Numerical Differentiation of Stationary Measures of Chaos

by

Angxiu Ni

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Mathematics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor John Strain, Chair
Professor Fraydoun Rezakhanlou
Professor Fai Ma

Spring 2021

Numerical Differentiation of Stationary Measures of Chaos

Abstract

Numerical Differentiation of Stationary Measures of Chaos

by

Angxiu Ni

Doctor of Philosophy in Mathematics

University of California, Berkeley

Professor John Strain, Chair

In this thesis we develop two algorithms, the non-intrusive shadowing and the fast linear response algorithms, for computing derivatives of SRB measures with respect to some parameters of the dynamical system, where SRB measures are fractal limiting stationary measures of chaotic systems. The accurate formula of such derivative is the linear response formula. The non-intrusive shadowing algorithm previously devised by the author is one of the fastest approximate algorithm for differentiating chaos. It restricts computations to the unstable subspace, whose dimension can be much lower than the system.

In this thesis, we first set the theoretical foundation of the shadowing method by showing that it accurately computes the shadowing contribution of the linear response formula, and it well approximates the entire linear response for some important cases, such as high-dimensional systems with low-dimensional unstable subspaces. Then we develop the finite difference version of the non-intrusive shadowing, which we demonstrate on a computational fluid problem with about a million of dimensions: this is one of the first time sensitivity analysis being performed on such complicated systems. Then we consider adjoint shadowing algorithms. We give the explicit formula of the adjoint operator, and the non-intrusive characterization by only adjoint solutions. This leads to the non-intrusive adjoint shadowing algorithm, which generalizes the traditional back-propagation method to chaos. We demonstrate non-intrusive adjoint shadowing in a high-dimensional computational fluids problem.

Finally, we devise the fast linear response algorithm, for accurately computing the other part of the linear response, which is called the unstable contribution. We derive the first computable expansion formula of the unstable divergence, a central object in the linear response theory for fractal attractors. Then we give a 'fast' characterization of the expansion by renormalized second-order tangent equations, whose second derivative is taken in a modified shadowing direction, computed by the non-intrusive shadowing algorithm. The new characterization makes the algorithm efficient and robust: its main cost is solving u, the unstable dimension, many first-order and second-order tangent equations, and it does

not compute oblique projections. Moreover, the algorithm works for chaos on Riemannian manifolds with any u; its convergence to the true derivative is proved for uniform hyperbolic systems. The algorithm is illustrated on an example which is difficult for previous methods. The procedure list is easy to understand and implement.

To my parents, Xiuming Liu and Guoqiang Ni.

Contents

List of Figures

List of Tables

Acknowledgments

I am extremely grateful to my adviser, John Strain, for supporting me carrying out this research and supervising my thesis. Without his help, my research would have never been accomplished. I also deeply thank committee members, Fraydoun Rezakhanlou and Fai Ma, for very kind supervision and very helpful discussions.

I am in much debt to the math department of UC Berkeley, which took me in and let me independently pursue my original interest in numerical differentiation of chaos. I have benefited a lot from talking to the best mathematicians in the world, from whom I learned not only math but also how to live and work as a professional mathematician. Among them, I thank in particular Alexandre Chorin, Per-Olof Persson, Lin Lin, Jon Wilkening, and Craig Evans. The free spirit of Berkeley has always encouraged me to work as an independent mathematician.

My research requires much knowledge in pure math, for which I have benefited from a lot of people. I am in much debt to Yi Lai for her constant support and encouragements, and many discussions on differential geometry. I bothered a lot of mathematicians on smooth ergodic theory, who have kindly and patiently helped me. I am extremely grateful to David Ruelle, Dmitry Dolgopyat, and Miaohua Jiang for discussions on their groundbreaking results on linear response. I am also extremely grateful to Charles Pugh and Peidong Liu for help on more general topics. I also learned a lot at the online student dynamics seminar at Penn State during the covid year, especially from Alansari Nawaf, Alp Uzman, and Ignacio Correa Duran.

I have made many friends over the past four years, who gave me much personal and professional support. Andrew Shi introduced me to the Berkeley environment; with Chan Bae and Charles Wang we formed the 'party four'. I would also like to thank other extremely valuable friends, Yixuan Li, Xiaohan Yan, Haoren Xiong, Yu Tong, Wenlong Mou, Huibin Chen, Roy Zhao, Yuan Yao, Jian Wang, Xinyu Zhao, and my beloved office-mates, Theo McKenzie and Lauren Heller.

I am also extremely grateful to many people during my two master degrees at Tsinghua and MIT. I am very grateful to Yufei Zhang, Haixin Chen, and Song Fu, for introducing me to numerical optimization, computational fluid, and turbulence. I am also very grateful to to Qiqi Wang for introducing me to chaos. I also greatly thank my friends, including Wensheng Zhang, Caolei Li, Lijun Xu, Zhao Li, Kezhe Xu, Tong Zhao, Yuandong Zhao, and Xiaoming Fang at Tsinghua, and Pablo Fernandez, Nisha Chandramoorthy, Zheng Wang, Ben Zhang, Arthur Huang, and Ophelia Liang at MIT. I am also extremely grateful to Youjin Zhang, Richard Melrose, and Michael Artin, who taught me math, and inspired me to become a mathematician.

Finally, I thank my parents, Xiuming Liu and Guoqiang Ni, for supporting me through this journey. They are the best parents in the world.

Chapter 1

Introduction

Chaos is ubiquitous across many disciplines, such as fluid mechanics, geophysics, and machine learning. In chaotic systems, while instantaneous snapshots seem random and unpredictable, the averaged behavior is deterministic, and can be predicted using the parameters of the system. This means that the averaged behavior of chaos, measured by the average of some objective functions, varies smoothly to the parameters of the system, and the derivative is well-defined. The derivative, or linear response, of the long-time average of observable/objective functions, is fundamental to many widely used numerical tools, such as gradient-based optimization, error analysis, and uncertainty quantification. However, due to the butterfly effect, computing the linear response is challenging, especially for general fractal attractors, where many quantities are non-differentiable.

Roughly speaking, for a chaotic dynamical system, if we make a perturbation on the governing equation, for each trajectory, we typically get a new trajectory that diverges exponentially fast from the old one. However, if we further average the perturbations on individual trajectory to the SRB measure, which models the long-time averaged statistics of chaotic systems, we can formally obtain the linear response formula. It is proved to give the true derivative of averaged objective with respect to the perturbation on governing equations, under various hyperbolic assumptions [68, 70, 69, 42, 27], where hyperbolicity is typically used as a model for general chaotic systems.

Numerically, the original linear response formula can be directly implemented, as done by the ensemble and operator-based algorithms [47, 28, 49, 38, 3]. Theoretically, these algorithms could give accurate linear response; however, both are very expensive, and in reality, it is typically unaffordable for convergence to actually happen [19]. This is because trajectory-wise perturbations grow exponentially fast, and that averaging a large integrand to get a small integration result requires a lot of data. It is also possible to approximate the dynamical systems by Markov chains, and use linear response for Markov chains as an approximation [72].

The shadowing method was another attempt to compute the derivative of averaged objectives for chaotic systems [2, 12, 63, 82, 74, 39, 7, 73, 46, 9]. The computational efficiency of shadowing algorithms is boosted by a 'non-intrusive' formulation devised by the author,

which means to involve only solutions of the most basis equations, in this case the tangent equations [59, 61]. Here tangent equations are linearized versions of the governing equations of the dynamical system. The benefit of non-intrusiveness is to constrain the computation to only the unstable subspace, and makes approximate numerical differentiation affordable for the first time for several high dimensional chaos, such as computational fluid systems with 4×10^6 degrees of freedom [57, 41].

However, shadowing method was developed under the strong assumption that shadowing trajectories are representative. This is not true in general, and shadowing methods can fail for simple systems. The goal of chapter 3 is to rebuild the theoretical foundation of shadowing methods. We will show that shadowing gives part of the correct derivative, which we call the shadowing contribution of the linear response; the other part is called the unstable contribution. Moreover, we show that shadowing is a good approximation for many interesting cases. In particular, under two statistical assumptions, one on the randomness of selecting objective functions and perturbations of dynamical system, the other on the decay of correlation, we show the error of shadowing is $O(\sqrt{u/M})$, where u is the unstable dimension, M is the system dimension. We will also prove the convergence of the non-intrusive shadowing algorithm to the shadowing contribution.

The original (tangent) non-intrusive shadowing algorithm requires solvers of tangent equations, or linearized governing equations. Chapter 4 presents the finite-difference version of non-intrusive shadowing. It does not require tangent solvers, and can be implemented with little modification to existing numerical simulation software. This enriches applications of finite-difference shadowing to engineering problems, since most numerical simulation software do not have accompanying tangent solvers. We also give a formula for solving the least-squares problem in finite-difference shadowing, which can be applied in non-intrusive shadowing as well. Finally, we apply finite-difference shadowing for sensitivity analysis of a chaotic flow over a 3-D cylinder at Reynolds number 525.

The marginal cost for a new parameter in non-intrusive shadowing is computing one extra inhomogeneous tangent solution. For cases with many parameters, an adjoint version is desired, because its computational cost is independent of the number of parameters. Chapter 5 presents the adjoint version of the non-intrusive shadowing algorithm, which performs adjoint sensitivity analysis of chaotic systems via computing the adjoint shadowing direction. Non-intrusive adjoint shadowing constrains its computation to the adjoint unstable subspace, and can be implemented with little modification to existing adjoint solvers. We also demonstrate the adjoint shadowing algorithm on a three-dimensional weakly turbulent flow.

In the linear response formula, the shadowing contribution has a bounded integrand given by the shadowing direction; the exponential growth of integrand is only in the unstable contribution. Non-intrusive shadowing well computes the shadowing direction and the shadowing contribution, and the left over problem, which is a long-lasting one, is to compute the unstable contribution. The exponential growth in the integrand of the unstable contribution is improved after integrating-by-parts on the unstable manifold, which yields a divergence

on the unstable manifold. However, this unstable divergence is very difficult to compute, because it is *a priori* only a distribution.

Chapter 6 presents the first numerical treatment of the unstable divergence. We derive the first computable expansion formula of the unstable divergence, where all terms are functions rather than distributions. Then we give a 'fast' characterization of the expansion by renormalized second-order tangent equations, whose second derivative is taken in a modified shadowing direction, computed by the non-intrusive shadowing algorithm. The main idea here is a 'combine then propagate' idea commonly seen in fast algorithms. This new characterization leads to the fast linear response algorithm, which is efficient and robust: its main cost is solving u, the unstable dimension, many first-order and second-order tangent equations, and it does not compute oblique projections. The algorithm is illustrated on an example which is difficult for previous methods. The procedure list is easy to understand and implement.

