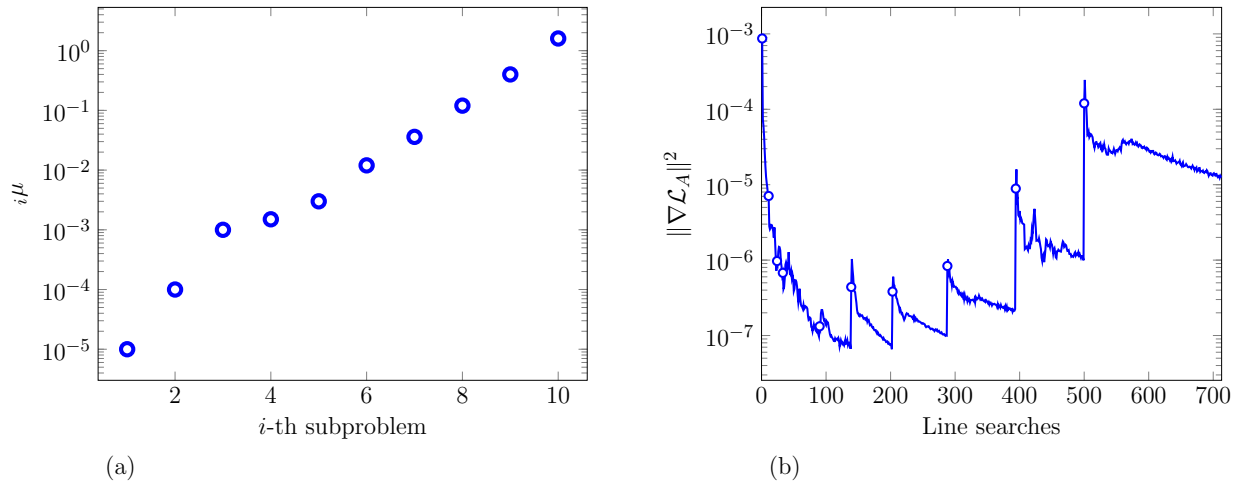


Figure 15.10: Three-dimensional Kolmogorov flow: (a) Penalty strength μ each update; and (b) Gradient magnitude in optimization procedure. Circles denote the starting points of the subproblems.

15.3 Advection+Kuramoto–Sivashinsky model (Adv+KS)

As for Kuramoto–Sivashinsky equation in Section 15.1, the optimization time period is split into 125 intervals. Algorithm 6 is implemented for 8 subproblems with quadratic penalty method. The penalty strength starts from ${}_1\mu = 10^{-4}$, increased by a factor of 10 up to ${}_3\mu = 10^{-2}$, and by a factor of 4 subsequently. Figure 15.9 (a) shows that the multi-point method achieves a similarly effective reduction of $\mathcal{J}$ as the reduced-order model shown in Figure 13.10, while gradually decreasing its $\sum \|\Delta \mathbf{q}_k\|^2$. The step sizes in Figure 15.9 (b) shows that it too seems to avoid poor local minima. In addition, Figure 15.9 (c) shows that the control found recovers the reduced model, without explicit representation of such a model. Whether this behavior can be achieved in a true turbulent flow, for which accurate reduced models are challenging, is assessed for Kolmogorov flow in the next section.

15.4 Three-dimensional Kolmogorov flow

As for the two-dimensional case in Section 15.2, the optimization time period is split into 40 segments of length $T = 0.11\tau_c$. Algorithm 6 is implemented with the quadratic penalty method, for which μ is shown in Figure 15.10 (a). The gradient magnitude throughout the optimization is shown in Figure 15.10 (b).

Figure 15.11 (a) shows that the multi-point framework again finds a better optimum than the standard method (88.3% versus 51.5% reduction), again by seeming to explore more of $\mathbb{T}$ space (Figure 15.11 b). Only the first two line searches of the standard method have comparable step sizes. In Figure 15.11 (c), the control found by the multi-point method is also stronger and distributed throughout the simulation

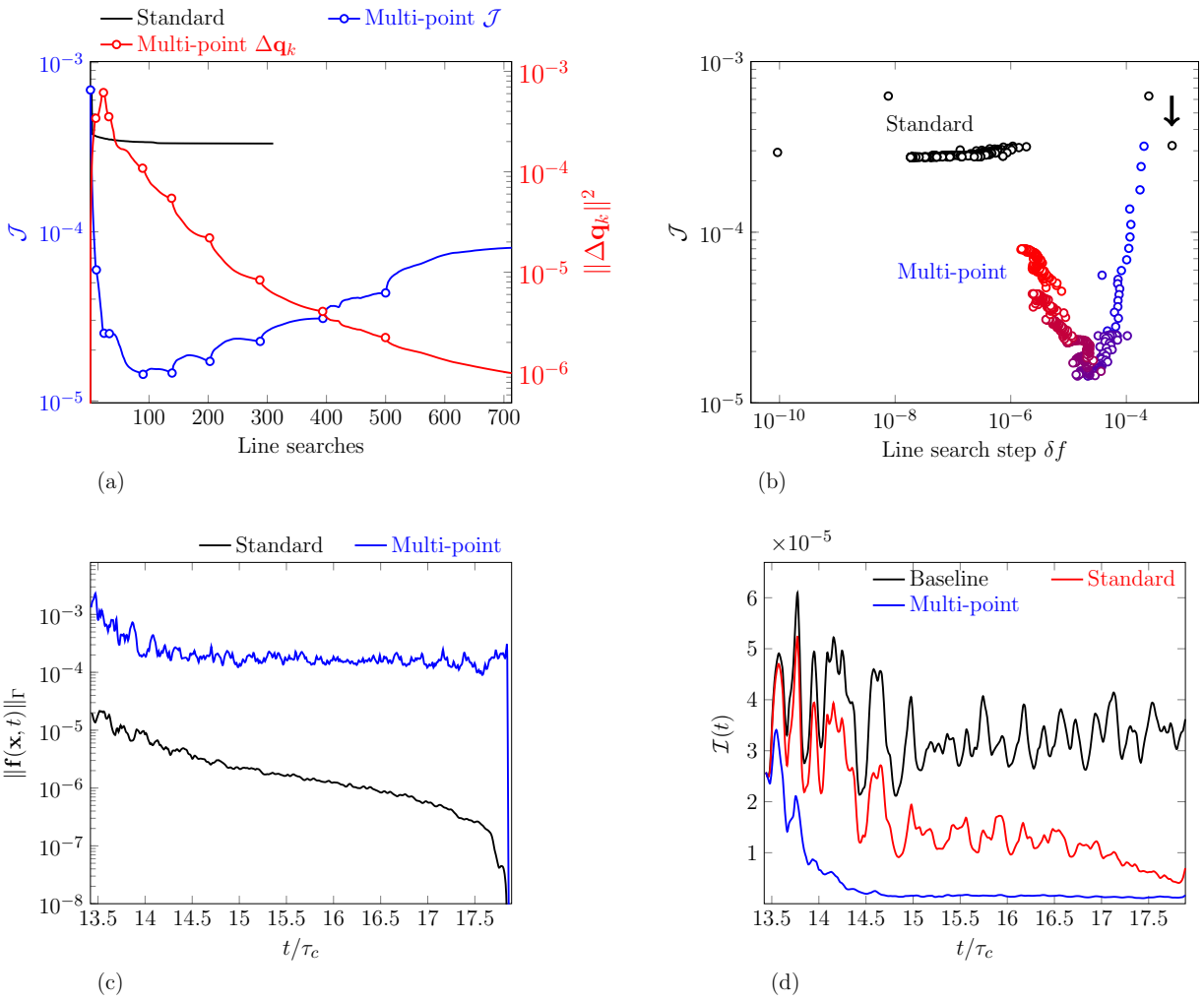


Figure 15.11: Optimization result for the three-dimensional Kolmogorov flow. The standard gradient-based result is plotted for comparison. (a) Reduction of $\mathcal{J}$ (13.14) and average $\|\Delta\mathbf{q}_k\|^2$, with markers denoting μ updates in Algorithm 6 and 7. (b) The search steps in optimization procedure. For the multi-point method, steps are marked from blue to red with increasing μ. (c) The control strength of the optimized controls. (d) $\mathcal{I}(t)$ (13.14b) of the baseline solution and the controlled solutions.

time. In Figure 15.11 (d), this control effectively suppresses the control objective after a short transient time $t \gtrsim 14.5\tau_c$. This transient time $t - t_i \approx 1.12\tau_c$ is longer than $\tau_A \approx 0.125\tau_c$, presumably because the control $\boldsymbol{\Theta}$ is set to manipulate only thermal energy of the flow. However, the control from the standard method has a longer transient response similar to the entire simulation time, leaving a significant of early times nearly unaffected, Figures 15.11 (c) and (d) together suggest again that the multi-point method is not blocked by small scale features of $\mathcal{J}$.

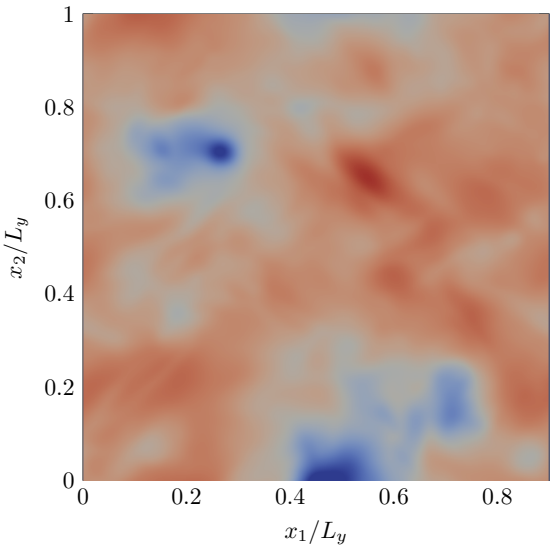
We can also confirm that the control found by the multi-point framework targets the large-scale structure. Figure 15.12 (a) and (b) show that the spanwise-averaged pressure fluctuations are suppressed. However, this is not achieved via laminarization. Figures 15.12 (c) and (d) show that smaller-scale turbulence still remains intact in the controlled flow, while large-scale structures are suppressed. This indicates that the control skirts these small-scale fluctuations which are unimportant for the control objective. Turbulence spectra in Figure 15.13 further supports this. The actuator only slightly decreases turbulence kinetic energy across small scales, while the large-scale spanwise pressure fluctuations in Figure 15.13 (b) are suppressed.

15.4.1 Flexibility of multi-point method with μ

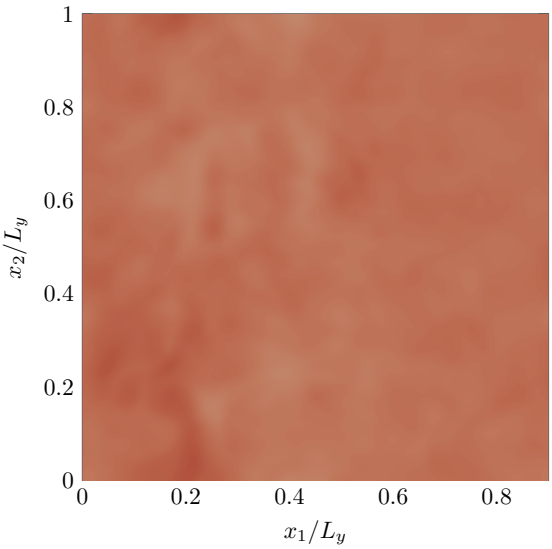
Before conclusion, we mark the importance of exploiting small μ in early line searches. In Section 14.2, it was illustrated with the logistics map example, that the augmented functional with small μ expands basin of attraction for the global optimum, which then becomes confined again as μ is increased. While the logistics map needs only one line search to reach the local optimum, typical flow control problems requires many line searches. Therefore, it is important to allow sufficient line searches with small μ, to find a smaller $\mathcal{J}$.

In Figure 15.14 we show another optimization for the three-dimensional Kolmogorov flow with a faster penalty strength increase. As expected, the discontinuity decreases faster as well. However, this decrease compromises the true objective of reducing $\mathcal{J}$. Abrupt jumps appear in $\mathcal{J}$. As the line searches continue, $\mathcal{J}$ decreases again, but when the subproblem terminates after only a few line searches, these upjumps are not fully countered. Thus, in Figure 15.14, optimization with faster increase of μ leads to an inferior $\mathcal{J}$. At the same discontinuity $\|\Delta\mathbf{q}_k\|^2 = 10^{-6}$, the slower case in Figure 15.14 (b) achieves more $\mathcal{J}$ reduction. In general, it is unclear what μ would best tailor the basin of attraction.