Chapter 2

Preliminaries

2.1 Hyperbolicity and notations

Let $f \in C^3$ be a diffeomorphism on a C^∞ Riemannian manifold $\mathcal{M}$, whose dimension is M. Assume that K is a hyperbolic compact invariant set, that is, $T_K M$ (the tangent bundle restricted to K) has a continuous f_*-invariant splitting $T_K M = V^s \bigoplus V^u$, such that there are constants $C > 0$, $\lambda < 1$, and

$$\max_{x \in K} \|f_*^{-n}|V^u(x)\|, \|f_*^n|V^s(x)\| \leq C\lambda^n \quad \text{for } n \geq 0,$$

Here f_* is the pushforward operator, which applies on vectors or measures. In $\mathbb{R}^M$, f_* can be represented by multiplying with the Jacobian matrix. (see definition 5 in appendix 6.5). We call V^u and V^s the stable and unstable subbundles, or subspaces; also let u, s denote the dimension of the unstable and stable manifolds, hence, $u + s = M$. For convenience, we also use superscript $+, -$ to denote unstable and stable subspaces, that is,

$$V^+ = V^u, \quad V^- = V^s.$$

Local stable manifolds, $\mathcal{V}^s(x)$, are manifolds as smooth as f, tangent to $V^s(x)$ at x, and there are $C > 0$ and $\lambda < 1$ such that if $y, z \in \mathcal{V}^s(x)$,

$$d(f^n y, f^n z) \leq C\lambda^n d(y, z) \quad \text{for } n \geq 0,$$

where d is the distance function. Local unstable manifolds are defined similarly.

K is said to be an Axiom A attractor if it is the closure of periodic orbit, and there is an open neighborhood U, called the basin of the attractor, such that $\cap_{n \geq 0} f^n U = K$. The SRB measure on an attractor can be easily found by numerical simulations, since it is the weak limit of the empirical measure of almost all trajectories starting from the basin. For all $x \in K$, the local unstable manifolds $\mathcal{V}^u(x)$ lie in K, whereas the local stable manifolds $\mathcal{V}^s(x)$ fill a neighborhood of K. For more details on hyperbolicity see [71, 75]. Due to the spectral

decomposition theorem, by taking out a basic set and raise f to some power, we may further assume that f is mixing on K [11, 77].

On a mixing Axiom A attractor, the SRB measure ρ of f on K is the unique f-invariant measure making $h(\rho) - \rho(\log J^u)$ maximum. Here $h(\rho)$ is the entropy of ρ and J^u is the unstable Jacobian defined in equation (6.3). Under our assumption, the SRB measure can be equivalently characterized by [87]:

- ρ has absolutely continuous conditional measures on unstable manifolds;
- there is a set $V \subset \mathcal{M}$ having full Lebesgue measure such that for every continuous observable $\varphi : \mathcal{M} \to \mathbb{R}$, we have, for every $x \in V$

$$\frac{1}{n}\sum_{i=0}^{n-1} \varphi(f^i x) \to \rho(\varphi).$$

Roughly speaking, the last characterization means the SRB measure is the limit of evolving Lebesgue measure, and it is the invariant measure most compatible with volume when volume is not preserved.

We explain some notation conventions used in this chapter. Let $x \in \mathcal{M}$ be the point of interest, and $x_k := f^k x$. Subscripts n, k, m are only for labeling steps, and subscripts for step zero are omitted. For a tensor field X on $\mathcal{M}$, let X_k be the pullback of X by f^k,

$$X_k(x) := X(x_k).$$

The subscript k always specify that the value of the tensor field is taken at x_k. However, when X_k is differentiated, we should further specify its domain, or when the pullback happens: we leave that to be determined by the differentiating vector. For example, let ∇ be the Riemannian connection, Y a vector field, then

$$(\nabla_{Y_k} X_k)(x) := (\nabla_Y X)(x_k). \tag{2.1}$$

Since $Y_k(x) \in T_{x_k}\mathcal{M}$, it must differentiate a tensor field at x_k, so X_k must be a function around x_k. Hence, X_k is X when being differentiated, and then the entire result is pulled-back to x. The other way does not work: if X_k is $X \circ f^k$ when being differentiated, this is a function of x, and can not be differentiated by Y_k. For another example, take a differentiable observable function φ on $\mathcal{M}$,

$$f_*^k Y(\varphi_k)(x) := f_*^k Y(\varphi)(x_k) = Y(\varphi \circ f^k)(x) =: Y(\varphi_k)(x), \tag{2.2}$$

where $Y(\cdot)$ means to differentiate in the direction of Y. Since $f_*^k Y(x) \in T_{x_k}\mathcal{M}$, the first φ_k is φ when the differentiation happens at x_k, and then the result is pulled-back to x; since $Y(x) \in T_x\mathcal{M}$, the second φ_k is $\varphi \circ f^k$ when the differentiation happens right at x. Finally, we omit the step subscript of the pushforward tensor f_*, since its location is well-specified by the vector it applies to.

Given a trajectory $\{x_n\}_{n\in\mathbb{Z}}$, we can define a sequence, say $\{w_n(x_0, w_0) \in T_{x_n}\mathcal{M}\}_{n\in\mathbb{Z}}$. We say such a sequence is covariant if its evaluation commutes with evolutions; that is, it does not depend on w_0 and further

$$w_n(x_0) = w_0(x_n).$$

Sequences obtained by taking values of a (rough) tensor field, such as X_n, X_n^u, Φ_n, p_n, are covariant. We can define non-covariant sequences by the inductive relation of some covariant sequences. The main theorem in our chapter is that, due to the stability in the induction, the non-covariant sequence, r, computed by the induction, well approximates the covariant counterpart, p, which is originally defined by complicated formulas. For other cases where the non-covariant approximations have been previously well understood, we use the same notation for both the covariant and non-covariant versions, such as the shadowing direction v and the unstable hyper-cube e.

2.2 Review of linear response formula

The linear response formula is an expression of $\delta\rho$, the derivative of the SRB measure, using δf. Ruelle gave a formal derivation of this formula by averaging perturbations on each trajectories over the SRB measure, and proved that the formula indeed gives the derivative [68]. The assumption made in Ruelle's work is also the setting of our current chapter. The linear response formula can be proved for more general cases, for example, Dolgopyat proved it for the partially hyperbolic systems with some mixing conditions [27]. It is plausible that our work can also be generalized, perhaps even more so, because the fast characterization does not involve oblique projections. It should also be noted that the linear response formula fails for some cases [4, 85].

Theorem 1 (Ruelle [68])**.** *Let K be a mixing Axiom A attractor for the C^3 diffeomorphism f of $\mathcal{M}$. Denote by ρ the SRB measure of f on K, $\mathcal{A}$ the space of C^3 diffeomorphisms sufficiently close to the f in a fixed neighborhood V of K, $\mathcal{A}$ has C^3 topology. Then the map $f \mapsto \rho$ is C^1 from $\mathcal{A}$ to $C^2(\mathcal{M})^*$. Further assume that f is parameterized by some scalar γ, and $\gamma \mapsto f$ is C^1 from $\mathbb{R}$ to $\mathcal{A}$, and define*

$$\delta(\cdot) := \frac{\partial(\cdot)}{\partial\gamma}. \tag{2.3}$$

δ can also be thought as finite perturbations caused by perturbing the governing equation of the dynamical system. Then, the derivative of the SRB measure is given by:

$$\delta\rho(\Phi) = \sum_{n=0}^{\infty} \rho\left(X(\Phi_n)\right) = \lim_{W\to\infty} \rho\left(\sum_{n=0}^{W} X(\Phi_n)\right),$$

where $X := \delta f \circ f^{-1}$ is a C^3 vector field, $X(\cdot)$ is to differentiate in the direction of X, and $\Phi \in C^2$ is a fixed observable function. Here $\Phi_n = \Phi \circ f^n$.

We refer to the particular formula in theorem 1 as the original linear response formula. Due to the existence of unstable components, the size of the integrand grows exponentially to W:

$$\sum_{n=0}^{W} X(\Phi_n) \sim O(\lambda_{max}^W),$$

where $\lambda_{max} > 1$ is the largest Lyapunov exponent. This phenomenon is also known as the 'exploding gradients'. Computationally, the number of samples requested to evaluate the integration to ρ also grows exponentially to W, incurring large computational cost [19]. Hence, algorithms based on the original linear response formula suffer from very high computational cost. Theoretically, it is also hard to justify the convergence of the summation in the original formula in theorem 1.

To prove the linear response formula actually gives the correct linear response, Ruelle decomposed the linear response into two parts. The first part accounts for the change of the location of the attractor via the conjugacy map, which is called the shadowing contribution. The second part, the unstable contribution, accounts for the fact that the pushforward of the old SRB onto the new attractor is no longer SRB. Integrate-by-parts the unstable contribution on the unstable manifold gives (see section 6.1 for details)

$$\begin{aligned} \delta\rho(\Phi) = S.C. - U.C. &= \rho(v(\Phi)) - \rho\left(\left(\sum_{n\in\mathbb{Z}} \Phi_n\right) \operatorname{div}_\sigma^u X^u\right), \\ \text{where} \quad S.C. &:= \sum_{n\geq 0} \rho(X^s(\Phi_n)) - \sum_{n\leq -1} \rho(X^u(\Phi_n)) = \rho(v(\Phi)), \\ U.C. &:= \sum_{n\in\mathbb{Z}} \rho\left(X^u(\Phi_n)\right) = \lim_{W\to\infty} \rho\left(\sum_{n=-W}^{W} \Phi_n \operatorname{div}_\sigma^u X^u\right). \end{aligned} \tag{2.4}$$

Here $S.C.$, $U.C.$ are shadowing and unstable contributions, v is the shadowing direction (see section 2.3), X^u and X^s are the unstable and stable oblique projections of X (appendix 6.5 figure 6.7). The unstable divergence, $\operatorname{div}_\sigma^u$, is the divergence on the unstable manifold under the conditional SRB measure. Now the unstable contribution subjects to decay of correlations, justifying its convergence.

We refer to equation (2.4) as the integrated-by-parts linear response formula. For the stable contribution, the integrand v is bounded. For the unstable contribution, notice that subtracting Φ by constant $\rho(\Phi)$ does not change the linear response, so

$$\psi \sim O(\sqrt{W}), \quad \text{where} \quad \psi := \sum_{m=-W}^{W} (\Phi_m - \rho(\Phi)).$$

Here $\sim$ means with large probability. Hence, the integrand of the integrated-by-parts formula is much smaller than the original formula, and algorithms based on the integrated-by-parts formula should have much faster convergence. However, accurately computing the unstable

divergence on fractal attractors has been an lasting open problem, not to mention algorithms that are both accurate and efficient: this is achieved in chapter 6.

A perhaps more familiar decomposition of linear response is to cancel the second summation in the shadowing contribution with half of terms in the unstable contribution, which gives a decomposition into stable and unstable contributions. However, computing the stable contribution requires computing oblique projection operators, which is twice the cost of computing the shadowing direction by the non-intrusive shadowing algorithm. Moreover, computing projections is not robust, because it requires making the artificial decision whether a Lyapunov vector is stable or unstable. Finally, for both decompositions, our algorithm for the unstable contribution is the same, which also requires computing a modified shadowing direction. Hence, there is no point in using a stable/unstable decomposition, since shadowing directions are computed anyway.

Another remark is that our results may hold beyond the strong assumptions on hyperbolicity. The two things that theorem 6 and the algorithm actually depend on are the integrability of shadowing directions and renormalized second-order tangent solutions, which should be more abundant in applications than the strong hyperbolicity assumptions. Our work is essentially two new forms of the linear response formula, an expansion and a fast characterization, and their equivalence to other forms can be proved more easily than the equivalence between the formula and the true derivative. Hence, there is a good chance that our results are correct whenever the linear response formula gives the correct sensitivity. It is mainly for the simplicity of discussions that we use strong assumptions on hyperbolicity.

2.3 Shadowing direction and non-intrusive shadowing algorithm

We define inhomogeneous and homogeneous tangent solutions on a trajectory $\{x_n\}_{n\geq 0}$ as non-covariant sequences $\{v'_n \in T_{x_n}\mathcal{M}\}_{n\geq 0}$ and $\{w_n \in T_{x_n}\mathcal{M}\}_{n\geq 0}$ which satisfy the inhomogeneous and homogeneous tangent equations,

$$v'_{n+1} = f_* v'_n + X_{n+1}, \quad w_{n+1} = f_* w_n. \tag{2.5}$$

Roughly speaking, v and w describe the perturbation of the trajectory due to perturbation on the governing equation and on the initial conditions, respectively.

Roughly speaking, a 'non-intrusive' algorithm, in our definition, is one which requires only the solutions of the most basic inductive relations. The non-intrusive shadowing algorithm requires only the tangent solutions as data. There are several ramifications of non-intrusiveness, making the algorithm efficient and robust.

1. Non-intrusive algorithms do not compute oblique projections (appendix 6.5 figure 6.7), since the efficient algorithm for oblique projections is 'little intrusive', which means that it requires adjoint solvers in addition to tangent ones.

2. No additional information on the tangent equation, such as the Jacobian matrices, is needed. Only the solutions are needed.

3. The instability of the tangent equation is removed by some operations using only tangent solutions, for example, by subtracting homogeneous solutions from inhomogeneous solutions.

4. Once the instability is suitably removed, precise initial condition is not necessary.

The basic solutions of non-intrusive algorithms may vary in different context. The non-intrusive formulation for the unstable contribution, to be discussed later in this chapter, allows both first and second order tangent solutions. In the adjoint shadowing lemma and algorithm, non-intrusiveness is defined by adjoint solutions [54, 58]. The 'little-intrusive' algorithm is a side result of adjoint shadowing lemma, and it can help other algorithms, such as the blended response, to improve efficiency.

We review the non-intrusive shadowing algorithm [59]. By invariance of the SRB measure, we can show that the shadowing contribution,

$$S.C. = \rho(v(\Phi)), \quad \text{where } \; v := \sum_{k\geq 0} f_*^k X_{-k}^s - \sum_{k\geq 1} f_*^{-k} X_k^u.$$

Here v is the shadowing direction, $X_k(x) \in T_{x_k}M$, and $f_*^{-k}X_k^u(x) \in T_xM$, hence $v(x) \in T_x\mathcal{M}$, and $v(\cdot)$ denotes taking derivative in the direction of v. This definition of shadowing direction explicitly involves the stable and unstable projections. Using the exponential growth of homogeneous tangent solutions, we can give a characterization which does not involve oblique projections.

Lemma 1. *v is the only bounded inhomogeneous tangent solution.*

This characterization involves only the tangent solution, urging us to look for a non-intrusive algorithm. On a given trajectory $\{x_n\}$, the non-intrusive shadowing algorithm computes the shadowing direction, as if a non-covariant sequence, $v_n := v(x_n)$, by the constrained minimization,

$$\min_{a\in\mathbb{R}^u} \frac{1}{2N} \sum_{n=0}^{N-1} \|v_n\|^2, \quad \text{s.t. } \; v_n = v'_n + \underline{w}_n a\,, \tag{2.6}$$

where v' is an arbitrarily chosen inhomogeneous tangent solution, such as one solved from the zero initial condition. $\underline{w}$ is a matrix with u columns, which are homogeneous tangent solutions. For almost all random initial conditions of W, its unstable projection span the unstable subspace V^u; hence, after evolving for some time, the unstable components outgrow the stable ones, and the span of W would well approximate V^u.

In other words, the boundedness property is approximated by a minimization, while the feasible space of all inhomogeneous tangent solutions, which has M dimensions, is reduced to

an affine subspace of only u dimensions, almost parallel to V^u. The reduced feasible space significantly reduces the computational cost, yet is still capable of finding the shadowing direction [55]. The non-intrusive shadowing algorithm is an ingredient of fast linear response algorithm, and its procedure list is included in section 6.3.

If the unstable contribution is small, we may choose to neglect it or approximate it crudely. For example, for systems with exponential decay of correlations, and some independence between X and Φ, the unstable contribution is typically in the order of $\sqrt{u/M}$. Hence, when $u \ll M$, we may approximate linear response by the non-intrusive shadowing algorithms. It is possible to further compute some unstable contributions using the little-intrusive algorithm for projection operators [55]. Since the non-intrusive shadowing algorithm is part of the fast linear response, it does not hurt to first try non-intrusive shadowing algorithms, which are faster and maybe accurate enough; if non-intrusive shadowing is not enough, we may further compute the unstable contribution.

2.4 Continuous-time systems

In this thesis, continuous-time systems are limited to $\mathbb{R}^M$, although generalization to general manifolds is straightforward. We consider a chaotic dynamical system given by the vector field f, that is, a trajectory $x(t)$ satisfies the so-called primal equation:

$$\frac{dx}{dt} = f(x). \tag{2.7}$$

Again assume that f is parameterized by γ. The SRB measure is still defined, in this thesis, as the infinite pushforward of Lebesgue measure by the flow. Define $X := \delta f := \partial f/\partial\gamma$.

Homogeneous solutions, w, and inhomogeneous tangent solutions, v, are

$$\frac{dw}{dt} = \partial_x f w, \quad \frac{dv}{dt} = \partial_x f v + X \tag{2.8}$$

where $\partial_x(\cdot) = \partial(\cdot)/\partial x$.

For continuous-time system, uniform hyperbolicity requires that the tangent space on the attractor can be split into stable subspace, unstable subspace, and a central subspace of dimension one, which is parallel to f. This central direction is inevitable, because f is a homogeneous tangent solution whose norm keeps roughly at the same level. The central direction does not pose too much difficulty for theoretical analysis, since we can consider finite-time maps of the flow [13]; however, for computations, we need a neater treatment.

A Characteristic Lyapunov Vector (CLV), $\zeta(t)$, is a homogeneous tangent solution whose norm behaves like an exponential function of time. That is, there are $C_1, C_2 > 0$ and $\lambda \in \mathbb{R}$, such that for any $t \in \mathbb{R}$,

$$C_1 e^{\lambda t}\|\zeta(0)\| \le \|\zeta(t)\| \le C_2 e^{\lambda t}\|\zeta(0)\|, \tag{2.9}$$

where the norm is the Euclidean norm in $\mathbb{R}^m$, and λ is defined as the Lyapunov Exponent (LE) corresponding to this CLV. CLVs with positive LEs are called unstable, CLVs with negative LEs are called stable, and with zero LEs are neutral. In this thesis, the j-th largest LE and its corresponding CLV will be referred as the j-th LE and j-th CLV, respectively.

The shadowing direction v^∞ is an inhomogeneous tangent solution, and its orthogonal projection perpendicular to the trajectory, $v^{\infty\perp}$, is uniformly bounded on a infinitely long trajectory. The orthogonal projection $p^\perp(t)$ of some any vector valued function of time, $p(t)$, is:

$$p^\perp(t) = p(t) - \frac{f^T(t)p(t)}{f^T(t)f(t)} f(t) \,, \tag{2.10}$$

where $\cdot^T$ is the matrix transpose.

For continuous dynamical systems, the base and the shadowing trajectories may move at different speed. Since we are considering averages taken with respect to time, we should take account of the fact that if the shadowing trajectory spend longer or shorter time in a particular neighborhood, then the weight of the objectives in this neighborhood should be respectively larger or smaller. We define a 'time dilation' terms η to denote this effect, and if the shadowing trajectory takes less time to travel the same length in the phase space, then $\eta < 0$. On the other hand, if the shadowing trajectory moves slower, then $\eta > 0$. We can show that for a given inhomogeneous tangent solution v, which describes a perturbation on the trajectory due to parameter change, η should satisfy the following equation [59]:

$$\frac{dv^\perp}{dt} = \partial_x f v^\perp + \delta f + \eta f \,. \tag{2.11}$$

We denote the particular time dilation corresponding to the shadowing direction v^∞ by η^∞.

The change in the location of the attractor is described by v^∞; the time difference the shadowing trajectory spend around a neighborhood is described by a corresponding η^∞. Taking both changes into account, the shadowing contribution of the linear response is

$$S.C. = \frac{1}{T} \int_0^T \left[\partial_x J \, v^\perp + \delta J + \eta (J - J_{avg}) \right] dt \,, \tag{2.12}$$

where we assume that v and η are legit approximations of v^∞ and η^∞. Another formula for the sensitivity is easier for computer programming:

$$S.C. = \frac{1}{T} \left[\int_0^T \left(\partial_x J \, v + \delta J \right) dt + \xi \Big|_0^T J_{avg} - (\xi J) \Big|_0^T \right] , \tag{2.13}$$

where the time difference term, ξ, is a time-dependent scalar function such that:

$$\xi f = v - v^\perp \,. \tag{2.14}$$

Intuitively, the right-hand-side of the above equation is how farther down the trajectory direction has the shadowing trajectory traveled. Divided by f, we can see ξ describes how

much more time should the base trajectory take to catch up with the shadowing trajectory. ξ is easier to compute than η, since its definition does not involve time derivatives. Notice that in equation (2.13), we use v instead of its projection $v^{\perp}$. The detailed derivation of equation (2.12) and (2.13) can be found in the appendix of [59].