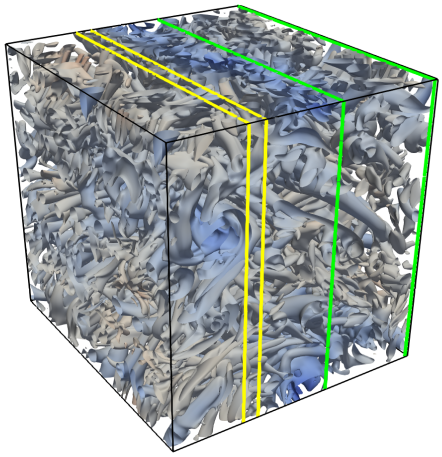
While it requires a caution, this still provides us with flexibility. To see this, we compare the optimization result of two regimens in Figure 15.14, mainly with their instantaneous objective functionals $\mathcal{I}(t)$. Figure 15.15 (a) shows $\mathcal{I}(t)$ in time at their final iterations, indicating that the flow of regimen B reaches a stationary state inferior to regimen A. However, this solution is obtained with about 4 times fewer line searches, as shown in Figure 15.14 (b), with a lower discontinuity than regimen A. This suggests that there is



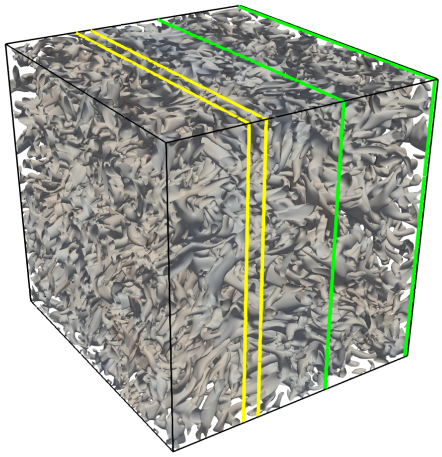
(a) Baseline x_3-averaged pressure



(b) Controlled x_3-averaged pressure



(c) Baseline



(d) Controlled

Figure 15.12: The effect of the control found by the multi-point method on the three-dimensional Kolmogorov flow at $t_0 = 14.51\tau_c$: pressure $p/p_0 \in [0.973, 1.008]$ averaged along x_3 direction of (a) the baseline solution and (b) the controlled solution, and; isosurfaces of Q-criterion ($Q = 20\tau_c^{-2}$) of (c) the baseline solution and (d) the controlled solution, colored by the pressure $p/p_0 \in [0.95, 1.05]$.

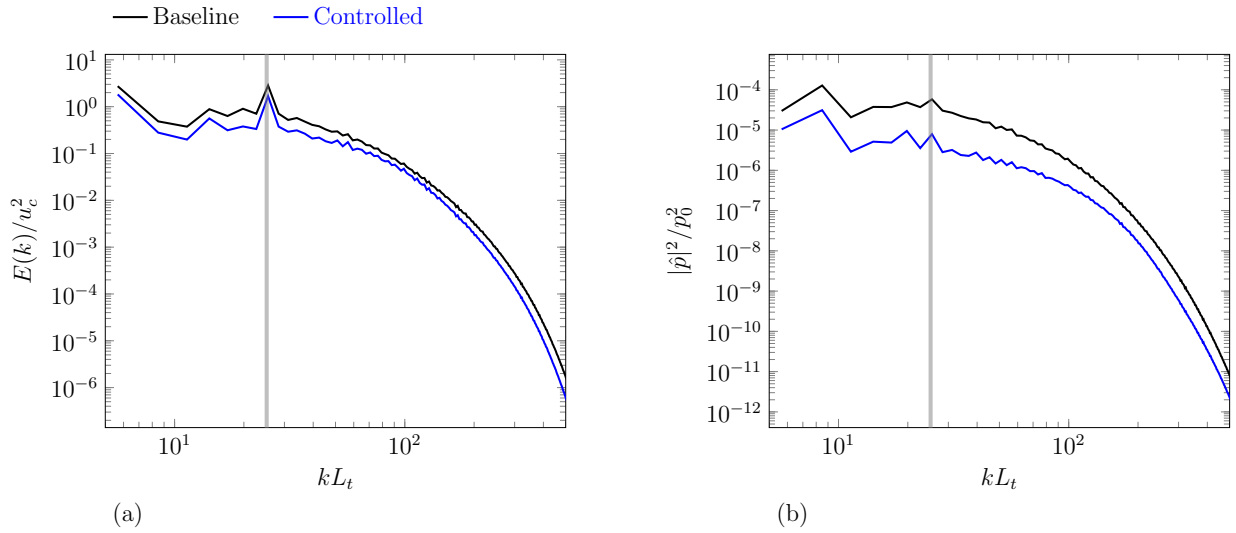


Figure 15.13: Three-dimensional Kolmogorov turbulence spectra of the baseline and controlled: (a) turbulence kinetic energy; and (b) pressure fluctuation.

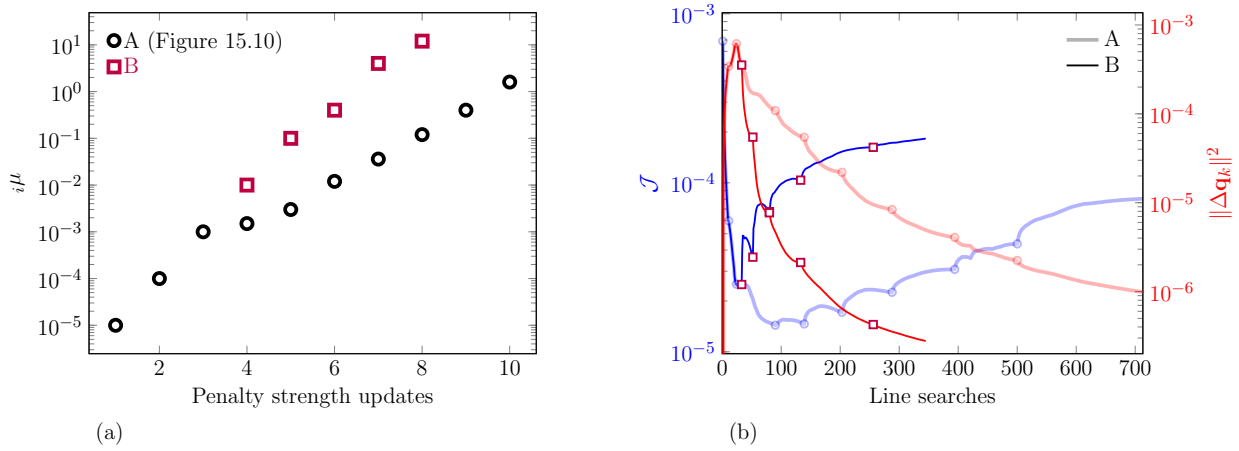
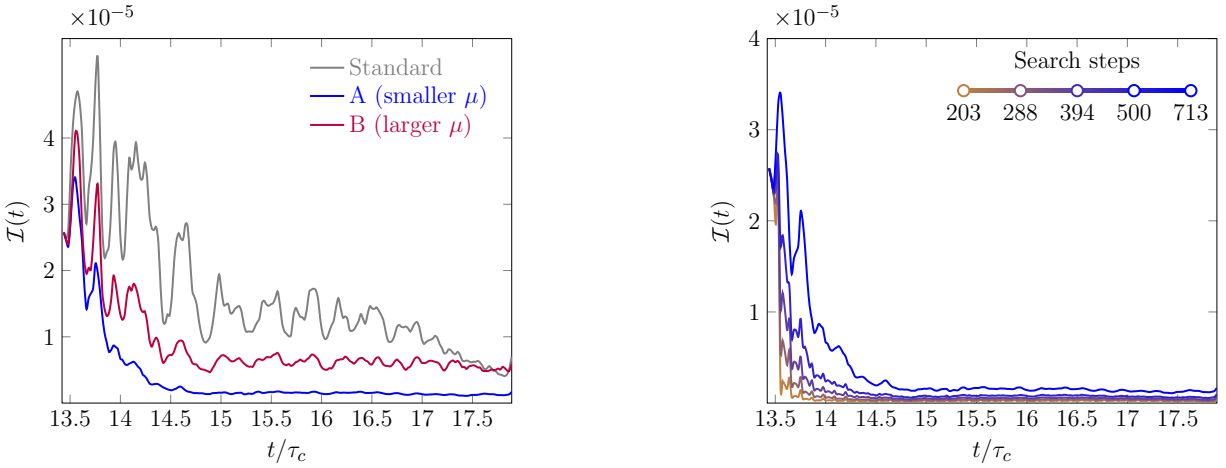


Figure 15.14: Another optimization result of three-dimensional Kolmogorov flow. The previous case (regimen A) is also plotted for comparison. (a) μ used for each subproblem. (b) Reduction of $\mathcal{J}$ (13.14) and average $\|\Delta\mathbf{q}_k\|^2$, with markers denoting μ updates.



(a) (b) Regimen A, after the 203-th line search

Figure 15.15: Optimization results from two regimens in Figure 15.14. (a) Instantaneous objective functional $\mathcal{I}(t)$ (13.14b) at their final iterations. Standard gradient-based optimization is included for a reference. (b) $\mathcal{I}(t)$ optimization of the regimen A after 203-th line search.

a trade-off between faster convergence and a better $\mathcal{J}$ outcome, and the multi-point method has a flexibility to choose between them by adjusting μ. Figure 15.15 (b) shows $\mathcal{I}(t)$ of regimen A changing throughout line searches. It changes mainly in its early transient response, while the terminal stationary value increases only slightly, still lower than that of regimen B. This shows that these two regimens converge to qualitatively different solutions. Both solutions, however, have similar transient time $t - t_i \approx 1.12\tau_c$ shorter than that of standard gradient-based optimization. The optimization behavior shown in Figure 15.15 suggests that the multi-point method has the flexibility to choose among these solutions by adjusting μ.

Chapter 16

Conclusion

The calculation of gradients, for analysis and optimization of turbulent flows is challenged by the chaotic dynamics of turbulence. A series of model problems, including a genuinely turbulent flow, was developed to assess this challenge and overcome it. These problems were used to quantify how the gradient grows in reverse time and its consequence. It was also shown how $\mathcal{J}$ becomes increasingly non-convex, so standard gradient-based algorithms find only a poor local optimum that does not reduce $\mathcal{J}$ significantly. Even in cases when it is conjectured that there are useful, relatively deterministic turbulence components, chaotic turbulence fluctuations obscure the gradient-based search methods. The advection plus Kuramoto–Sivashinsky (Adv+KS) system was introduced as a model for this. Reduced models might seek descriptions that avoid this challenge, and can be realized for the Adv+KS system, however they have not found a definite and broad success. A detailed simulation remains the only means of accurately predicting many flows, and then only if computer resources are available. In face of this challenge, we have developed a method to skirt the chaos and optimize without extracting an explicit reduced-order model for turbulence.

The method is developed as an extension of standard gradient-based method as part of a standard framework for equality-constrained optimization. Two well-understood features of this framework summarized in Chapter 11 are that the governing equation is strictly enforced and that it seeks the closest local minimum. These are hindering for chaotic dynamical systems, which was demonstrated and analyzed with simple examples (Logistics map and Lorenz system) in Chapter 12. A result of the exponential sensitivity is that the optimization is biased toward controls that reduce $\mathcal{J}$ at late times with actuations at earliy times. Another is that $\mathcal{J}$ becomes extremely non-convex, so gradient-based methods cannot search large regions of parameter space. Even exact gradients cannot circumvent the nonconvexity. This perspective is introduced in Section 12.2 with the typical horseshoe mapping illustrated with the one-dimensional logistics map. To evaluate these same mechanisms in more complex flow cases, two time scales are introduced in Section 12.3. The e-folding time t_λ represents gradient growth, and t_ϕ the viable search step decay due to the non-convexity of $\mathcal{J}$.

More complex flow examples are examined in Chapter 13. It is first confirmed with Kuramoto–Sivashinsky

Equation and two-dimensional Kolmogorov flow, that the optimization is similarly impacted by chaotic dynamics, even though the control problem is anticipated from its setup to have an effective control. The Adv+K–S model is introduced to show how optimization of a deterministic component can be obscured by chaotic component, which is thought to be similar to the challenge of flow turbulence. Turbulence per se is studied in a three-dimensional Kolmogorov flow, where large-scale pressure fluctuations are obscured by finer-scale turbulence. The two time scales t_λ and t_ϕ are quantified for all these examples. The t_ϕ is not proportional to t_λ, suggesting that t_λ (or the inverse Lyapunov exponent) alone does not fully describe the impact of chaos, and raised a need to investigate and establish the scaling behavior of t_ϕ.

Based on these observations a new optimization framework is proposed in Chapter 14. Its key feature is that the simulation is split into multiple intervals of length $T \lesssim \tau_\lambda$, each starting with an auxiliary intermediate condition. Allowing discontinuities at these times expands the optimization space, reducing the non-convexity of $\mathcal{J}$, thereby enlarging the initial search scale of gradient-based methods. As the search progresses, these discontinuities are penalized with increasing strength, ultimately becoming negligible. Two widely-used methods, quadratic penalty and augmented Lagrangian method, are introduced and shown to reduces $\mathcal{J}$ significantly.

The utility of the optimized solutions is, of course, in question because of its susceptibility to the same sensitivities that make such analysis of chaotic systems challenging. Two main points are discussed in this regard. The optimized solutions still reflects the true dynamics, at least to the same degree any numerical simulation of a chaotic system can, so as such they approximate continuous trajectories in a shadowing sense. This is supported by numerically constructing solutions that shadow the optimized solutions. In addition, the optimized solutions can be useful when they are interpreted as instances of an underlying control law. Like many open-loop controls, these solutions can be expected to be unstable, however they can provide useful data to infer a closed-loop control law. This is demonstrated for the Lorenz example, where an effective control law is inferred from the optimized solution using a machine learning technique.

The new method is demonstrated in Chapter 15 on the examples from Chapter 13. In all cases, the optimized controls are not biased to early times. This is the case even for the turbulent Kolmogorov flow. for which it targets large-scale pressure fluctuations without laminarization.

Numerical simulations in the multi-step penalty method has additional arithmetic operations and memory use compared to standard gradient-based methods. This computational cost mainly comes from evaluating intermediate discontinuities and their gradients, though it is not significant compared to time integration of the governing equation and its adjoint. Rather, multiple intervals enhances the scalability of the system, since each interval starts with its own initial condition, thereby enabling parallel-in-time integration of both

governing equation and its adjoint.

The multi-step penalty method has a flexibility with penalty strength μ to compromise between faster convergence and better solution. However, there are only loose rules for updating μ and it requires more systematic investigations to set up a specific guideline. Although all the examples in this study are optimal control problems, we believe it can be extended toward a general optimization problem.

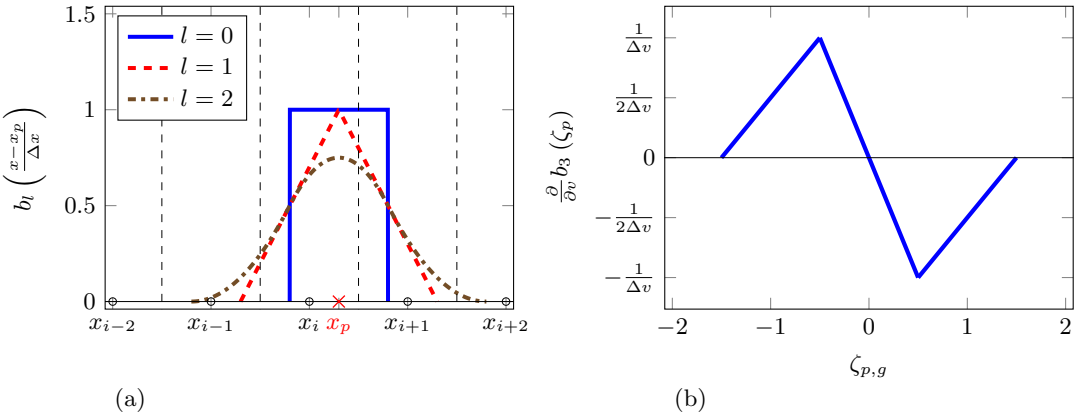
Appendix A

Shape functions for PIC formulation

A tensor-product of B-splines is often chosen as the shape function of computational particles $\mathbf{x}$ [23],

$$\begin{aligned} S_{\mathbf{x}}(\mathbf{x}-\mathbf{x}_p) &= \frac{1}{\Delta \mathbf{x}} \mathbf{b}_l \left(\frac{\mathbf{x}-\mathbf{x}_p}{\Delta \mathbf{x}} \right) \\ &= \frac{1}{\Delta x \Delta y \Delta z} b_l \left(\frac{x-x_p}{\Delta x} \right) b_l \left(\frac{y-y_p}{\Delta y} \right) b_l \left(\frac{z-z_p}{\Delta z} \right). \end{aligned} \tag{A.1}$$

Cloud-in-cell ($l = 0$) and triangular-shaped-cloud ($l = 1$) are the most common [20], which is shown in Figure A.1 (a).



(a) (b)

Figure A.1: (a) B-spline functions of order $l = 0, 1, 2$, and (b) derivative of B-spline of order $l = 3$.

Derivatives of B-splines are used in the source evaluation (3.22) of particle-pdf approach. We describe here $\frac{\partial}{\partial v} b_l \left(\frac{v_{i_v} - v_p}{\Delta v} \right)$ for $l = 3$:

$$\frac{\partial}{\partial v} b_3 \left(\zeta_{p,i_v} = \frac{v_{i_v} - v_p}{\Delta v} \right) = \begin{cases} -\frac{1}{\Delta v} \left(\frac{3}{2} - \zeta_{p,i_v} \right) & \frac{1}{2} \le \zeta_{p,i_v} \le \frac{3}{2} \\ -\frac{2\zeta_{p,i_v}}{\Delta v} & -\frac{1}{2} \le \zeta_{p,i_v} \le \frac{1}{2} \\ \frac{1}{\Delta v} \left(\frac{3}{2} + \zeta_{p,i_v} \right) & -\frac{3}{2} \le \zeta_{p,i_v} \le -\frac{1}{2}, \end{cases} \tag{A.2}$$

which is plotted in Figure A.1 (b).

Appendix B

Particle-exact sensitivity with respect to simulation parameters

B.1 Number of particles

For the Debye shielding configuration in Chapter 4, standard deviation of $\mathcal{J}_D$ and its particle-exact sensitivity $\partial_\theta \mathcal{J}_D$ is measured over 10^4 realizations, with different numbers of particles. The statistics are evaluated at $\omega_p t = 150.0$, when the particle-exact sensitivity starts to diverge in initial response. Figure B.1 shows that standard deviation of the particle-exact sensitivity also decreases with N_p, though not enough to suppress the exponential growth of standard deviation shown in Figure 5.4.

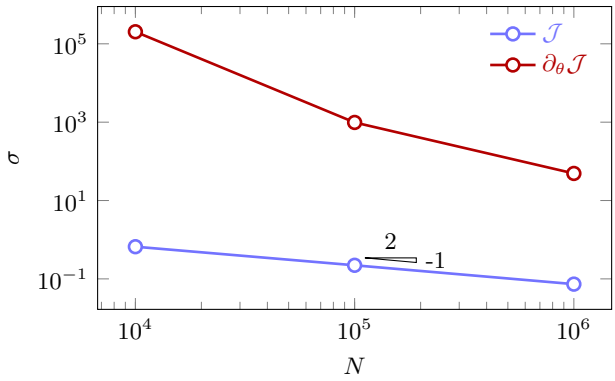


Figure B.1: Standard deviations of QoI (4.5) and its particle-exact sensitivity versus number of simulation particles.

B.2 Discretization parameters

Standard deviation of $\mathcal{J}$ and its particle-exact sensitivity is further investigated for different mesh sizes and time steps. Standard deviation of particle methods mainly depends on the number of particles, unless a numerical instability arises through discrete-particle noise and fluctuation due to too large Δx or Δt [20, 21, 38]. The particle-exact sensitivities are sampled at $\omega_p t = 150.0$, with different mesh sizes and time steps which does not induce artificial instability, and their statistics are tabulated in table B.1. It is shown

that standard deviation of QoI is insensitive to mesh size Δx or time step Δt. The standard deviation of particle-exact sensitivities is also insensitive to time step Δt, but not to mesh size Δx. When the mesh size is decreased, the standard deviation of sensitivities increase almost linearly. Considering that the use of mesh in force interpolation regularizes the discontinuity (or singularity) of short-range interaction with particles, it seems that decrease in mesh size weakens the regularizing effect, resulting in the increase of Lyapunov exponent [40]. Δx and Δt may affect statistical accuracy by correcting the mean, although not quantitatively confirmed in this paper.

	N_m	$\omega_p \Delta t$	μ	σ
$\mathcal{J}$	64	0.05	619.875	0.2223
	256	0.05	619.838	0.2226
	64	0.01	619.700	0.2205
$\partial_\theta \mathcal{J}$	64	0.05	26.449	984.04
	256	0.05	9.705	3.15×10^3
	64	0.01	9.1592	997.145

Table B.1: Mean and standard deviation of $\mathcal{J}$ and its particle-exact sensitivity, with different discretization parameters. Samples are taken with 10^5 particles at simulation time $t = 150\omega_{p,e}^{-1}$.

Appendix C

Nonlinear feedback control of the Lorenz system

We design a nonlinear feedback controller $f(t)$ for the Lorenz equation (10.1),

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y-x)\\ \frac{dy}{dt} &= x(\rho-z)-y\\ \frac{dz}{dt} &= xy-\beta z+f(t),\end{aligned}$$

with $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$. We denote the state as $\mathbf{q} = (x, y, z)^T \in \mathbb{R}^3$. The controller is sought to minimize the objective functional (12.3),

$$\mathcal{J} = \frac{1}{t_f - t_0}\int_{t_0}^{t_f} \mathcal{I}[\mathbf{q}]\, dt,$$

with the instantaneous objective functional $\mathcal{I}$ (12.4),

$$\mathcal{I}[\mathbf{q}] = \begin{cases} \frac{1}{2}\left(\frac{2x+y}{5}\right)^2 & 2x+y \geq 0 \\ 0 & \text{otherwise.}\end{cases}$$

The nonlinear feedback controller is designed with a nonlinear transformation of state space which converts the dynamics into a controllable linear dynamics [136, 137]. The formulation closely follows the work by Chen and Liu [137].

C.1 State-space exact linearization

We first introduce the preliminary theorems for the transformation [137]. Its actual application for the Lorenz system is in Section C.2.

Consider a single-input nonlinear control system

$$\dot{\mathbf{q}} = \mathcal{R}(\mathbf{q}) + \mathcal{G}(\mathbf{q})u, \tag{C.1}$$

where $\mathbf{q} \in \mathbb{R}^n$ and $u \in \mathbb{R}$ are the state variables and control parameter respectively. $\mathcal{R} : \mathbb{R}^n \to \mathbb{R}^n$ is the right-hand side of the nonlinear governing equation, and $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ defines the specific form of the controller. We define the adjoint action of $\mathcal{R}$ on $\mathcal{G}$,

$$\mathrm{ad}_{\mathcal{R}}\mathcal{G} = \nabla_{\mathbf{q}}\mathcal{G} \cdot \mathcal{R} - \nabla_{\mathbf{q}}\mathcal{R} \cdot \mathcal{G}, \tag{C.2}$$

where $\nabla_{\mathbf{q}}$ is the Jacobian with respect to $\mathbf{q}$. This is also known as the Lie bracket $[\mathcal{R}, \mathcal{G}] \equiv \mathrm{ad}_{\mathcal{R}}\mathcal{G}$.

Theorem C.1 (Exact linearization)**.** *For a given state $\mathbf{q}_0 \in \mathbb{R}^n$, suppose*

1. *that the controllability matrix*

$$\mathcal{C} = \left[\mathcal{G}(\mathbf{q}_0), \mathrm{ad}_{\mathcal{R}}\mathcal{G}(\mathbf{q}_0), \mathrm{ad}^2_{\mathcal{R}}\mathcal{G}(\mathbf{q}_0), \cdots, \mathrm{ad}^{n-1}_{\mathcal{R}}\mathcal{G}(\mathbf{q}_0)\right] \in \mathbb{R}^n \times \mathbb{R}^n \tag{C.3}$$

   has rank n, and

2. *that near $\mathbf{q}_0$ the vector space associated with the state $\mathbf{q}_0$*

$$D = \mathrm{span}\{\mathcal{G}(\mathbf{q}_0), \mathrm{ad}_{\mathcal{R}}\mathcal{G}(\mathbf{q}_0), \mathrm{ad}^2_{\mathcal{R}}\mathcal{G}(\mathbf{q}_0), \cdots, \mathrm{ad}^{n-2}_{\mathcal{R}}\mathcal{G}(\mathbf{q}_0)\} \tag{C.4}$$

   is closed under the adjoint action (C.2).