The non-intrusive shadowing problem on a given trajectory is a least squares problem with arguments $a \in \mathbb{R}^u$,

$$\min_{a \in \mathbb{R}^u} \frac{1}{2} \int_0^T (v'^{\perp} + W^{\perp} a)^T (v'^{\perp} + W^{\perp} a)\, dt, \tag{2.15}$$

Here v' is an inhomogeneous tangent solution, $W(t)$ is a matrix whose columns are orthogonal projections of randomly initialized homogeneous tangent solutions, $W^{\perp}(t) = [w_1^{\perp}(t), \cdots, w_M^{\perp}(t)]$. The shadowing solution is given by $v = v^* + Wa$, which is an inhomogeneous tangent solution, but we replace prescribing its initial condition by minimizing its L^2 norm. We denote the covariant matrix, the coefficient matrix for the second order terms, by C; we denote the coefficient vector for the first order terms by d. More specifically, we define

$$C = \int_0^T (W^{\perp})^T W^{\perp}\, dt\,, \quad d = \int_0^T (v^{*\perp})^T W^{\perp}\, dt\,. \tag{2.16}$$

Now the non-intrusive shadowing problem is equivalently written as

$$\min_{a \in \mathbb{R}^M} \frac{1}{2} a^T C a + d^T a\,. \tag{2.17}$$

Chapter 3

Approximating linear response by non-intrusive shadowing algorithms

Shadowing methods approximately compute derivatives of averaged objectives of chaos with respect to parameters of the dynamical system. However, previous convergence proofs of shadowing methods wrongly assume that shadowing trajectories are representative. In contrast, the linear response formula is proved rigorously, but is more difficult to compute.

In this chapter, we first prove that the shadowing method computes a part of the linear response formula, which we call the shadowing contribution. Then, under two statistical assumptions, one on the randomness of selecting objective functions and perturbations of dynamical system, the other on the decay of correlation, we show the error of shadowing is $O(\sqrt{u/M})$, where u is the unstable dimension, M is the system dimension. For partly reducing this error, we give a correction which can be easily implemented. Finally, we prove the convergence of the non-intrusive shadowing, the fastest shadowing algorithm, to the shadowing contribution.

The shortcoming of the shadowing method is that it makes the strong assumption that shadowing trajectories are representative. This is not true in general, and shadowing methods can fail for simple systems such as the 1-dimensional expanding circle [8]. Hence, it is of interest to rebuild the theoretical foundation of shadowing methods. As we shall see in this chapter, shadowing method does not give the accurate exact derivative, yet, we can show that it gives part of the correct derivative, which we call the shadowing contribution of the linear response. Moreover, we show that shadowing is a good approximation for many interesting cases. This partially explains the success of shadowing in fluid mechanics.

It is hence of interest to ask how much error is caused by this reduction. This is answered in the later part of this chapter, where we show that this reduction causes no more error comparing to original shadowing methods. Together with the first part of the chapter, we give an error analysis of approximating linear response by non-intrusive shadowing.

Moreover, this chapter is the first step towards the linear response algorithm. This chapter shows that the linear response can be decomposed into the shadowing contribution and the unstable contribution. Computing the unstable contribution is solved by the linear response

algorithm, via a new characterization by second-order tangent equations, whose second derivative is taken in a modified shadowing direction [56]. The linear response algorithm is accurate, and faster than most previous algorithms except the non-intrusive shadowing. Linear response algorithm uses non-intrusive shadowing twice, one for computing the shadowing contribution, one for the modified shadowing direction in the unstable contribution. Hence, it is still of interest to analyze the error of non-intrusive shadowing. It is also of interest to partly reduce the systematic error of shadowing methods without involving second-order tangent solvers, which rarely exist for engineering applications. Such a correction is also given in this chapter.

This chapter is organized as follows. First, we show that the shadowing method computes the shadowing contribution of the linear response. Moreover, we bound the remaining part, the unstable contribution, of the linear response. We also explain how to compute part of the unstable contribution by an easy implementation. Finally, we prove the convergence of the non-intrusive formulation to the shadowing contribution.

3.1 Notations

We define some notations that are used only in this chapter. In this chapter, we consider only the case $\mathcal{M} = \mathbb{R}^M$, because we want to impose the Gaussian assumption later. Some sequences are covariant, such as the shadowing direction v, and later v^A; however, some are not covariant, such as v^P, and e^P, e^N, e^{PN}. Also notice that in this section we assume Φ does not depend on γ; if it does, the linear response has an additional term $\rho(\delta\Phi)$, which is easier to deal with than the other parts we consider here.

For now, we assume that shadowing trajectories are representative of the long-time behavior; hence, we can take their difference, the shadowing direction v, to compute the change in the averaged objective.

$$v_k := \sum_{n\geq 0} f_*^n X_{k-n}^- - \sum_{n\leq -1} f_*^n X_{k-n}^+ \,, \tag{3.1}$$

Due to boundedness of the shadowing directions, the limit of summation and the limit in the derivative can now interchange place, so

$$\delta\rho(\Phi) = \delta \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \Phi(x_k) \approx \delta^{sd}\rho(\Phi) := \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \Phi_{xk} v_k \,, \tag{3.2}$$

where $\Phi_{xk} := \partial\Phi/\partial x(x_k)$ is a row vector, and the approximation sign reflects the error introduced by our extra assumption, and upper script '*sd*' is for 'shadowing'.

Equation (2.4) circumvents the issue of exploding gradients, since the first term involves propagating only the stable components into the future, while the second term is subject to the exponential decay of correlation, that is, there is $C_2' > 0$ and $\kappa \in (0,1)$, such that

$$\mathrm{Cor}_n := \left|\rho((\Phi\circ f^n)\,\mathrm{div}_\sigma^+ X^+) - \rho(\Phi)\rho(\mathrm{div}_\sigma^+ X^+)\right| \leq C_2'\kappa^n.$$

Since $\rho(\operatorname{div}_\sigma^+ X^+) = 0$, we have $\mathrm{Cor}_n = \left|\rho((\Phi \circ f^n) \operatorname{div}_\sigma^+ X^+)\right|$. It is very convoluted to express C_2' and κ by properties of the dynamical systems. Even if we could theoretically derive such formulas, they would be too difficult to compute for engineering applications.

In this chapter, we make a statistical assumption about decay of correlation, that is, the decay of the sequence Cor_n starts from the first term. More specifically, we assume that for some C_2 whose magnitude is about 1,

$$\mathrm{Cor}_n = \left|\rho((\Phi \circ f^n) \operatorname{div}_\sigma^+ X^+)\right| \leq C_2 \kappa^n \rho(|\Phi_x X^+|).$$

Here $\rho(|\Phi_x X^+|)$ is a loose bound for Cor_0, since

$$\mathrm{Cor}_0 = \left|\rho(\Phi \operatorname{div}_\sigma^+ X^+)\right| = \left|\rho(\Phi_x X^+)\right| \leq \rho(|\Phi_x X^+|).$$

To reveal the connection between shadowing and the linear response in section 3.2, we further explain how the linear response formula was proved. When changing γ to $\tilde{\gamma}$, f is changed to $\tilde{f} := f(\cdot, \tilde{\gamma})$, and the SRB measure is changed to $\tilde{\rho}$, whose support also moves. Ruelle showed that there is a Holder diffeomorphism, j, so that $\tilde{f} \circ j = j \circ f$. Let $\mu(\cdot) := \tilde{\rho}(j(\cdot))$, then μ has the same support as ρ, and $\rho(\Phi) = \tilde{\rho}(\Phi) = \mu(\Phi \circ j)$. Differentiating $\rho(\Phi)$ by the product rule yields

$$\delta\rho(\Phi) = \rho(\delta(\Phi \circ j)) + \delta\mu(\Phi).$$

Here the term $\rho(\delta(\Phi \circ j))$ accounts for the change of location of the attractor. Via the conjugation map, $\tilde{\rho}$ is pulled back to μ, which is supported on the previous attractor, and the term $\delta\mu(\Phi)$ accounts for its difference from the previous SRB measure, ρ. Ruelle derived expressions for both terms, those are,

$$\begin{aligned}
&\delta^{(1)}\rho(\Phi) := \rho(\delta(\Phi \circ j)) = \sum_{n \geq 0} \rho\left\langle \mathrm{grad}(\Phi \circ f^n), X^- \right\rangle - \sum_{n \leq -1} \rho\left\langle \mathrm{grad}(\Phi \circ f^n), X^+ \right\rangle, \\
&\delta\mu(\Phi) = \delta^{(2)}\rho(\Phi) + \delta^{(3)}\rho(\Phi), \quad \text{where} \\
&\delta^{(2)}\rho(\Phi) := \sum_{n < N} \rho\left\langle \mathrm{grad}(\Phi \circ f^n), X^+ \right\rangle, \; \delta^{(3)}\rho(\Phi) := -\sum_{n \geq N} \rho\left((\Phi \circ f^n) \operatorname{div}_\sigma^+ X^+\right).
\end{aligned} \tag{3.3}$$

Here we further dissect $\delta\mu(\Phi)$ into two parts, and N is a positive integer, whose selection will be addressed later. We call $\delta^{(1)}\rho(\Phi)$ the shadowing contribution, and $\delta\mu(\Phi)$ the unstable contribution of the linear response.

3.2 Approximating linear response by shadowing

In this section, we examine the difference between the linear response formula and the shadowing method. Notice that the non-intrusive formulation does not appear in this section, and our discussion applies to all shadowing methods, including the original least square shadowing. Comparing to previous proofs of shadowing methods [20, 82], which make the extra assumption that shadowing trajectories are representative, here we replace that assumption by an error estimation of its difference with the linear response formula.

Shadowing computes $\rho(\delta(\Phi \circ j))$

In the linear response formula, the term $\rho(\delta(\Phi \circ j))$ is the derivative while assuming μ is fixed, that is, assuming that the SRB measure is preserved by the conjugation map j. Since the SRB measure depicts the long-time behavior, this assumption is very similar to the assumption we made for shadowing methods, hinting the following equivalence.