Then there exists a real-valued function $\lambda(\mathbf{q}) : \mathbb{R}^n \to \mathbb{R}$ defined in a neighborhood $U(\mathbf{q}_0)$ of $\mathbf{q}_0$ such that for all $\mathbf{q} \in U(\mathbf{q}_0)$

$$L_{\mathcal{G}}\lambda(\mathbf{q}) = L_{\mathrm{ad}_{\mathcal{R}}\mathcal{G}}\lambda(\mathbf{q}) = \cdots = L_{\mathrm{ad}^{n-2}_{\mathcal{R}}\mathcal{G}}\lambda(\mathbf{q}) = 0, \tag{C.5a}$$

and

$$L_{\mathrm{ad}^{n-1}_{\mathcal{R}}\mathcal{G}}\lambda(\mathbf{q}_0) \neq 0, \tag{C.5b}$$

where $\mathcal{L}_{\mathcal{G}}\lambda$ denotes the Lie derivative of $\lambda(\mathbf{q})$ with regard to the vector field $\mathcal{G}$,

$$\mathcal{L}_{\mathcal{G}}\lambda(\mathbf{q}) = (\mathcal{G} \cdot \nabla_{\mathbf{q}})\lambda(\mathbf{q}). \tag{C.6}$$

Furthermore, in $U(\mathbf{q}_0)$ there exists the nonlinear transformation $\hat{\mathbf{q}}(\mathbf{q}) : \mathbb{R}^n \to \mathbb{R}^n$ associated with $\lambda(\mathbf{q})$,

$$\begin{aligned}\hat{\mathbf{q}}(\mathbf{q}) &= [\hat{q}_1(\mathbf{q}), \hat{q}_2(\mathbf{q}), \cdots, \hat{q}_n(\mathbf{q})]^T \in \mathbb{R}^n \\ &= [\lambda(\mathbf{q}), L_{\mathcal{R}}\lambda(\mathbf{q}), \cdots, L_{\mathcal{R}}^{n-1}\lambda(\mathbf{q})]^T,\end{aligned} \tag{C.7}$$

and the nonlinear transformation v for the control u,

$$v = L_{\mathcal{R}}^n \lambda(\mathbf{q}) + \{L_{\mathcal{G}} L_{\mathcal{R}}^{n-1}\lambda(\mathbf{q})\}u, \tag{C.8}$$

such that the nonlinear dynamics (C.1) is transformed into linear controllable dynamics,

$$\frac{d\hat{q}_1}{dt} = \hat{q}_2, \quad \frac{d\hat{q}_2}{dt} = \hat{q}_3, \quad \cdots, \quad \frac{d\hat{q}_{n-1}}{dt} = \hat{q}_n, \quad \frac{d\hat{q}_v}{dt} = v, \tag{C.9}$$

by the transformation (C.7) and (C.8).

Proof. See Isidori [136, Chapter 4]. □

C.2 Feedback control design for the Lorenz equation

We apply Theorem C.1 to the Lorenz equation. $\mathcal{R}(\mathbf{q})$ and $\mathcal{G}(\mathbf{q})$ in (C.1) corresponding to the Lorenz equation (10.1) are

$$\mathcal{R}(\mathbf{q}) = \begin{pmatrix} \sigma(y - x) \\ x(\rho - z) - y \\ xy - \beta z \end{pmatrix}, \tag{C.10}$$

and

$$\mathcal{G}(\mathbf{q}) = (0, 0, 1)^T, \tag{C.11}$$

with u in (C.1) being the controller $f(t)$ in (10.1) we seek to design.

We first evaluate the controllability of the system with the two conditions in Theorem C.1. The adjoint

actions of $\mathcal{R}$ on $\mathcal{G}$ are then

$$\begin{aligned}\mathrm{ad}_{\mathcal{R}}\mathcal{G} &= \nabla_{\mathbf{q}}\mathcal{G}\cdot\mathcal{R} - \nabla_{\mathbf{q}}\mathcal{R}\cdot\mathcal{G} \\ &= \begin{pmatrix}0&0&0\\0&0&0\\0&0&0\end{pmatrix}\begin{pmatrix}\sigma(y-x)\\x(\rho-z)-y\\xy-\beta z\end{pmatrix} - \begin{pmatrix}-\sigma&\sigma&0\\\rho-z&-1&-x\\y&x&-\beta\end{pmatrix}\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}0\\x\\\beta\end{pmatrix}\end{aligned} \tag{C.12a}$$

$$\begin{aligned}\mathrm{ad}^2_{\mathcal{R}}\mathcal{G} &= \nabla_{\mathbf{q}}\mathrm{ad}_{\mathcal{R}}\mathcal{G}\cdot\mathcal{R} - \nabla_{\mathbf{q}}\mathcal{R}\cdot\mathrm{ad}_{\mathcal{R}}\mathcal{G} \\ &= \begin{pmatrix}0&0&0\\1&0&0\\0&0&0\end{pmatrix}\begin{pmatrix}\sigma(y-x)\\x(\rho-z)-y\\xy-\beta z\end{pmatrix} - \begin{pmatrix}-\sigma&\sigma&0\\\rho-z&-1&-x\\y&x&-\beta\end{pmatrix}\begin{pmatrix}0\\x\\\beta\end{pmatrix} \\ &= \begin{pmatrix}-\sigma x\\\sigma(y-x)+(1+\beta)x\\\beta^2-x^2\end{pmatrix}.\end{aligned} \tag{C.12b}$$

The controllability matrix (C.3) is then constructed with (C.11) and (C.12),

$$\mathcal{C} = \begin{pmatrix}0&0&-\sigma x\\0&x&(1+\beta-\sigma)x+\sigma y\\1&\beta&\beta^2-x^2\end{pmatrix}, \tag{C.13}$$

which determinant is $\det\mathcal{C} = \sigma x^2$. Therefore, as long as $\sigma \neq 0$ and $x \neq 0$, the controllability matrix has rank $n = 3$. Meanwhile, the vector space $D = \mathrm{span}\{\mathcal{G}, \mathrm{ad}_{\mathcal{R}}\mathcal{G}\}$ in (C.14) is closed under the adjoint action, since

$$\mathrm{ad}_{\mathcal{G}}(\mathrm{ad}_{\mathcal{R}}\mathcal{G}) \equiv -\mathrm{ad}_{\mathrm{ad}_{\mathcal{R}}\mathcal{G}}\mathcal{G} = (0,0,0)^T. \tag{C.14}$$

(C.13) and (C.14) together show that the Lorenz equation (10.1) satisfies the preliminary conditions for controllability in Theorem C.1, and therefore the nonlinear transformation $\hat{\mathbf{q}}(\mathbf{q})$ (C.7) exists.

We construct $\lambda(\mathbf{q})$ in Theorem C.1 that satisfies (C.5), in order to find the nonlinear transformation $\hat{\mathbf{q}}(\mathbf{q})$ (C.7). For the Lorenz equation, (C.5) becomes

$$L_{\mathcal{G}}\lambda(\mathbf{q}) = \frac{\partial\lambda}{\partial z} = 0 \tag{C.15a}$$

$$L_{\mathrm{ad}_{\mathcal{R}}\mathcal{G}}\lambda(\mathbf{q}) = x\frac{\partial\lambda}{\partial y} + \beta\frac{\partial\lambda}{\partial z} = 0 \tag{C.15b}$$

$$L_{\mathrm{ad}^2_{\mathcal{R}}\mathcal{G}}\lambda(\mathbf{q}) = -\sigma x\frac{\partial\lambda}{\partial x} + \{\sigma(y-x) + (1+\beta)x\}\frac{\partial\lambda}{\partial y} + (\beta^2 - x^2)\frac{\partial\lambda}{\partial z} \neq 0, \tag{C.15c}$$

For (C.15) $\lambda(\mathbf{q})$ must be a function of x, having no dependency on y and z. In fact, any arbitrary non-trivial $\lambda(x)$ can be used to construct the nonlinear transformation $\hat{\mathbf{q}}(\mathbf{q})$ (C.7). We use

$$\lambda(\mathbf{q}) = x, \tag{C.16}$$

with which the $\hat{\mathbf{q}} = (\hat{q}_1, \hat{q}_2, \hat{q}_3)^T$ is

$$\hat{q}_1 = \lambda(\mathbf{q}) = x \tag{C.17a}$$

$$\hat{q}_2 = L_{\mathcal{R}}\lambda(\mathbf{q}) = \sigma(y-x) \tag{C.17b}$$

$$\hat{q}_3 = L^2_{\mathcal{R}}\lambda(\mathbf{q}) = \sigma(\sigma+\rho)x - \sigma(\sigma+1)y - \sigma xz, \tag{C.17c}$$

and its inverse

$$x = \hat{q}_1 \tag{C.18a}$$

$$y = \hat{q}_1 + \frac{1}{\sigma}\hat{q}_2 \tag{C.18b}$$

$$z = \frac{1}{\sigma\hat{q}_1}\left\{(\rho-1)\hat{q}_1 - (\sigma+1)\hat{q}_2 - \hat{q}_3\right\}. \tag{C.18c}$$

The control transformation (C.8) is

$$\begin{aligned} v &= L^n_{\mathcal{R}}\lambda(\mathbf{q}) + \{L_{\mathcal{G}}L^{n-1}_{\mathcal{R}}\lambda(\mathbf{q})\}u, \\ &= \sigma^2(y-x)(\sigma+\rho-z) - \sigma(\sigma+1)\{x(\rho-z)-y\} - \sigma^2 z(y-x) - \sigma x(xy-\beta z) - \sigma xu. \end{aligned} \tag{C.19}$$

(C.17) and (C.19) transforms the Lorenz equation (10.1) into the linear control system,

$$\begin{aligned} \frac{d\hat{\mathbf{q}}}{dt} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}\hat{\mathbf{q}} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} v \\ &= \hat{\mathbf{A}}\hat{\mathbf{q}} + \hat{\mathbf{B}}v. \end{aligned} \tag{C.20}$$

The linear system (C.20) is controllable, and the design of linear systems is well-established. Any standard methodology can be used to design v, and once v is designed, the nonlinear control u can be deduced through the transformation (C.19).

Now we design a linear state-feedback controller v for (C.20) that can satisfy the objective functional (12.3). We recall the observable (12.2),

$$\mathcal{O} = 2x + y = \begin{pmatrix} 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix},$$

which is the term in the objective $\mathcal{I}$ (12.4). With the inverse transformation (C.18), the observable becomes

$$\mathcal{O} = 2x + y = 3\hat{q}_1 + \frac{1}{\sigma}\hat{q}_2 = \begin{pmatrix} 3 & \frac{1}{\sigma} & 0 \end{pmatrix} \begin{pmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{pmatrix} = \hat{\mathbf{C}}\hat{\mathbf{q}}. \tag{C.21}$$

We consider the following form of linear feedback v,

$$\begin{aligned} v &= -\begin{pmatrix} k_1 & k_2 & k_3 \end{pmatrix} \begin{pmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \end{pmatrix} + k_r r \\ &= -\hat{\mathbf{K}}\hat{\mathbf{q}} + k_r r, \end{aligned} \tag{C.22}$$

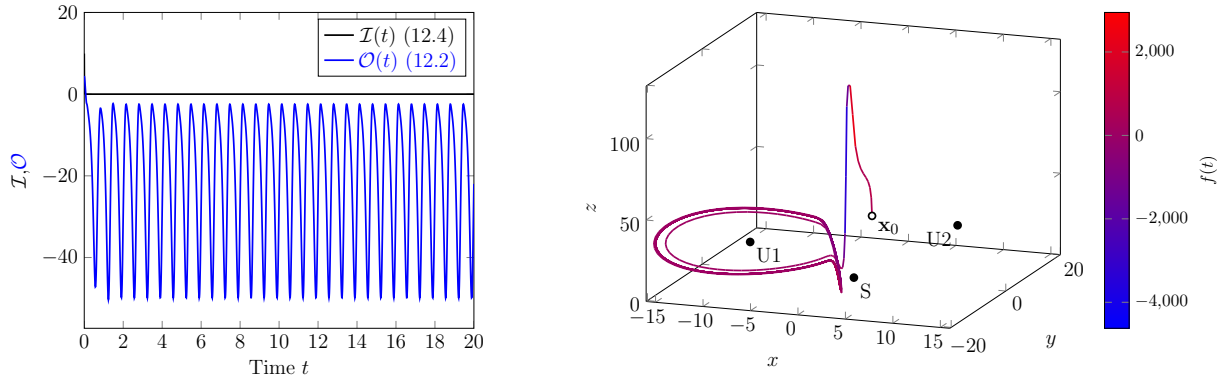
where the feedback gain $\hat{\mathbf{K}} = (k_1, k_2, k_3)$ determines the stability and performance of the controller, and we seek to control the observable $\mathcal{O}$ to match the reference value r. The constant k_r will be determined so that $\lim_{t\to\infty} \mathcal{O} = r$. With (C.22), the transformed Lorenz equation (C.20) is

$$\frac{d\hat{\mathbf{q}}}{dt} = (\hat{\mathbf{A}} - \hat{\mathbf{B}}\hat{\mathbf{K}})\hat{\mathbf{q}} + k_r r\hat{\mathbf{B}}, \tag{C.23}$$

and is stable if all eigenvalues of $\hat{\mathbf{A}} - \hat{\mathbf{B}}\hat{\mathbf{K}}$ are negative. Its characteristic equation is

$$\det(s\mathbf{I} - \hat{\mathbf{A}} + \hat{\mathbf{B}}\hat{\mathbf{K}}) = s^3 + k_3 s^2 + k_2 s + k_1 = 0, \tag{C.24}$$

Therefore the eigenvalues of $\hat{\mathbf{A}} - \hat{\mathbf{B}}\hat{\mathbf{K}}$ can be arbitrarily designed by the feedback gain $\hat{\mathbf{K}}$. In other words,



(a) (b)

Figure C.1: Nonlinear feedback control of the Lorenz system (10.1). (a) The instantaneous objective functional (12.4), and the observable (12.2) in time. (b) The feedback-controlled trajectory colored with the actuation magnitude.

the system is stabilizable. We consider somewhat strongly negative eigenvalues of $s_1 = -10$, $s_2 = -20$, and $s_3 = -30$ for a fast reaction of control. The corresponding feedback gain is then

$$\hat{\mathbf{K}} = \begin{pmatrix} 6000 & 1100 & 60 \end{pmatrix}. \tag{C.25}$$

The steady-state of this control system can be obtained from (C.23) with $d\hat{\mathbf{q}}/dt = 0$,

$$\hat{\mathbf{q}}_{ss} = -k_r r(\hat{\mathbf{A}} - \hat{\mathbf{B}}\hat{\mathbf{K}})^{-1}\hat{\mathbf{B}}, \tag{C.26}$$

where we target its observable to match the reference value r,

$$\mathcal{O}_{ss} = \hat{\mathbf{C}}\hat{\mathbf{q}}_{ss} = -k_r r\hat{\mathbf{C}}(\hat{\mathbf{A}} - \hat{\mathbf{B}}\hat{\mathbf{K}})^{-1}\hat{\mathbf{B}} = r, \tag{C.27}$$

which determines k_r to be

$$k_r = -\frac{1}{\hat{\mathbf{C}}(\hat{\mathbf{A}} - \hat{\mathbf{B}}\hat{\mathbf{K}})^{-1}\hat{\mathbf{B}}}, \tag{C.28}$$

the same as the inverse of zero-frequency gain of the feedback controller. (C.25) and (C.28) complete the design of v (C.22), which can control the observable $\mathcal{O}$ (C.21) to any arbitrary value r. The original control u then can be deduced from (C.19) together with (C.17) and (C.22),

$$u = \frac{1}{\sigma x}\left\{-v + \sigma^2(y-x)(\sigma+\rho-z) - \sigma(\sigma+1)\{x(\rho-z)-y\} - \sigma^2 z(y-x) - \sigma x(xy-\beta z)\right\}. \tag{C.29}$$

While the designed control can drive $\mathcal{O}$ to any value, we only need to keep $\mathcal{O} < 0$ to minimize the objective functional $\mathcal{J}$ (12.3). This indicates that the control u needs not be activated at all times. Furthermore, the numerator of u (C.29) shows that u is ill-defined near $\sigma x = 0$. While this is consistent with the controllability condition shown from (C.13), we simply do not have to activate the control when the state is near $x = 0$. Therefore, we augment the control u (C.29) with a smooth switch function,

$$W(\mathbf{q}) = \left(1 + \frac{1}{2}\tanh\sigma(x - 0.3) - \frac{1}{2}\tanh\sigma(x + 0.3)\right)\left(\frac{1}{2} + \frac{1}{2}\tanh(2x + y + 5)\right), \tag{C.30}$$

which becomes zero when $|x| < 0.3$ or $\mathcal{O} < -5$. This completes the design for the nonlinear feedback controller $f(t) = W(\mathbf{q})u$ for the Lorenz equation (10.1). Figure C.1 shows the numerical result of applying the designed feedback control, with the reference value $r = -10$.

Appendix D

An illustration for the decreasing utility of gradient in chaos

We first illustrate how the non-convexity of $\mathcal{J}$, observed in Section 12.1.4, impacts the utility of gradient. We consider four model objective functionals $\mathcal{J}_k$ for $k = 1, 2, 3, 4$,

$$\mathcal{J}_k(\theta) = \sin\left(G_k\theta - \frac{\pi}{6}\right), \tag{D.1}$$

with $G_k = -1.1 \times 10^{k-1}$. Figure D.1 (a) shows these increasingly non-convex $\mathcal{J}_k$. This mimicks a typical $\mathcal{J}$ of chaotic dynamical systems, with larger k representing longer time for gradient amplification. The gradient and Hessian increases with k,

$$\frac{\partial \mathcal{J}}{\partial \theta} = \frac{\sqrt{3}}{2}G_k \qquad \frac{\partial^2 \mathcal{J}}{\partial \theta^2} = \frac{1}{2}G_k^2, \tag{D.2}$$

and their ratio also increases exponentially with k. Figure D.1 (b) shows $\epsilon[\Delta\theta]$ (12.6), where the $\mathcal{O}[\Delta\theta]$ asymptotes shift toward smaller $\Delta\theta$ with larger k, as for the Lorenz example in Figure 12.5 (a).

On the other hand, these $\mathcal{J}_k$ are qualitatively different from the Lorenz example in Figure 12.5, in that the minimum errors remain the same. The increasing minimum error shown in Figure 12.5 is not a typical behavior of a finite-difference as in Figure D.1 (b), which suggests an additional impact of chaos. We consider round-off errors from its arithmetic operations of the finite-difference (12.5),

$$\left.\frac{\Delta \mathcal{J}}{\Delta \Theta}\right|_{\epsilon_1} = \frac{\mathcal{J}[\mathbf{q}; \mathbf{\Theta}_0 + \Delta\Theta\mathbf{e}_\theta] - \mathcal{J}[\mathbf{q}; \mathbf{\Theta}_0]}{\Delta\Theta} + \mathcal{O}\left(\frac{\epsilon_r}{\Delta\Theta}\right), \tag{D.3}$$

with ϵ_r the error in the difference in the numerator due to finite precision. While this error term is associated with any finite difference including $\mathcal{J}_k$ (D.1), $\mathcal{J}$ of chaotic dynamical systems involves another round-off error, as $\mathcal{J}$ is evaluated over many numerical time steps. Each time step involves many arithmetic operations, which induce round-off errors independent of $\Delta\mathbf{\Theta}$. This also amplifies state deviation $\delta\mathbf{q}$. This can be

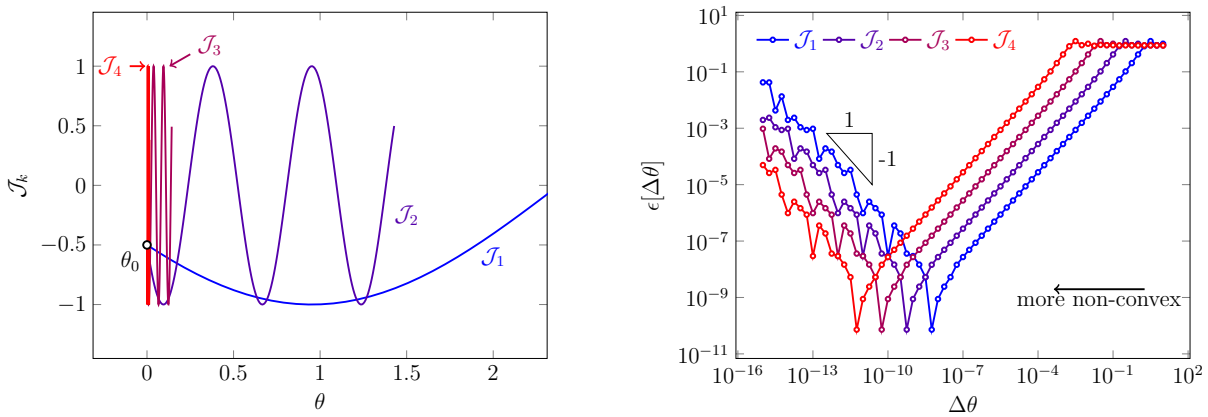


Figure D.1: (a) Model objective functionals $\mathcal{J}_k$ (D.1). (b) Relative error (12.6) for the gradient $\frac{\partial \mathcal{J}_k}{\partial \theta}$ of the model objective functionals.

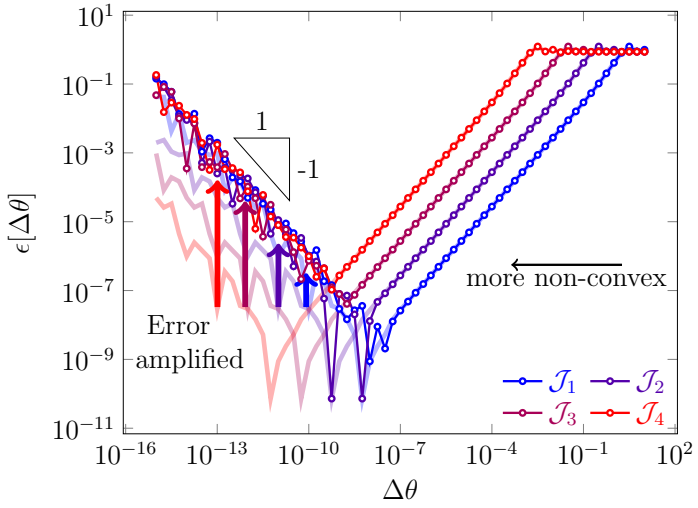


Figure D.2: Relative error (12.6) for the gradient $\frac{\partial \mathcal{J}_k}{\partial \theta}$ of the model objective functionals $\mathcal{J}_k$ (D.1), with the modified finite-difference (D.4). The relative errors from Figure D.1 (b) are plotted with light colors for comparison.

modeled as a random fluctuation in the parameter variation, so

$$\begin{aligned}\left.\frac{\Delta\mathcal{J}}{\Delta\Theta}\right|_{\epsilon_2} &= \frac{\mathcal{J}[\mathbf{q};\boldsymbol{\Theta}_0+\Delta\Theta\mathbf{e}_\theta+\epsilon_r'\xi]-\mathcal{J}[\mathbf{q};\boldsymbol{\Theta}_0]}{\Delta\Theta}+\mathcal{O}\left(\frac{\epsilon_r}{\Delta\Theta}\right)\\ &= \frac{\mathcal{J}[\mathbf{q};\boldsymbol{\Theta}_0+\Delta\Theta\mathbf{e}_\theta]-\mathcal{J}[\mathbf{q};\boldsymbol{\Theta}_0]}{\Delta\Theta}+\mathcal{O}\left(\frac{\epsilon_r}{\Delta\Theta}\right)+\mathcal{O}\left(\left\|\frac{\partial\mathcal{J}}{\partial\boldsymbol{\Theta}}\right\|\frac{\epsilon_r'}{\Delta\Theta}\right),\end{aligned} \tag{D.4}$$

where ξ is a random unit vector in $\mathbb{T}$. The last term in (D.4) shows the finite-precision error ϵ' amplified by the chaotic gradient. To model this error amplification for $\mathcal{J}_k$ (D.1), we evaluated the modified finite-difference in (D.4), with a random fluctuation $\epsilon_r\xi$ added in $\mathcal{J}[\mathbf{q};\boldsymbol{\Theta}_0+\Delta\Theta\mathbf{e}_\theta+\epsilon_r\xi]$. Figure D.2 shows $\epsilon[\Delta\theta]$ of the modified finite-differences, which exhibits the same behavior as the Lorenz example in Figure 12.5 (a). Figure D.2 completes the picture of how chaos impacts the utility of gradients: the increasing non-convexity of $\mathcal{J}$ makes the gradient accurate over a diminishing range of $\Delta\boldsymbol{\Theta}$; $\mathcal{J}$ evaluation over many time steps amplifies the finite-precision errors, increasing the minimum error of the gradient. These two factors inhibit the utility of the gradient for representing a useful $\mathcal{J}$ variation.

Appendix E

The metric entropy for a chaotic dynamical system

Although we do not compute it directly, the concept of entropy measures underlies some of our discussions about increasing non-convexity and the t_ϕ time scale. We therefore introduce it in some detail in this appendix. We start with a question for the logistics example in Section 12.2.2: for a given value $q_n = q$ at step n, what is its initial condition at q_0? From Figure 12.6, it is obvious that due to the folding motion there are two candidate q_{n-1}. The number of candidate q_0 increases as 2^n with the number of horseshoe mappings. If all candidates q_0 have the equal probability, then $\mathcal{P}_n = \frac{1}{2^n}$. This uncertainty can be quantified by Shannon's information entropy [159],

$$H[\mathcal{P}_n] = -E\left[\log \mathcal{P}_n\right] = -\sum_{k=1}^{2^n} \frac{1}{2^n} \log\left(\frac{1}{2^n}\right) = n \log 2, \tag{E.1}$$

which growth rate $\frac{1}{n} H[\mathcal{P}_n] = \log 2$ is positive. This concept defines the entropy growth of a chaotic dynamical system, which is introduced as metric entropy by Kolmogorov and Sinai [116, 117].