Lemma 2. *The shadowing contribution of the linear response is accurately exactly the one computed by the shadowing methods. That is,*

$$\delta^{(1)}\rho(\Phi) = \delta^{sd}\rho(\Phi).$$

Here $\delta^{(1)}\rho(\Phi)$ is defined in equation (3.3)*, and $\delta^{sd}\rho(\Phi)$ is defined in equation* (3.2)*.*

Proof. Apply the invariance of SRB measure, we have

$$\delta^{(1)}\rho(\Phi) = \sum_{n\geq 0} \rho\left[\left\langle \mathrm{grad}(\Phi \circ f^n), X^-\right\rangle \circ f^{-n}\right] - \sum_{n\leq -1} \rho\left[\left\langle \mathrm{grad}(\Phi \circ f^n), X^+\right\rangle \circ f^{-n}\right].$$

By the exponential decay, the above formula converges absolutely, hence we can use Fubini's theorem to interchange summation and integration, and

$$\delta^{(1)}\rho(\Phi) = \rho\left[\sum_{n\geq 0} \left\langle \mathrm{grad}(\Phi \circ f^n), X^-\right\rangle \circ f^{-n} - \sum_{n\leq -1} \left\langle \mathrm{grad}(\Phi \circ f^n), X^+\right\rangle \circ f^{-n}\right]$$

Since SRB measure can almost surely be evaluated by long-time averages,

$$\delta^{(1)}\rho(\Phi) = \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \left[\sum_{n\geq 0} \left\langle \mathrm{grad}(\Phi \circ f^n), X^-\right\rangle (x_{k-n}) - \sum_{n\leq -1} \left\langle \mathrm{grad}(\Phi \circ f^n), X^+\right\rangle (x_{k-n})\right]$$

By definition of pushfoward operators,

$$\left\langle \mathrm{grad}(\Phi \circ f^n), X^\pm\right\rangle (x_{k-n}) = \Phi_{xk} f^n_* X^\pm_{k-n}.$$

$$\delta^{(1)}\rho(\Phi) = \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \left[\sum_{n\geq 0} \Phi_{xk} f^n_* X^-_{k-n} - \sum_{n\leq -1} \Phi_{xk} f^n_* X^+_{k-n}\right] = \lim_{K\to\infty} \frac{1}{K} \sum_{k=0}^{K-1} \Phi_{xk} v_k,$$

where the shadowing direction, v_k, is defined in equation (3.1). □

The result of the shadowing method is off from the correct linear response by the error term, $\delta\mu(\Phi)$. A sufficient condition for this term to be zero is that j can be extended to a C^1 map over the entire phase space. For a nice j, absolute continuity to the Lebesgue measure is preserved, and μ is the limit of a measure absolutely continuous to Lebesgue. Since SRB measure is the unique limit of evolving the Lebesgue measure, μ must always be the SRB measure on the original attractor, which yields $\delta\mu \equiv 0$. However, this rarely happens, so instead of hoping the error to disappear, we shall give an estimation of the error term and examine when it can be small.

Error estimation for shadowing

In this subsection, we bound the error term of shadowing methods, $\delta\mu(\Phi)$. By equation (3.3), the error is related to the magnitude of the unstable components of X. Intuitively, if X has no particular reason to be aligned with the unstable directions, projection to a low dimensional unstable subspace reduces the vector norm. Hence, the error should be related to the ratio m/M.

For fixed X and Φ, it is difficult to give an apriori error bound for shadowing methods, because even computing X^+ is already more expensive than non-intrusive shadowing, at which point apriori estimation would stop bringing any benefits. To give an estimation of the shadowing error beforehand, we view Φ, X as random functions. Then we can bound the expectation of the shadowing error under the particular statistical model we choose for Φ and X. From here on in this chapter, we use U to denote the random variable whose value is x, we let U be distributed according to the SRB measure, whose total measure is normalized to 1.

We first define two norms. For a measurable function $g(\Phi, X)$, define

$$\|g\| := (\mathbb{E}(g^2))^{0.5}, \quad \|\rho(g)\| := (\mathbb{E}(\rho(g)^2))^{0.5} = (\mathbb{E}(\mathbb{E}(g|\Phi, X)^2))^{0.5},$$

where the expectation $\mathbb{E}$ is with respect to the joint distribution of Φ, X and x, with x marginally distributed according to the SRB measure ρ; the conditional expectation $\mathbb{E}(\cdot|\Phi, X) = \rho(\cdot)$. By Jensen's inequality, $(\rho(g))^2 \leq \rho(g^2)$. Hence

$$\|\rho(g)\| \leq \|g\|. \tag{3.4}$$

In the remaining part of this subsection, we bound $\|\delta\mu(\Phi)\|$ under two statistical assumptions. First, we assume that X and Φ_x are not particularly aligned with the unstable subspace. Then we bound $\|\Phi_x X^+\|/\|\Phi_x X\|$, where $\|\Phi_x X\|$ is an estimation of the magnitude of the true sensitivity. Then we make an assumption on the rate for exponential decorrelation. Finally, we bound $\|\delta\mu(\Phi)\|$.

Assumption 1. *For any x, $X(x)$ and $\Phi_x(x)$ follow multivariate normal distributions $\mathcal{N}(0, I_M)$. Moreover, for any sequence $\{x_n\}_{n\geq 0}$, the sequence $\{X(x_n)\}_{n\geq 0}$ is independent of $\{\Phi_x(x_n)\}_{n\geq 0}$. Written using conditional probability,*

$$(X(U) \,|\, U = x) \sim \mathcal{N}(0, I_M), \quad (\Phi_x(U) \,|\, U = x) \sim \mathcal{N}(0, I_M), \quad \forall u.$$
$$\{X(U_n)\}_{n\geq 0} \perp\!\!\!\perp \{\Phi_x(U_n)\}_{n\geq 0} \,|\, \{U_n = x_n\}_{n\geq 0}, \quad \forall \{x_n\}_{n\geq 0}$$

Remark. For our purpose, it suffices to assume only for the case where $\{x_n\}_{n\geq 0}$ is a trajectory. An example satisfying this assumption is that both X and Φ_x are constant vector fields on $\mathbb{R}^M$, whose values are drawn from two independent Gaussian.

Lemma 3. *Under assumption 1,*

$$\frac{\|\Phi_x X^+\|}{\|\Phi_x X\|} \leq \frac{1}{\sin\alpha}\sqrt{\frac{m}{M}},$$

where α is the smallest angle between stable and unstable subspace on the attractor.

Remark. This lemma can be generalized in several ways: α can be replaced by some kind of averages instead of the lower bound; assumption 1 can also be replaced by more general model, so long as it somehow describes that X and Φ_x are not intentionally selected to be aligned with the unstable direction. Also, say, if X and Φ_x are models of a large number of random particles, then the assumption 1 may follow from the law of large numbers. We do not attempt to say to much why assumption 1 holds, and certainly more experiments are needed for verifying both assumptions, although current available examples do suggest the assumptions hold in some way, and shadowing works quite well for systems with low unstable ratio.

Proof. By assumption, $X(U)$ and Φ_x have the same distribution for all U, hence

$$\mathbb{E}(\Phi_x X)^2 = \mathbb{E}(\sum_{j=1}^{M} \Phi_x^j X^j)^2 = \mathbb{E}\mathbb{E}[(\sum_{j=1}^{M} \Phi_x^j X^j)^2|U] = \mathbb{E}[(\sum_{j=1}^{M} \Phi_x^j X^j)^2|U].$$

By independence, $\mathbb{E}[\Phi_x^i X^j \Phi_x^k X^l|U] = 0$ unless $i = k$ and $j = l$, where X^j is the j-th coordinate of X. Hence,

$$\mathbb{E}(\Phi_x X)^2 = \sum_{j=1}^{M} \mathbb{E}[(\Phi_x^j X^j)^2|U] = M \quad \Rightarrow \quad \|\Phi_x X\| = \sqrt{M}. \tag{3.5}$$

Denote the entries in the oblique projection matrix P^+ by P_{ij}^+, then

$$\begin{aligned}\mathbb{E}(\Phi_x X^+)^2 = \mathbb{E}(\Phi_x P^+ X)^2 = \mathbb{E}(\sum_{i,j} \Phi_x^i P_{ij}^+ X^j)^2 = \mathbb{E}\mathbb{E}[(\sum_{i,j} \Phi_x^i P_{ij}^+ X^j)^2|U] \\ = \mathbb{E}\sum_{i,j} \mathbb{E}[(\Phi_x^i P_{ij}^+ X^j)^2|U] = \rho\left(\sum_{i,j}(P_{ij}^+)^2\right).\end{aligned}$$

The orthogonal invariance of Frobenius norm says that, for any $M \times M$ orthogonal matrix A,

$$\sum_{i,j}(P_{ij}^+)^2 = tr(P^{+T}P^+) = tr((P^+A)^T(P^+A)) = \sum_{i,j}(P^+A)_{ij}^2.$$

Let the first m and the rest $M - m$ columns of A be orthonormal basis of $(V^-)^\perp$ and V^-, then only the first m columns of P^+A are non-zero, and their norms are bounded above by $1/\sin\alpha$. Hence,

$$\mathbb{E}(\Phi_x X^+)^2 = \rho\left(\sum_{i,j}(P^+A)_{ij}^2\right) \le \rho\left(\frac{m}{(\sin\alpha)^2}\right) = \frac{m}{(\sin\alpha)^2}.$$

The lemma is proved by dividing by equation (3.5). □

Assumption 2. *For the entire distribution of* Φ *and* X*, there are uniform constants* $C_2 > 0, 0 < \kappa < 1$*, such that*

$$\mathrm{Cor}_n := \left|\rho((\Phi \circ f^n)\,\mathrm{div}_\sigma^+ X^+)\right| \leq C_2\kappa^n \rho(|\Phi_x X^+|).$$

Remark. A typical trick to break this uniformity assumption is to pass Φ to $\Phi \circ f^n$; however, this trick does not affect $\delta\mu(\Phi)$, which is what we are really interested in. Moreover, this assumption is backed by observations in such as [15]. It is also worth noticing that the decorrelation rate is faster than κ in the short time [21], making the bound safer.

Theorem 2 (error of shadowing)**.** *Under assumption 1 and 2,*

$$\frac{\|\delta\mu(\Phi)\|}{\|\Phi_x X\|} \leq \left(\frac{C_1}{(1-\lambda)\sin\alpha} + \frac{C_2\kappa}{(1-\kappa)\sin\alpha}\right)\sqrt{\frac{m}{M}}.$$

Remark. (1) Our estimation here also bounds the error of S3 [16] and blended response algorithm [1]. Both S3 and blended response introduce approximations on the unstable contribution, hence their errors should be somewhat smaller than shadowing, although it is difficult to quantify those errors more accurately without extra assumptions. (2) To generalize this lemma, we may replace the lower bounds on decay rate, κ and λ, by some form of averages. Slow decorrelation or decay not only affect shadowing methods; they make most theories and computations related to SRB measures difficult. (3) For a given application, posteriori error of shadowing can be obtained by comparing with finite differences.