To formally introduce the metric entropy, we consider the discrete mapping (12.10) to be operated on a distribution (or set) of the states $A \subset \mathbb{Q}$,

$$\mathbf{q}(A) \equiv \{\mathbf{q}_{n+1} \in \mathbb{Q} | \mathbf{q}_{n+1} = \mathbf{q}(\mathbf{q}_n, T) \quad \text{for} \quad \forall \mathbf{q}_n \in A\}, \tag{E.2}$$

and its pre-image

$$\mathbf{q}^{-1}(A) \equiv \{\mathbf{q}_n \in \mathbb{Q} | \mathbf{q}_{n+1} = \mathbf{q}(\mathbf{q}_n, T) \quad \text{for} \quad \forall \mathbf{q}_{n+1} \in A\}. \tag{E.3}$$

In the example above $A = \{q\}$, and its successive pre-images $\mathbf{q}^{-n}(A)$ are the candidates for its initial condition.

In the example above, the probability of the initial condition was assumed to be uniform over all candidates. However, the probability of each candidate can be more completely formulated, based on ergodic theory [160]. We assume that the state space $\mathbb{Q}$ is a complete probability space with the probability measure $\mathcal{P}$, and the mapping (E.2) is a measure-preserving transformation [160, Chap 2]. We consider a set $\overline{Q} \subset \mathbb{Q}$

that is invariant under the discrete mapping (E.2),

$$\overline{Q} = \mathbf{q}(\overline{Q}). \tag{E.4}$$

Note that for any subset $A \subset \overline{Q}$, both its map and pre-image are subsets as well, i.e. $\mathbf{q}(A), \mathbf{q}^{-1}(A) \subset \overline{Q}$. Thus if the probability $\mathcal{P}$ in this ergodic distribution $\overline{Q}$ is known, then the probabilities of the initial condition candidates can be evaluated.

For $A = \{q\}$, the pre-image $\mathbf{q}^{-n}(A)$ include all the candidates for $q_n = q$. We want to discern each candidate and evaluate its probability respectively. This can be done systematically by partitioning $\overline{Q}$ with a finite partition $\xi = \{C_1, C_2, \cdots, C_p\}$, such that

$$\begin{aligned} C_i \cap C_j &= \emptyset \qquad \forall i \neq j \\ \bigcup_{i=1}^{p} C_i &= \overline{Q}. \end{aligned} \tag{E.5}$$

An intersection of pre-images of this partition can sort initial conditions according to their trajectories. For example, an intersection

$$C_{i1} \cap \mathbf{q}^{-1}(C_{i2}) \cap \cdots \cap \mathbf{q}^{n-1}(C_{in}) \cap \mathbf{q}^{-n}(A), \tag{E.6}$$

indicates initial conditions $q_0 \in C_{i1}$ that pass through $q_1 \in C_{i2}$, $\cdots$, $q_{n-1} \in C_{in}$ successively and arrive in $q_n \in A$. Its probability can be denoted as $\mathcal{P}\left(C_{i1} \cap \mathbf{q}^{-1}(C_{i2}) \cap \cdots \cap \mathbf{q}^{n-1}(C_{in}) \cap \mathbf{q}^{-n}(A)\right)$. Likewise, all the initial condition candidates can be sorted out by using different combinations of $C_{i1}, C_{i2}, \cdots, C_{in}$. We generalize the Shannon's entropy (E.1) using the partition ξ,

$$H_n[\mathbf{q}, \xi] = -\sum_{i_0, i_1, \cdots, i_n} \mathcal{P}\left(\bigcap_{k=0}^{n} \mathbf{q}^{-k}(C_{i_k})\right) \log \mathcal{P}\left(\bigcap_{k=0}^{n} \mathbf{q}^{-k}(C_{i_k})\right), \tag{E.7}$$

which increases with n and requires a finer partition. The metric entropy is defined as its maximum growth rate over all finite partitions [116, 117],

$$h[\mathbf{q}] = \sup_{\xi} \lim_{n \to \infty} \frac{1}{n} H_n[\mathbf{q}, \xi]. \tag{E.8}$$

While it is clear that the positive entropy is related with the non-convexity of $\mathcal{J}$ via the folding of horseshoe mapping, direct evaluation of (E.8) for a turbulent flow simulation is infeasible. It requires the ergodic distribution of all turbulent flow states and its probabilistic distribution. Furthermore, one needs to

come up with a partition that maximizes the entropy. This is very complex tasks even for one-dimensional maps [161], and so it does not seem possible for turbulence.

Many practical estimations of the metric entropy uses its relation with the Lyapunov exponents λ_k (**??**). It is shown that the metric entropy is upper-bounded by the sum of positive λ_k [162],

$$h[\mathbf{q}] \leq \sum_{\lambda_k > 0} \lambda_k, \tag{E.9}$$

and furthermore the equality holds true for Hamiltonian dynamics [163] and dissipative dynamical systems that are uniformly hyperbolic [162]. Its connection to turbulent flows is supported by a chaotic hypothesis [8, 154, 155], an assumption that sufficiently large systems may behave hyperbolically. Berera *et al.* [164] computed the metric entropy of two-dimensional turbulence based on the equality in (E.9) and studied its scaling behavior with Reynolds number and other invariants. This is also studied for three-dimensional isotropic turbulence with the same numerical approach [165]. In their study, Lyapunov spectra are computed with the method proposed by Benettin *et al.* [166], for which computational cost scales as $\mathcal{O}(M^2)$ for M Lyapunov exponents. The number of positive λ_k are theoretically estimated for turbulent flows as power laws of Reynolds number [167–169], and this has been numerically confirmed by Berera and Clark [165]. They found that turbulent flows with $\mathrm{Re} \leq 212$ have at most 3000 positive Lyapunov exponents. The $\mathcal{O}(M^2)$ scaling of the cost to compute M Lyapunov exponents is the main bottleneck for directly computing the entropy of turbulent flows at higher Reynolds numbers. There are additional definitions of entropy for chaotic dynamical systems, though all are expected to be similarly expensive [114].

More salient to our purposes, the probabilistic viewpoint of the entropy is not a direct description of the impact of chaos on optimization. The rate the uncertainty is produced, or how a system becomes less predictable in time, only indirectly quantifies the non-convexity of $\mathcal{J}$. For example, it is more important how quickly the typical size of neighborhood for a local minimum, as in Definition 11.1, decreases in time.

Additional quantities, which might better describe the complexity for purposes of optimization, such as fractal dimensions [114, 118–120], are similarly difficult to evaluate. They too require multiple Lyapunov exponents [164, 165, 170]. Furthermore, while fractal dimensions may characterize non-convexity, they lack the notion of the rate of change in time. They are only applied to the invariant sets, such as the ergodic distribution $\overline{Q}$ introduced above, which are only valid for $t \to \infty$. This is crucial for our study, since the optimization time period is finite and limited by how fast $\mathcal{J}$ becomes non-convex.

References

[1] A. Piel, Plasma physics: an introduction to laboratory, space, and fusion plasmas, Springer Science & Business Media, 2010.

[2] F. Filbet, E. Sonnendrücker, Numerical methods for the Vlasov equation, in: Numerical mathematics and advanced applications, Springer, Milano, 2003, pp. 459–468.

[3] A. Grassi, L. Fedeli, A. Sgattoni, A. Macchi, Vlasov simulation of laser-driven shock acceleration and ion turbulence, Plasma Physics and Controlled Fusion 58 (034021).

[4] J. Büchner, N. Elkina, Vlasov code simulation of anomalous resistivity, Space Science Reviews 121 (1-4) (2005) 237–252.

[5] J. P. Verboncoeur, Particle simulation of plasmas: review and advances, Plasma Physics and Controlled Fusion 47 (2005) A231–A260.

[6] D. J. Lea, T. W. N. Haine, M. R. Allen, J. A. Hansen, Sensitivity analysis of the climate of a chaotic ocean circulation model, Quarterly Journal of the Royal Meteorological Society 128 (586) (2002) 2587–2605.

[7] D. Ruelle, Differentiation of SRB states, Communications in Mathematical Physics 187 (1).

[8] D. Ruelle, A review of linear response theory for general differentiable dynamical systems, Nonlinearity 22 (4) (2009) 855.

[9] J. Thuburn, Climate sensitivities via a Fokker–Planck adjoint approach, Quarterly Journal of the Royal Meteorological Society 131 (605) (2005) 73–92.

[10] P. J. Blonigan, S. A. Gomez, Q. Wang, Least squares shadowing for sensitivity analysis of turbulent fluid flows, 52nd Aerospace Sciences Meeting.

[11] G. L. Eyink, T. W. N. Haine, D. J. Lea, Ruelle's linear response formula, ensemble adjoint schemes and Lévy flights, Nonlinearity 17.5.

[12] R. V. Abramov, A. J. Majda, Blended response algorithms for linear fluctuation-dissipation for complex nonlinear dynamical systems, Nonlinearity 20 (12) (2007) 2793.

[13] Q. Wang, Forward and adjoint sensitivity computation of chaotic dynamical systems, Journal of Computational Physics 235 (2013) 1–13.

[14] L. Y. Adrianova, Introduction to linear systems of differential equations, American Mathematical Society, 1995.

[15] D. Frenkel, B. Smit, Understanding molecular simulation: from algorithms to applications, Vol. 1, Academic Press, 2001.

[16] Y.-C. Lai, T. Tél, Transient chaos: complex dynamics on finite time scales, Vol. 173, Springer Science & Business Media, 2011.

[17] J. M. Ottino, The kinematics of mixing: stretching, chaos, and transport, Cambridge University Press, 1989.

[18] P. Welander, Studies on the general development of motion in a two-dimensional, ideal fluid, Tellus 7 (2) (1955) 141–156.

[19] J. E. Hicken, D. W. Zingg, The role of dual consistency in functional accuracy: error estimation and superconvergence, 20th AIAA Computational Fluid Dynamics Conference.

[20] R. W. Hockney, J. W. Eastwood, Computer simulation using particles, Taylor and Francis Group, 1988.

[21] C. K. Birdsall, A. B. Langdon, Plasma physics via computer simulation, CRC Press, 2004.

[22] J. M. Dawson, Particle simulation of plasmas, Reviews of Modern Physics 55 (2).

[23] G. Lapenta, Particle simulations of space weather, Journal of Computational Physics 231 (3) (2012) 795–821.

[24] C. K. Birdsall, Particle-in-cell charged-particle simulations, plus Monte Carlo collisions with neutral atoms, PIC-MCC, IEEE Transactions on Plasma Science 19 (2) (1991) 65–85.

[25] P. D. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, Journal of Computational Physics 138 (2) (1997) 821–857.

[26] G.-H. Cottet, P. D. Koumoutsakos, Vortex methods: theory and practice, Cambridge University Press, 2000.

[27] A. K. Chaniotis, D. Poulikakos, P. Koumoutsakos, Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flow, Journal of Computational Physics 182 (1) (2002) 67–90.

[28] Y. Chen, S. E. Parker, Coarse-graining phase space in δf particle-in-cell simulations, Physics of Plasmas 14 (8).

[29] B. Wang, G. H. Miller, P. Collela, A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas, SIAM Journal on Scientific Computing 33 (6) (2011) 3509–3537.

[30] J. J. Monaghan, Extrapolating B-splines for interpolation, Journal of Computational Physics 60 (1985) 253–262.

[31] R. J. Procassini, C. K. Birdsall, E. C. Morse, A fully kinetic, self-consistent particle simulation model of the collisionless plasma–sheath region, Physics of Fluid B: Plasma Physics 2 (12) (1990) 3191–3205.

[32] R. H. Landau, M. J. Páez, C. C. Bordeianu, A survey of computational physics: introductory computational science, Princeton University Press, 2008.

[33] R. J. Leveque, Finite volume methods for hyperbolic problems, Cambridge University Press, 2002.

[34] A. Y. Aydemir, A unified Monte Carlo interpretation of particle simulations and applications to non-neutral plasmas, Physics of Plasmas 1 (4) (1994) 822–831.

[35] W. M. Nevins, G. W. Hammett, A. M. Dimits, W. Dorland, D. E. Shumaker, Discrete particle noise in particle-in-cell simulations of plasma microturbulence, Physics of Plasmas 12 (12).

[36] I. Holod, Z. Lin, Statistical analysis of fluctuations and noise-driven transport in particle-in-cell simulations of plasma turbulence, Physics of Plasmas 14 (3).

[37] A. Campa, Physics of long-range interacting systems, OUP Oxford, 2014.

[38] A. B. Langdon, Kinetic theory for fluctuations and noise in computer simulation of plasma, Physics of Fluids 22 (1) (1979) 163–171.

[39] W. Hayes, Computer simulations, exact trajectories, and the gravitational N-body problem, American Journal of Physics 72 (9) (2004) 1251–1257.

[40] H. E. Kandrup, H. Smith Jr, On the sensitivity of the N-body problem to small changes in initial conditions, The Astrophysical Journal 374 (1991) 255–265.

[41] J. Goodman, D. C. Heggie, P. Hut, On the exponential instability of N-body systems, The Astrophysical Journal 415 (1993) 715.

[42] M. Hemsendorf, D. Merritt, Instability of the gravitational N-body problem in the large-N limit, The Astrophysical Journal 580 (1) (2002) 606.

[43] H. Spohn, Large scale dynamics of interacting particles, Springer Science & Business Media, 2012.

[44] R. G. Bartle, D. R. Sherbert, Introduction to real analysis, NJ: Wiley, 2011.

[45] L. Ambrosio, N. Gigli, G. Savaré, Gradient flows: in metric spaces and in the space of probability measures, Springer Science & Business Media, 2008.

[46] C. Villani, Optimal transport: old and new, Vol. 338, Springer Science & Business Media, 2008.

[47] E. del Barrio, E. Gine, C. Matran, Central limit theorems for the Wasserstein distance between the empirical and the true distributions, Annals of Probability (1999) 1009–1071.

[48] E. del Barrio, J.-M. Loubes, Centeral limit theorems for empirical transportation cost in general dimension, arXiv preprint arXiv:1705.01299.

[49] F. P. Vasilyev, A. Y. Ivanitskiy, Dual simplex method, in: In-depth analysis of linear programming, Springer, Dordrecht, 2001.

[50] G. Nastac, J. W. Labahn, L. Magri, M. Ihme, Lyapunov exponent as a metric for assessing the dynamic content and predictability of large-eddy simulations, Physical Review Fluids 2 (9). doi:10.1103/physrevfluids.2.094606.
URL http://dx.doi.org/10.1103/PhysRevFluids.2.094606

[51] P. C. Liewer, V. K. Decyk, A general concurrent algorithm for plasma particle-in-cell simulation codes, Journal of Computational Physics 85 (2) (1989) 302–322.

[52] A. Fridman, Plasma chemistry, Cambridge University Press, 2008.

[53] K.-U. Riemann, The Bohm criterion and sheath formation, Journal of Physics D: Applied Physics 24 (4) (1991) 493.

[54] J. Craske, Adjoint sensitivity analysis of chaotic systems using cumulant truncation, Chaos, Solitons and Fractals: the interdisciplinary journal of Nonlinear Science, and Nonequilibrium and Complex Phenomena 119 (2019) 243–254. doi:10.1016/j.chaos.2018.12.024.
URL https://doi.org/10.1016/j.chaos.2018.12.024

[55] J.-P. M. Pèraud, N. G. Hadjiconstantinou, Adjoint-based deviational Monte Carlo methods for phonon transport calculations, Physics of Fluid B 91 (23).

[56] J. R. Martins, Perspectives on aerodynamic design optimization, AIAA Scitech 2020 Forum (January) (2020) 1–21. doi:10.2514/6.2020-0043.

[57] J. Nocedal, S. Wright, Numerical optimization, Springer Science & Business Media, 2006.

[58] D. P. Bertsekas, Constrained Optimization and Lagrange Multiplier Methods, 1982.

[59] D. Leonard, N. V. Long, Optimal control theory and static optimization in economics, Cambridge University Press;, 1992.

[60] T. R. Bewley, P. Moin, R. Temam, DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms, Journal of Fluid Mechanics 447 (2001) 179–225. doi:10.1017/S0022112001005821.

[61] J. Kim, T. R. Bewley, A linear systems approach to flow control, Annual Review of Fluid Mechanics 39 (2007) 383–417. doi:10.1146/annurev.fluid.39.050905.110153.

[62] A. Jameson, L. Martinelli, N. A. Pierce, Optimum aerodynamic design using the navier–stokes equations, Theoretical and computational fluid dynamics 10 (1998) 213–237.

[63] N. Sikarwar, P. Morris, The use of an adjoint method for optimization of blowing in a convergent-divergent nozzle, International Journal of Aeroacoustics 14 (1-2) (2015) 327–351. doi:10.1260/1475-472X.14.1-2.327.

[64] G. A. Mensah, J. P. Moeck, Acoustic Damper Placement and Tuning for Annular Combustors: An Adjoint-Based Optimization Study, Journal of Engineering for Gas Turbines and Power 139 (6). doi:10.1115/1.4035201.

[65] K. J. Fidkowski, D. L. Darmofal, Review of output-based error estimation and mesh adaptation in computational fluid dynamics, AIAA Journal 49 (4) (2011) 673–694. doi:10.2514/1.J050073.

[66] L. Shi, Z. J. Wang, Adjoint-based error estimation and mesh adaptation for the correction procedure via reconstruction method, Journal of Computational Physics 295 (2015) 261–284. doi:10.1016/j.jcp.2015.04.011.
URL http://dx.doi.org/10.1016/j.jcp.2015.04.011

[67] R. A. Nasralla, A. M. Daoud, K. A. Fattah, M. H. Sayyouh, Fast and efficient sensitivity calculation using adjoint method for three-phase field-scale history matching, Journal of Petroleum Science and Engineering 77 (3-4) (2011) 338–350. doi:10.1016/j.petrol.2011.04.009.

[68] E. J. Parish, K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, Journal of Computational Physics 305 (2016) 758–774. doi:10.1016/j.jcp.2015.11.012.

[69] Using field inversion to quantify functional errors in turbulence closures, Physics of Fluids 28 (4). doi:10.1063/1.4947045.

[70] A. P. Singh, S. Medida, K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, AIAA Journal 55 (7) (2017) 2215–2227. doi:10.2514/1.J055595.

[71] C. He, Y. Liu, L. Gan, A data assimilation model for turbulent flows using continuous adjoint formulation, Physics of Fluids 30 (10). doi:10.1063/1.5048727.

[72] M. P. Juniper, Triggering in the horizontal Rijke tube: Non-normality, transient growth and bypass transition, Journal of Fluid Mechanics 667 (2011) 272–308. doi:10.1017/S0022112010004453.

[73] S. M. Rabin, C. P. Caulfield, R. R. Kerswell, Triggering turbulence efficiently in plane Couette flow, Journal of Fluid Mechanics 712 (2012) 244–272. doi:10.1017/jfm.2012.417.

[74] R. Kerswell, Nonlinear Nonmodal Stability Theory, Annual Review of Fluid Mechanics 50 (1) (2018) 319–345. doi:10.1146/annurev-fluid-122316-045042.
URL http://www.annualreviews.org/doi/10.1146/annurev-fluid-122316-045042

[75] T. S. Eaves, C. P. Caulfield, Disruption of SSP=VWI states by a stable stratification, Journal of Fluid Mechanics 784 (2015) 548–564. doi:10.1017/jfm.2015.596.

[76] A. Kord, J. Capecelatro, Optimal perturbations for controlling the growth of a Rayleigh-Taylor instability, Journal of Fluid Mechanics (2019) 150–185doi:10.1017/jfm.2019.532.

[77] J. B. Freund, Adjoint-based optimization for understanding and suppressing jet noise, Journal of Sound and Vibration 330 (17) (2011) 4114–4122.

[78] M. Wei, J. B. Freund, A noise-controlled free shear flow, Journal of Fluid Mechanics 546 (2006) 123–152.

[79] J. Kim, D. J. Bodony, J. B. Freund, Adjoint-based control of loud events in a turbulent jet, Journal of Fluid Mechanics 741 (2014) 28–59.

[80] E. N. Lorenz, Deterministic nonperiodic flow, Journal of the Atmospheric Sciences 20 (2) (1963) 130–141.

[81] Q. Wang, J. H. Gao, The drag-adjoint field of a circular cylinder wake at Reynolds numbers 20, 100 and 500, Journal of Fluid Mechanics 730 (2013) 145–161. doi:10.1017/jfm.2013.323.

[82] R. Vishnampet, D. J. Bodony, J. B. Freund, A practical discrete-adjoint method for high-fidelity compressible turbulence simulations, Journal of Computational Physics 285 (2015) 173–192.

[83] P. J. Blonigan, M. Farazmand, T. P. Sapsis, Are extreme dissipation events predictable in turbulent fluid flows?, Physical Review Fluids 4 (4) (2019) 1–21. arXiv:1807.10263, doi:10.1103/PhysRevFluids.4.044606.

[84] J. Javier Otero Pérez, A. Sharma, R. D. Sandberg, Direct numerical simulations for adjoint-based optimal flow and noise control of a backward-facing step, 22nd AIAA/CEAS Aeroacoustics Conference, 2016doi:10.2514/6.2016-2889.

[85] P. M. Pardalos, A. Zilinskas, J. Zilinskas, Non-Convex Multi-Objective Optimization, Springer, 2017.

[86] R. Horst, H. Tuy, Global optimization: Deterministic approaches, Springer, 1996. arXiv:arXiv:1011.1669v3.

[87] R. Horst, Introduction to global optimization, Nonconvex optimization and its applications ; v. 3, Kluwer Academic Publishers, Dordrecht, 1995.

[88] J. Mockus, Bayesian approach to global optimization : theory and practice, Mathematics and its applications. Soviet series, Kluwer Academic, Dordrecht, 1989.

[89] R. Strongin, Y. Sergeyev, Global optimization with non-convex constraints: Sequential and parallel algorithms, Vol. 3, Springer Science & Business Media, 2013.

[90] R. Schaefer, Foundation of Global Genetic Optimization, Springer, Berlin, 2007.

[91] A. Zhigljavsky, A. Zilinskas, Stochastic Global Optimization, Springer, Dordrecht, 2008.

[92] P. J. Blonigan, Q. Wang, Probability density adjoint for sensitivity analysis of the mean of chaos, Journal of Computational Physics 270 (2014) 660–686.

[93] P. J. Blonigan, Adjoint sensitivity analysis of chaotic dynamical systems with non-intrusive least squares shadowing, Journal of Computational Physics 348 (2017) 803–826. doi:10.1016/j.jcp.2017.08.002.
URL http://dx.doi.org/10.1016/j.jcp.2017.08.002

[94] P. J. Blonigan, Q. Wang, Multiple shooting shadowing for sensitivity analysis of choatic dynamical systems, Journal of Computational Physics 354 (2018) 447–475.

[95] N. Chandramoorthy, Q. Wang, A computable realization of Ruelle's formula for linear response of statistics in chaotic systems (feb 2020). arXiv:2002.04117.

[96] S. W. Chung, S. D. Bond, E. C. Cyr, J. B. Freund, Regular sensitivity computation avoiding chaotic effects in particle-in-cell plasma methods, Journal of Computational Physics 400 (2020) 108969.

[97] D. G. Crighton, M. Gaster, Stability of slowly diverging jet flow, Journal of Fluid Mechanics 77 (2) (1976) 397–413. doi:10.1017/S0022112076002176.

[98] P. Jordan, T. Colonius, Wave Packets and Turbulent Jet Noise, Annual Review of Fluid Mechanics 45 (1) (2013) 173–195. doi:10.1146/annurev-fluid-011212-140756.
URL http://www.annualreviews.org/doi/10.1146/annurev-fluid-011212-140756

[99] K. Gudmundsson, T. Colonius, Instability wave models for the near-field fluctuations of turbulent jets, Journal of Fluid Mechanics 689 (2011) 97–128. doi:10.1017/jfm.2011.401.

[100] A. V. Cavalieri, D. Rodríguez, P. Jordan, T. Colonius, Y. Gervais, Wavepackets in the velocity field of turbulent jets, Journal of Fluid Mechanics 730 (2013) 559–592. doi:10.1017/jfm.2013.346.

[101] B. Callender, E. Gutmark, S. Martens, Far-field acoustic investigation into chevron nozzle mechanisms and trends, AIAA Journal 43 (1) (2005) 87–95. doi:10.2514/1.6150.

[102] B. Callender, E. Gutmark, S. Martens, Near-field investigation of chevron nozzle mechanisms, AIAA Journal 46 (1) (2008) 36–45. doi:10.2514/1.17720.

[103] K. Gudmundsson, T. Colonius, Spatial stability analysis of chevron jet profiles, 13th AIAA/CEAS Aeroacoustics Conference (28th AIAA Aeroacoustics Conference) (2007) 1–14doi:10.2514/6.2007-3599.

[104] K. Gudmundsson, Instability wave models of turbulent jets from round and serrated nozzles, Ph.D. thesis (2010).