Proof. Set $N = 1$ in equation (3.3). First notice that the exponential decay of terms in $\delta^{(2)}\rho(\Phi)$ is given by propagating unstable vectors forward in time. Note that $\Phi_x(f^n(u))$ and $X(u)$ are independent by assumption 1, we have

$$\left\|\left\langle \mathrm{grad}(\Phi \circ f^n), X^+\right\rangle\right\|^2 = \left\|\Phi_x f_*^n P^+ X\right\|^2 = \rho\left(\sum_{i,j}(f_*^n P^+)_{ij}^2\right).$$

Use the same A as in the proof of lemma 3, then use the fact that the non-zero columns in P^+A are in the unstable subspace, and f_*^n reduces their norms for $n \leq 0$,

$$\rho\left(\sum_{i,j}(f_*^n P^+)_{ij}^2\right) = \rho\left(\sum_{i,j}(f_*^n P^+ A)_{ij}^2\right) \leq C_1^2\lambda^{-2n}\frac{m}{(\sin\alpha)^2}.$$

Hence, by equation (3.4),

$$\left\|\rho\left\langle \mathrm{grad}(\Phi \circ f^n), X^+\right\rangle\right\| \leq \left\|\left\langle \mathrm{grad}(\Phi \circ f^n), X^+\right\rangle\right\| \leq C_1\lambda^{-n}\sqrt{m}/\sin\alpha.$$

On the other hand, the exponential decay of terms in $\delta^{(3)}\rho(\Phi)$ is due to the decorrelation, with the rate given by assumption 2.

$$\left\|\rho\left((\Phi \circ f^n)\,\mathrm{div}_\sigma^+ X^+\right)\right\| \leq C_2\kappa^n\|\rho(|\Phi_x X^+|)\| \leq C_2\kappa^n\|\Phi_x X^+\|.$$

Further use the estimation of $\|\Phi_x X^+\|$ in lemma 3, we have

$$\left\|\rho\left((\Phi\circ f^n)\operatorname{div}_\sigma^+ X^+\right)\right\| \le C_2\kappa^n\sqrt{m}/\sin\alpha.$$

Finally, the error of shadowing methods is bounded by sums of two geometric series.

$$\left\|\delta^{(2)}\rho(\Phi)\right\| \le \sum_{n\le 0}\left\|\rho\left\langle \operatorname{grad}(\Phi\circ f^n), X^+\right\rangle\right\| \le \frac{C_1\sqrt{m}}{(1-\lambda)\sin\alpha};$$
$$\left\|\delta^{(3)}\rho(\Phi)\right\| \le \sum_{n\ge 1}\left\|\rho\left((\Phi\circ f^n)\operatorname{div}_\sigma^+ X^+\right)\right\| \le \frac{C_2\kappa\sqrt{m}}{(1-\kappa)\sin\alpha}.$$

The proof is completed by the definition $\delta\mu(\Phi) := \delta^{(2)}\rho(\Phi) + \delta^{(2)}\rho(\Phi)$. □

By our estimation, an interesting scenario where shadowing methods have small error is when the unstable ratio $m/M \ll 1$. This is typically the case for systems with dissipation, such as fluid mechanics, where non-intrusive shadowing is successful [57, 58, 59, 7, 18]. In fact, SRB measure was invented for dissipative systems, many of which have low dimensional unstable subspaces. However, there are counter examples with large unstable ratio, and shadowing methods fail. A remedy to reduce the systematic error is given in the next section.

Corrections to shadowing methods

When the error of shadowing method is large, it can be reduced by further adding $\delta^{(2)}\rho(\Phi)$ defined in equation (3.3). This correction reduces, though not eliminate, the systematic error of shadowing. By proof of theorem 2, the relative error is reduced to

$$\frac{\left\|\delta^{(3)}\rho(\Phi)\right\|}{\|\Phi_x X\|} \le \frac{C_2\kappa^N}{(1-\kappa)\sin\alpha}\sqrt{\frac{m}{M}}. \tag{3.6}$$

Increasing N exhausts the unstable contribution, however, the computational cost would grow exponentially for large N. In fact, earlier work on shadowing methods suggested that relaxing the constraint in the optimization could improved the accuracy [9]; by our current analysis, we now know that is because relaxing constraint may allow some unstable contributions.

We illustrate the correction term on the 1-dimensional sawtooth map, or the expanding circle, which was previously used as a counter example of shadowing methods [8]. It is also the underlying source of chaos for several other counter examples such as the solenoid map. Now we know that shadowing methods fail because the only dimension is unstable. However, the proposed correction fixes the error with a small N.

Example (expanding circle)**.** Consider the dynamical system on $[0, 2\pi)$ given by

$$x_{k+1} = f(x_k, s) := 2x_k + s\sin x_k \pmod{2\pi}, \quad \Phi(x) := \cos x.$$

The base parameter is $s = 0$, at which we compute the derivative. Although this map is 2-to-1 rather than a diffeomorphism, the linear response formula is still correct [4]. [1]

The SRB measures, ρ, of a 2-to-1 map is still defined as the long-time limit of evolving the Lebesgue measure. However, $f^n(\cdot)$ is no longer a function for $n < 0$, for example, $f^{-1}x$ can be either $x/2$ or $x/2 + \pi$. For a random variable U distributed according to ρ, $\{U_n := f^n(U)\}_{n\le 0}$ is a Markov chain, with U_{n-1} equally distributed given U_n. More specifically, for $n \le 0$, the conditioned probability

$$\mathbb{P}\left(U_{n-1} = \frac{1}{2}U_n \mid U_n\right) = \mathbb{P}\left(U_{n-1} = \frac{1}{2}U_n + \pi \mid U_n\right) = \frac{1}{2}.$$

Since there is no stable subspace,

$$X^+(U) = X(U) = \sin(U_{-1}).$$

By the chain rule,

$$\operatorname{grad}(\Phi \circ f^n)(U) = -2^n \sin(U_n).$$

Hence,

$$\left\langle \operatorname{grad}(\Phi \circ f^n), X^+ \right\rangle = -2^n \sin(U_n)\sin(U_{-1}).$$

To show that shadowing with correction gives the true derivative for any $N \ge 0$, we only need to check that each term in $\delta^{(3)}\rho(\Phi)$ is zero. For $n \ge 0$, $U_n = 2^n U$ is a well-defined function, and the n-th term in $\delta^{(3)}\rho(\Phi)$ is

$$\begin{aligned} &-\rho\left\langle (\Phi \circ f^n)\operatorname{div}_\sigma^+ X^+ \right\rangle = \rho\left\langle \operatorname{grad}(\Phi \circ f^n), X^+ \right\rangle \\ =& -\mathbb{E}(2^n \sin(2^n U)\sin U_{-1}) = -\mathbb{E}(2^n \sin(2^n U)\mathbb{E}(\sin U_{-1} \mid U)) = 0. \end{aligned}$$

We also directly compute $\delta^{(2)}\rho(\Phi)$. For $n \le -2$,

$$\rho\left\langle \operatorname{grad}(\Phi \circ f^n), X^+ \right\rangle = -\mathbb{E}(2^n \sin U_n \sin U_{-1}) = -\mathbb{E}(2^n \sin U_{-1}\mathbb{E}(\sin U_n \mid U_{-1})) = 0$$

The only non-zero term is $n = -1$,

$$\rho\left\langle \operatorname{grad}(\Phi \circ f^{-1}), X^+ \right\rangle = \frac{1}{2}\rho\left(-\frac{1}{2}\sin^2\frac{x}{2}\right) + \frac{1}{2}\rho\left(-\frac{1}{2}\sin^2\frac{x+2\pi}{2}\right) = -\frac{1}{4}.$$

By the same computations given above, we can see that the shadowing contribution is $\delta^{sd}\rho(\Phi) = 1/4$. This is the same as the computational result in figure 2-17(a) of Blonigan's thesis [8], where the interval was shrunk to $[0, 1]$. □

[1] We thank Adam A. Sliwiak for pointing out that the simple symmetry argument for zero derivative, in the previous draft, is wrong. We do need to cite Baladi even only for the case $s = 0$.

When $M > 1$, X^+ can be efficiently computed by a 'little-intrusive' formulation, which requires both tangent and adjoint solvers. Denote the adjoint unstable subspace by $\overline{V}^+$, then $\dim \overline{V}^+ = \dim V^+$, and $\overline{V}^+ \perp V^-$ [54]. Moreover, both the unstable tangent and adjoint subspaces can be obtained by evolving homogeneous tangent and adjoint equations [54]. To find X^+, just solve the vector such that

$$X^+ \in V^+, \quad \left\langle X - X^+, \overline{V}^+ \right\rangle = 0.$$

With $\{w_i\}_{i=1}^m$ as the basis of V^+, we can write X^+ as $X^+ = \sum_{i=1}^m c^i w^i$, then there are exactly m linear equations for m undetermined coefficients, $\{c_i\}_{i=1}^m$. The blended response algorithm also requires computing X^+, which was done with cost $O(M)$ [1]; in contrast, the little-intrusive formulation requires only $O(m)$, hence it can help improving efficiency of the blended response algorithm.

3.3 Convergence of non-intrusive shadowing

In this section we prove the convergence of the non-intrusive shadowing algorithm given in equation (2.6), to the shadowing contribution $\delta^{(1)}$ given in equation (3.3). Together with the error analysis of the shadowing contribution in section 3.2, we have the error of computing linear response using non-intrusive shadowing algorithms.

In this section, we assume that in the non-intrusive shadowing algorithm in equation (2.6),

$$\operatorname{span}(w_1, \cdots, w_x) = V^+.$$

This assumption can be achieved by evolving w_i's for some time before the zeroth step, since the unstable components in w_i's grow faster than stable components. In reality, such pre-process is typically not needed for non-intrusive shadowing to converge, but making this assumption simplifies our theoretical analysis. Should we want to extend our analysis to cases without this pre-process, we need a sharp estimation of the unstable components in the random initial conditions of w_i's.

We start with some definitions. Denote the total number of steps by K. In this section, v is the shadowing direction given in equation (3.1). In the non-intrusive shadowing algorithm, let v' be

$$v'_k := \sum_{0 \le n \le k-1} f_*^n X_{k-n} \,.$$

We will show v' is the inhomogeneous tangent solution solved from zero initial condition. Moreover, let v^P be the pivot solution defined by

$$v_k^P := \sum_{0 \le n \le k-1} f_*^n X_{k-n}^- - \sum_{n \le -1} f_*^n X_{k-n}^+ \,.$$

We will show it is in the feasible set of the non-intrusive shadowing problem, and also close to v. Denote the solution of the non-intrusive shadowing algorithm by v^N. Define v^A, which bounds both v and v^P, by

$$v_k^A := \sum_{0\le n} |f_*^n X_{k-n}^-| + \sum_{n\le -1} |f_*^n X_{k-n}^+|\,, \tag{3.7}$$

where $|\cdot|$ is the vector norm. v^A and v are covariant, that is,

$$v_k^A = v_0^A \circ f^k.$$

However, notice that v^P is not covariant: that is why we will mostly bound it by v^A. Moreover, we define the errors

$$e^N := v^N - v\,, \quad e^P := v^P - v\,, \quad e^{PN} := v^P - v^N\,.$$

Finally, the error of computing the shadowing contribution using non-intrusive shadowing is

$$\tilde{e}^N := \frac{1}{K}\sum_{k=0}^{K-1} \left\langle e_k^N, \Phi_{xk} \right\rangle.$$

In the remaining part of this section, we show that $e^N = e^{PN} + e^P$ converges to zero, by showing the convergence of e^{PN} and e^P. We will bound e_{K-1}^{PN} and e_0^P by v^A. Then, due to the exponential decay of $e^{PN} \in V^+$ and $e^P \in V^-$, the averaged error, $\tilde{e}^N$, goes to zero as $K \to \infty$; moreover, we give a quantitative bound on $\tilde{e}^N$ under assumption 1. We start by verifying some basic properties of the terms we just defined.

Lemma 4. *v, v', and v^P are inhomogeneous tangent solutions satisfying equation (2.5); $v_0' = 0$; v^P is in the feasible set of non-intrusive shadowing, that is, $v^P - v' \in V^+$. e^N, e^P, and e^{NP} are homogeneous tangent solutions.*

Proof. To see v^P is inhomogeneous tangent, apply definitions,

$$\begin{aligned} v_{k+1}^P - f_* v_k^P &= \sum_{0\le n\le k} f_*^n X_{k+1-n}^- - \sum_{n\le -1} f_*^n X_{k+1-n}^+ - \sum_{0\le n\le k-1} f_*^{n+1} X_{k-n}^- + \sum_{n\le -1} f_*^{n+1} X_{k-n}^+ \\ &= \sum_{0\le n\le k} f_*^n X_{k+1-n}^- - \sum_{n\le -1} f_*^n X_{k+1-n}^+ - \sum_{1\le l\le k} f_*^l X_{k+1-l}^- + \sum_{l\le 0} f_*^l X_{k+1-l}^+ \\ &= X_{k+1}^- + X_{k+1}^+ = X_{k+1}. \end{aligned}$$

Similarly we can verify that v, defined by equation (3.1), and v', are inhomogeneous tangent. Also, by definitions, $v_0' = 0$, and

$$v_k^P - v_k' = -\sum_{n\le -1} f_*^n X_{k-n}^+ \in V_k^+.$$

Finally, e^N, e^P, and e^{NP} are homogeneous tangent solutions, since they are differences between inhomogeneous tangent solutions. □

Lemma 5. *The peak values of $e^{PN} \in V^+$ and $e^P \in V^-$ are bounded by*

$$|e^{PN}_{K-1}| \le \sum_{k=0}^{K-1} \lambda^{K-1-k} v^A_k \,, \quad |e^P_0| < v^A_0.$$

Remark. The main tool for bounding $e^{PN} \in V^+$ is that, the unstable homogeneous tangent has a spike at $K-1$, hence e^{PN} can not to be too large without increasing $\|v^N\|_{l^2}$. Hence minimizing $\|v^N\|_{l^2}$ controls e^{PN}. The large spike is encoded by the fact that $\|e^{PN}\|_{l^2} \approx |e^{PN}_{K-1}|$.

Proof. Since $\|v^N\|$ is minimized in the non-intrusive shadowing problem, $\|v^N + \alpha w\|$ is minimal at $\alpha = 0$, for any $w \in V^+$. By computing derivative with respect to α, we have the so-called first-order optimality condition,

$$\left\langle v^N, w \right\rangle_K := \sum_{k=0}^{K-1} \left\langle v^N_k, w_k \right\rangle = 0, \quad \text{for all } w \in V^+. \tag{3.8}$$

Notice that $v^P - v' \in V^+$ and $v^N - v' \in V^+$ by definitions, hence

$$e^{PN} := v^P - v^N \in V^+.$$

Substitute $w = e^{PN}$ and $v^N = v^P - e^{PN}$ into equation (3.8), we have specifically

$$\left\langle v^P - e^{PN}, e^{PN} \right\rangle_K = 0 \quad \Rightarrow \quad \left\langle e^{PN}, e^{PN} \right\rangle_K = \left\langle e^{PN}, v^P \right\rangle_K .$$

The peak value of e^{PN} is at step $K-1$, which is smaller than its l^2 norm, hence

$$|e^{PN}_{K-1}|^2 \le \left\langle e^{PN}, e^{PN} \right\rangle_K = \left\langle e^{PN}, v^P \right\rangle_K .$$

Apply the Cautchy-Schwarz and exponential decay of e^{PN}, we have

$$|e^{PN}_{K-1}|^2 \le \left\langle e^{PN}, v^P \right\rangle_K \le \sum_{k=0}^{K-1} |e^{PN}_k||v^P_k| \le \sum_{k=0}^{K-1} \lambda^{K-1-k} |e^{PN}_{K-1}||v^P_k|.$$

Cancel $|e^{PN}_{K-1}|$ from both sides, we get

$$|e^{PN}_{K-1}| \le \sum_{k=0}^{K-1} \lambda^{K-1-k} |v^P_k| \le \sum_{k=0}^{K-1} \lambda^{K-1-k} v^A_k.$$

To prove the second inequality in the lemma, notice that by definition,

$$e^P_k = \sum_{n \ge k} f^n_* X^-_{k-n} \in V^-_k \,,$$

and the inequality is obtained by the definition of v^A. □

Theorem 3 (convergence of non-intrusive shadowing)**.** *Under assumption 1,*

$$\frac{\|\tilde{e}^N\|}{\|\Phi_x X\|} \leq \frac{1}{K\|X\|}\sum_{k=0}^{K-1}\|e_k^N\| \leq \frac{4}{K(1-\lambda)^3\sin\alpha}.$$

Here K is the length of the trajectory.

Remark. (1) Previous shadowing methods, such as the least squares shadowing, has the same $O(K^{-1})$ convergence speed [82]. Hence, the non-intrusive shadowing reduces the computation with no additional error. Also note that the convergence to linear response in previous shadowing literature was wrong, it should be convergence to the shadowing contribution. (2) The bound on e^N is useful when the non-intrusive shadowing is used for computing only the shadowing direction but not the shadowing contribution, for example, when computing the modified shadowing direction in the linear response algorithm [56].

Proof. By definition,

$$\|\tilde{e}^N\| \leq \frac{1}{K}\sum_{k=0}^{K-1}\|\left\langle e_k^N, \Phi_{xk}\right\rangle\| = \frac{1}{K}\sum_{k=0}^{K-1}\left[\mathbb{E}\mathbb{E}\left(\left\langle e_k^N, \Phi_{xk}\right\rangle^2 | x_0, X\right)\right]^{0.5}.$$

Here e^N is determined given x_0 and X. We choose a coordinate whose first axis is parallel to e_k^N, then Φ_{xk} is still multi-variate Gaussian in this new coordinate. In particular, its first coordinate, $\Phi_{xk}^1 \sim \mathcal{N}(0,1)$, whereas other coordinate components are orthogonal to e_k^N. Hence,

$$\mathbb{E}\left(\left\langle e_k^N, \Phi_{xk}\right\rangle^2 | x_0, X\right) = \mathbb{E}\left(\left\langle e_k^N, \Phi_{xk}^1\right\rangle^2 | x_0, X\right) = |e_k^N|^2\mathbb{E}(\Phi_{xk}^1)^2 = |e_k^N|^2.$$

By substitution,

$$\|\tilde{e}^N\| \leq \frac{1}{K}\sum_{k=0}^{K-1}\|e_k^N\|.$$

Since $e^N = e^{PN} + e^P$, where $e^{PN} \in V^+$, $e^P \in V^-$,

$$\frac{1}{K}\sum_{k=0}^{K-1}\|e_k^N\| \leq \frac{1}{K}\sum_{k=0}^{K-1}\|e_k^{PN}\| + \|e_k^P\| \leq \frac{1}{K}\sum_{k=0}^{K-1}\lambda^{K-1-k}\|e_{K-1}^{PN}\| + \lambda^k\|e_0^P\| \leq \frac{\|e_{K-1}^{PN}\| + \|e_0^P\|}{K(1-\lambda)}.$$

By lemma 5, also notice that $\rho(v_k^A) = \rho(v_0^a)$ since v^A is covariant, we have

$$\|e_{K-1}^{PN}\| \leq \sum_{k=0}^{K-1}\lambda^{K-1-k}\|v_k^A\| \leq \frac{\|v_0^A\|}{1-\lambda}, \quad \|e_0^P\| < \|v_0^A\|.$$

To estimate v_0^A, use its definition in equation (3.7),

$$\|v_0^A\| \leq \sum_{0\leq n}\|f_*^n X_{-n}^-\| + \sum_{n\leq -1}\|f_*^n X_{-n}^+\| \leq \sum_{0\leq n}\lambda^n\|X_{-n}^-\| + \sum_{n\leq -1}\lambda^{-n}\|X_{-n}^+\|,$$

Since $X^-_{-n}(\cdot) := X^- \circ f^{-n}(\cdot)$, X_n is covariant, hence $\|X^-_{-n}\| = \|X^-\|$, and

$$\|v_0^A\| \leq \frac{\|X^-\| + \|X^+\|}{1-\lambda} \leq \frac{2\|X\|}{(1-\lambda)\sin\alpha}$$

Under assumption 1, $\|X\| = \sqrt{M}$, hence

$$\|\tilde{e}^N\| \leq \frac{1}{K}\sum_{k=0}^{K-1}\|e_k^N\| \leq \frac{2\|v_0^A\|}{K(1-\lambda)^2} \leq \frac{4\|X\|}{K(1-\lambda)^3\sin\alpha} = \frac{4\sqrt{M}}{K(1-\lambda)^3\sin\alpha}$$

By equation (3.5) in lemma 3, $\|\Phi_x X\| = \|X\| = \sqrt{M}$, hence this lemma theorem is proved. □

Theorem 4. *The error of approximating linear response by non-intrusive shadowing, under assumption 1 and 2, is bounded by the sum of the bounds in theorem 2 and 3.*

3.4 Conclusions

This chapter estimates bound the error in approximating linear response by the non-intrusive shadowing algorithm. First, we estimate bound the error of approximating linear response by the shadowing contribution, then we prove the convergence of the non-intrusive algorithm to the shadowing contribution. For engineering applications, especially dissipative systems with large degrees of freedom such as computational fluids, we suggest to first try non-intrusive shadowing, then add on the little-intrusive correction if error is large. For many previous applications, non-intrusive shadowing can be quite accurate even without correction. A full-blown realization of Ruelle's formula requires further computing the unstable contribution, which is achieved by the fast linear response algorithm in chapter 6, which has no systematic error, and is not much more complicated than shadowing.