[105] M. Koenig, Réduction de bruit de jet par injection fluidique en corps central tournant, Ph.D. thesis, Poitiers (2011).

[106] J. B. Freund, T. Colonius, Turbulence and Sound-Field POD Analysis of a Turbulent Jet, International Journal of Aeroacoustics 8 (4) (2009) 337–354. doi:10.1260/147547209787548903.

[107] C. W. Rowley, S. T. M. Dawson, Model Reduction for Flow Analysis and Control, Annual Review of Fluid Mechanics 49 (2017) 387–417. doi:10.1146/annurev-fluid-010816-060042.

[108] L. Magri, M. P. Juniper, Sensitivity analysis of a time-delayed thermo-acoustic system via an adjoint-based approach, Journal of Fluid Mechanics 719 (2013) 183–202. arXiv:1303.4267, doi:10.1017/jfm.2012.639.

[109] M. P. Juniper, R. I. Sujith, Sensitivity and Nonlinearity of Thermoacoustic Oscillations, Annual Review of Fluid Mechanics 50 (2018) 661–689. doi:10.1146/annurev-fluid-122316-045125.

[110] V. Nair, G. Thampi, S. Karuppusamy, S. Gopalan, R. I. Sujith, Loss of chaos in combustion noise as a precursor of impending combustion instability, International Journal of Spray and Combustion Dynamics 5 (4) (2013) 273–290. doi:10.1260/1756-8277.5.4.273.

[111] V. Nair, G. Thampi, R. I. Sujith, Intermittency route to thermoacoustic instability in turbulent combustors, Journal of Fluid Mechanics 756 (2014) 470–487. doi:10.1017/jfm.2014.468.

[112] J. Tony, E. A. Gopalakrishnan, E. Sreelekha, R. I. Sujith, Detecting deterministic nature of pressure measurements from a turbulent combustor, Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 92 (6) (2015) 1–11. doi:10.1103/PhysRevE.92.062902.

[113] V. R. Unni, R. I. Sujith, Multifractal characteristics of combustor dynamics close to lean blowout, Journal of Fluid Mechanics 784 (2015) 30–50. doi:10.1017/jfm.2015.567.

[114] E. Ott, Chaos in Dynamical Systems, 2002. doi:10.1017/CBO9781107415324.004.

[115] P. Cvitanović, R. Artuso, R. Mainieri, G. Tanner, G. Vattay, N. Whelan, A. Wirzba, Chaos: Classical and Quantum, ChaosBook.org (Niels Bohr Institute, Copenhagen 2005), 2005.

[116] A. N. Kolmogorov, Entropy per unit time as a metric invariant of automorphisms, in: Dokl. Akad. Nauk SSSR, Vol. 124, 1959, pp. 754–755.

[117] Y. G. Sinai, On the notion of entropy of dynamical systems, in: Doklady Akademii Nauk, Vol. 124, 1959, pp. 768–771.

[118] P. Grassberger, Generalized dimensions of strange attractors, Physics Letters A 97 (6) (1983) 227–230. doi:10.1016/0375-9601(83)90753-3.

[119] M. Katětov, On the Rényi dimension, Commentationes Mathematicae Universitatis Carolinae 27 (4) (1986) 741–753.

[120] Y. Chen, Equivalent relation between normalized spatial entropy and fractal dimension, Physica A: Statistical Mechanics and its Applications 553 (2020) 124627. arXiv:1608.02054, doi:10.1016/j.physa.2020.124627.
URL https://doi.org/10.1016/j.physa.2020.124627

[121] Y. Kuramoto, T. Tsuzuki, Persistent Propagation of Concentration Waves in Dissipative Media Far from Thermal Equilibrium, Progress of Theoretical Physics 55 (2) (1976) 356–369.

[122] G. I. Sivashinsky, On flame propagation under conditions of stoichiometry, SIAM Journal on Applied Mathematics 39 (1) (1980) 67–82.

[123] N. Platt, L. Sirovich, N. Fitzmaurice, An investigation of chaotic Kolmogorov flows, Physics of Fluids A 3 (4) (1991) 681–696. doi:10.1063/1.858074.

[124] G. J. Chandler, R. R. Kerswell, Invariant recurrent solutions embedded in a turbulent two-dimensional Kolmogorov flow, Journal of Fluid Mechanics 722.

[125] D. Lucas, R. Kerswell, Spatiotemporal dynamics in two-dimensional Kolmogorov flow over large domains, Journal of Fluid Mechanics 750 (2014) 518–554.

[126] A. R. Conn, N. I. M. Gould, P. L. Toint, Trust-Region Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.

[127] R. H. Byrd, F. E. Curtis, J. Nocedal, An inexact SQP method for equality constrained optimization, SIAM Journal on Optimization 19 (1) (2008) 351–369.

[128] W. T. Vetterling, S. A. Teukolsky, B. P. Flannery, W. H. Press, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 2002.

[129] C. Lanczos, Linear differential operators, Classics in applied mathematics ; 18, Society for Industrial and Applied Mathematics SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104, 1996.

[130] F. L. Lewis, D. L. Vrabie, V. L. Syrmos, Optimal Control, 2012. doi:10.1002/9781118122631.

[131] L. S. Pontryagin, Mathematical Theory of Optimal Processes, in: L. S. Pontryagin Selected Works, 1st Edition, Vol. 4, Taylor & Francis, London, 2017. arXiv:arXiv:1011.1669v3.

[132] R. Vishnampet, An exact and consistent adjoint method for high-fidelity discretization of the compressible flow equations, Ph.D. thesis, University of Illinois at Urbana-Champaign (2015).

[133] P. A. Thompson, Compressible-Fluid Dynamics, Advanced engineering series, McGraw-Hill, 1971.
URL https://books.google.com/books?id=LP8oAQAAMAAJ

[134] J. Capecelatro, D. J. Bodony, J. B. Freund, Adjoint-based sensitivity and ignition threshold mapping in a turbulent mixing layer, Combustion Theory and Modelling (2018) 1–33doi:10.1080/13647830.2018.1495342.

[135] K. Mattsson, M. Svärd, J. Nordström, Stable and Accurate Artificial Dissipation, Journal of Scientific Computing 21 (1).

[136] A. Isidori, Nonlinear Control Systems, 3rd Edition, Vol. 53, Springer, 2013.

[137] L. Chen, Y. Liu, Control of the Lorenz chaos by the exact linearization, Applied Mathematics and Mechanics 19 (1) (1998) 67.

[138] P. V. Kuptsov, U. Parlitz, Theory and computation of covariant Lyapunov vectors, Journal of Nonlinear Science 22 (5) (2012) 727–762. doi:10.1007/s00332-012-9126-5.

[139] B. Eckhardt, D. Yao, Local lyapunov exponents in chaotic systems, Physica D 65 (1993) 100–108.

[140] S. L. Brunton, C. W. Rowley, Fast computation of finite-time Lyapunov exponent fields for unsteady flows, Chaos 20 (1). doi:10.1063/1.3270044.

[141] L.-S. Young, Mathematical theory of Lyapunov exponents, Journal of Physics A: Mathematical and Theoretical 46 (25) (2013) 254001. doi:10.1088/1751-8113/46/25/254001.

[142] F. E. Udwadia, H. F. Von Bremen, Computation of Lyapunov characteristic exponents for continuous dynamical systems (2002). doi:10.1007/s00033-002-8146-7.

[143] F. Christiansen, P. Cvitanović, V. Putkaradze, Spatiotemporal chaos in terms of unstable recurrent patterns, Nonlinearity 10 (1) (1997) 55–70.

[144] D. Lasagna, Sensitivity analysis of turbulence using unstable periodic orbits: a demonstration on the kuramoto-sivashinsky equation, 2017.

[145] J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach, Physical Review Letters 120 (2) (2018) 24102. URL https://doi.org/10.1103/PhysRevLett.120.024102

[146] N. A. Kudryashov, Solitary and Periodic Solutions of the Generalized Kuramoto – Sivashinsky Equation, Regular and Chaotic Dynamics 13 (3) (2008) 234–238.

[147] J. M. Wallace, Space – time correlations in turbulent flow : A review, Theoretical and Applied Mechanics Letters 4 (022003) (2014) 1–16. arXiv:arXiv:1508.01258v1, doi:10.1063/2.1402203.

[148] I. E. Sarris, H. Jeanmart, D. Carati, G. Winckelmans, Box-size dependence and breaking of translational invariance in the velocity statistics computed from three-dimensional turbulent Kolmogorov flows, Physics of Fluids 19 (9). doi:10.1063/1.2760280.

[149] S. P. Han, O. L. Mangasarian, Exact penalty functions in nonlinear programming, Mathematical Programming 17 (1) (1979) 251–269. doi:10.1007/BF01588250.

[150] D. V. Anosov, Geodesic flows on closed Riemann manifolds with negative curvature, Proceedings of the Steklov Institute of Mathematics, no. 90, 1967, American Mathematical Society, Providence, 1969.

[151] R. Bowen, ω-Limit sets for Axiom A diffeomorphisms, Journal of Differential Equations 18 (2) (1975) 333–339. doi:10.1016/0022-0396(75)90065-0.

[152] C. Grebogi, S. M. Hammel, J. A. Yorke, T. Sauer, Shadowing of physical trajectories in chaotic dynamics: Containment and refinement, Physical Review Letters 65 (13) (1990) 1527–1530. doi:10.1103/PhysRevLett.65.1527.

[153] W. B. Hayes, K. R. Jackson, Rigorous Shadowing of Numerical Solutions of Ordinary Differential Equations by Containment, SIAM Journal of Numerical Analysis 41 (5) (2003) 1948–1973.

[154] G. Gallavotti, E. G. D. Cohen, Dynamical ensembles in stationary states, Journal of Statistical Physics 80 (5-6).

[155] G. Gallavotti, E. G. D. Cohen, Dynamical ensembles in nonequilibrium statistical mechanics, Physical Review Letters 74 (14).

[156] W. L. Hallauer, Introduction to Linear, Time-Invariant, Dynamic Systems for Students of Engineering, Virginia Polytechnic Institute and State University, 2021.
URL https://eng.libretexts.org/@go/page/7620

[157] M. F. Golnaraghi, Automatic control systems, tenth edition. Edition, McGraw-Hill Education, New York, 2017.

[158] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, Classification and regression trees, CRC press, 1984.

[159] C. E. Shannon, A Mathematical Theory of Communication, ACM SIGMOBILE mobile computing and communications review 5 (1) (2001) 3–55. doi:10.1002/j.1538-7305.1968.tb00069.x.

[160] K. E. Petersen, Ergodic theory, Cambridge University Press, 1989.

[161] C. S. Hsu, M. C. Kim, Construction of maps with generating partitions for entropy evaluation, Physical Review A 31 (5) (1985) 3253–3265. doi:10.1103/PhysRevA.31.3253.

[162] Ruelle, David, Chaotic Evolution and Strange Attractors, Vol. 173, Cambridge University Press, 1989.

[163] Y. B. Pesin, Lyapunov characteristic exponents and ergodic properties of smooth dynamical systems with an invariant measure, in: Doklady Akademii Nauk, Vol. 226, Russian Academy of Sciences, 1976, pp. 774–777.

[164] D. Clark, L. Tarra, A. Berera, Chaos and information in two-dimensional turbulence, Physical Review Fluids 5 (6) (2020) 1–18. arXiv:2003.08159, doi:10.1103/physrevfluids.5.064608.

[165] A. Berera, D. Clark, Information production in homogeneous isotropic turbulence, Physical Review E 100 (4) (2019) 41101. doi:10.1103/PhysRevE.100.041101.
URL https://doi.org/10.1103/PhysRevE.100.041101

[166] G. Benettin, L. Galgani, A. Giorgilli, J. M. Strelcyn, Lyapunov Characteristic Exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. Part 1: Theory, Meccanica 15 (1) (1980) 9–20. doi:10.1007/BF02128236.

[167] L. D. Landau, E. M. Lifshitz, Fluid Mechanics, 2nd Edition, Pergamon Press, 1959. doi:10.1017/cbo9780511497513.013.

[168] P. Constantin, C. Foias, O. P. Manley, R. Temam, Determining modes and fractal dimension of turbulent flows, Journal of Fluid Mechanics 150 (1985) 427–440. doi:10.1017/S0022112085000209.

[169] J. D. Gibbon, E. S. Titi, Attractor dimension and small length scale estimates for the three-dimensional Navier-Stokes equations, Nonlinearity 10 (1) (1997) 109–119. doi:10.1088/0951-7715/10/1/007.

[170] J. L. Kaplan, J. A. Yorke, Chaotic behavior of Multidimensional difference equations, in: Functional Differential Equations and Approximation of Fixed Points, 1979, pp. 204–227. arXiv:arXiv:1011.1669v3.