Chapter 4

Finite-difference non-intrusive shadowing

We present the finite-difference version of the non-intrusive shadowing algorithm for computing sensitivities of long-time averaged quantities in chaotic dynamical systems. It is also known as the finite-difference shadowing, or the Finite-difference non-intrusive least-squares shadowing (FD-NILSS) algorithm. Finite-difference shadowing does not require tangent solvers, and can be implemented with little modification to existing numerical simulation software. This enriches applications of finite-difference shadowing to engineering problems, since most numerical simulation software do not have accompanying tangent solvers. We also give a formula for solving the least-squares problem in finite-difference shadowing, which can be applied in non-intrusive shadowing as well. Finally, we apply finite-difference shadowing for sensitivity analysis of a chaotic flow over a 3-D cylinder at Reynolds number 525, where finite-difference shadowing computes pretty accurate sensitivities and the computational cost is in the same order as the numerical simulation.

4.1 Deriving finite-difference shadowing

The finite-difference shadowing seeks to implement non-intrusive shadowing with only primal numerical solvers, which solves the primal equation in equation (2.7). The main difficulty is primal solvers typically do not provide partial derivatives. To resolve this difficulty, finite-difference shadowing approximately computes these unprovided quantities through finite-differences.

Tangent solvers compute tangent solutions via solving the tangent equations, whereas in finite-difference shadowing we compute tangent solutions via the their finite-difference approximations. More specifically, on a trajectory $x(t), t \in [0, T]$, with initial condition x^0, to approximate a homogeneous solution w_j with initial condition w_j^0, we compute primal solution x_j^w by keeping the same γ but using initial conditions $x^0 + \Delta\varphi_j w_j^0$, where $\Delta\varphi_j$ is a

small number. The approximation for w_j is thus

$$w_j \approx \frac{x_j^w - x}{\Delta\varphi_j} \,. \tag{4.1}$$

Let W be the matrix whose columns are $\{w_j\}_{j=1}^u$, now with each column approximated via finite-difference, we also obtain an approximation of W.

Similarly, to approximate an inhomogeneous tangent solution v' with initial condition v'^0, we compute primal solution x' with parameter $\gamma + \Delta\gamma$ and initial condition $x^0 + \Delta\gamma v'^0$. The approximation for v' is thus

$$v' \approx \frac{x' - x}{\Delta\gamma} \,. \tag{4.2}$$

These approximations allows us to compute tangent solutions from primal solvers. With those tangent solutions, we can compute a via solving the minimization in equation (2.17), and compute the shadowing direction by $v = v' + Wa$.

We explain how to compute ξ evaluated at $t = 0, T$, which appear in equation (2.13). At any t, the map $\psi : \mathbb{R}^m \to \mathbb{R}$ which maps $v(t)$ to $\xi(t)$ is a linear map defined as:

$$\psi(p) = \frac{p^T f}{f^T f} \,, \tag{4.3}$$

where $p \in \mathbb{R}^m$, and f is evaluated at t. Since we are expressing v as $v = v' + Wa$, we can compute ξ from the same linear combination:

$$\xi = \psi(v) = \psi(v' + Wa) = \psi(v') + [\psi(w_1), \ldots, \psi(w_u)]a \,, \tag{4.4}$$

where v' and $\{w_j\}_{j=1}^u$ are computed via finite-difference. This way of computing ξ saves computer memory, since we no longer need to store vectors v and $\{w_j\}_{j=1}^u$ at $t = 0$ and $t = T$; instead, we only need to store scalars $\psi(v'), \psi(w_1), \ldots, \psi(w_u)$ evaluated at $t = 0$ and $t = T$.

Finally we explain how to approximate, via finite-differences, terms in equation (2.13) involving $\partial_x\Phi$ and $\delta\Phi$, which are typically not provided in numerical primal solvers. More specifically,

$$\begin{aligned}
&\int_0^T (\partial_x\Phi\, v + \delta\Phi)\, dt \\
=&\int_0^T [\partial_x\Phi\, (v' + Wa) + \delta\Phi]\, dt \\
=&\int_0^T (\partial_x\Phi\, v' + \delta\Phi)\, dt + \sum_{j=1}^u a_j \int_0^T (\partial_x\Phi\, w_j)\, dt \\
\approx&\int_0^T \frac{1}{\Delta\gamma} (\Phi(\gamma + \Delta\gamma) - \Phi(\gamma))\, dt + \sum_{j=1}^u a_j \int_0^T \frac{1}{\Delta\varphi_j} (\Phi(\varphi_j + \Delta\varphi_j) - \Phi(\varphi_j))\, dt \\
=&\tilde{\Phi}' + \sum_{j=1}^u a_j \tilde{\Phi}_j^w \,.
\end{aligned} \tag{4.5}$$

Here $\tilde{\Phi}'$ and $\tilde{\Phi}_j^w$ are defined as:

$$\begin{aligned} \tilde{\Phi}' &= \frac{1}{\Delta\gamma}\int_0^T \left(\Phi(\gamma+\Delta\gamma)-\Phi(\gamma)\right)dt \\ \tilde{\Phi}_j^w &= \frac{1}{\Delta\varphi}\int_0^T \left(\Phi(\varphi_j+\Delta\varphi_j)-\Phi(\varphi_j)\right)dt\ , \end{aligned} \tag{4.6}$$

where $\Phi(\gamma+\Delta\gamma)$ is short for $\Phi(x(\gamma+\Delta\gamma,\varphi_1,\ldots,\varphi_u,t),\gamma+\Delta\gamma)$, that is, the instantaneous objective evaluated from using parameter $\gamma+\Delta\gamma$, while all φ_j's are fixed as base values. Similarly, $\Phi(\varphi_j+\Delta\varphi_j)$ is short for $\Phi(x(\gamma,\varphi_1,\ldots,\varphi_j+\Delta\varphi_j,\ldots,\varphi_u,t),\gamma)$.

4.2 Finite-difference shadowing algorithm

Dividing trajectory into segments

There are two numerical issues when applying finite-difference shadowing on a whole trajectory with a large time length T. The first issue, similar to non-intrusive shadowing, is that tangent solutions become dominated by the fasted growing CLV, as a result, the minimization problem in equation (2.17) becomes ill-conditioned. The second issue, unique to finite-difference shadowing, is that the perturbation on the trajectory falls out of the linear region, thus finite-differences no longer approximate tangent solutions. For finite-difference shadowing we use a similar technique as non-intrusive shadowing to solve these issues, that is, dividing the whole trajectory into multiple segments, and renormalize at interfaces.

Our solution is analogous to the multiple shooting method for boundary value problem [53], and the multiple shooting shadowing algorithm [9]. However, the main difference here is that we are trying to find a continuous tangent solution, v, within a u-dimensional affine subspace. This requires more work than just matching v at end points of segments: we must first of all have a connected affine subspace. This is achieved by the renormalization procedure we introduce at the end of each segment.

We first describe how we select the subscript representing segment number for different quantities, as shown in figure 4.1. T is the time length of the entire trajectory, which is further divided into A segments, each of length ΔT. The i-th segment spans $[t_i, t_{i+1}]$, where $t_0 = 0, t_A = T$. For quantities defined on a entire segment such as W_i, v_i', C_i, d_i and a_i, their subscripts are the same as the segment they are defined on. For quantities defined only at the interfaces between segments such as Q_i, R_i, b_i and λ_i, their subscripts are the same as the time point they are defined at.

We use $W_{i-1}(t) = [w_{i-1,1}(t), \cdots w_{i-1,u}(t)]$, a $M \times u$ matrix-valued function of time, to denote homogeneous tangent solutions on the $(i-1)$-th segment. Assume that we have computed W_{i-1}, we explain how to generate initial conditions for W_i. At time t_i, we first project all homogeneous tangent solutions to the subspace perpendicular to $f(t_i)$, to get $W_{i-1}^{\perp}(t_i)$, upon which we then perform QR factorization, and use Q, the matrix with

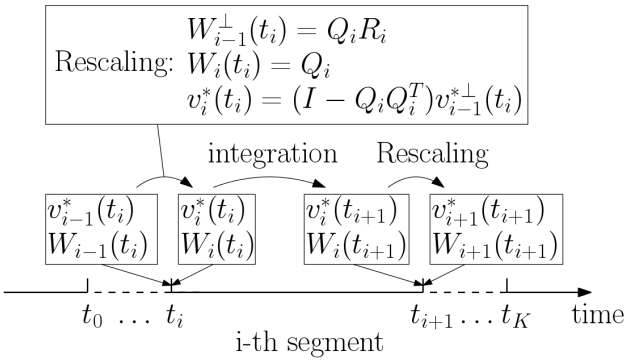


Figure 4.1: Subscripts used in this chapter, where $t_0 = 0$, $t_A = T$. $W_i(t), v_i'(t)$ are defined on the i-th segment, which spans $t \in [t_i, t_{i+1}]$; Q_i, R_i are defined at t_i. Integrating tangent equations happens within one segment. Rescaling tangent solutions happens at the interface between two segments.

orthonormal columns, as the initial condition for W_i on segment i. More specifically,

$$W_{i-1}^{\perp}(t_i) = Q_i R_i \,, \quad \text{and} \quad W_i(t_i) = Q_i \,. \tag{4.7}$$

We use $v_{i-1}'(t)$, a M-dimensional vector-valued function of time, to denote the particular inhomogeneous tangent solution on the $(i-1)$-th segment. Assume that we have computed v_{i-1}', we explain how to generate initial conditions for v_i'. At time t_i, we first project to get $v_{i-1}'^{\perp}(t_i)$, from which we then subtract its orthogonal projection onto homogeneous tangent solutions. More specifically,

$$b_i = Q_i^T v_{i-1}'^{\perp}(t_i) \,, \quad \text{and} \quad v_i'(t_i) = v_{i-1}'^{\perp}(t_i) - Q_i b_i \,. \tag{4.8}$$

This rescaling maintains the continuity of the affine space $v_i'^{\perp} + span\{w_{ij}^{\perp}\}_{j=1}^{u}$ across different segments.

The continuity of affine space allows us to impose continuity condition for $v_i^{\perp}$, which is the approximate shadowing direction on the i-th segment. On each segment, define $v_i = v_i' + W_i a_i$, where $a_i \in \mathbb{R}^u$. The continuity condition can now be expressed via a relation between a_i and a_{i-1}:

$$v_i'^{\perp}(t_i) + W_i^{\perp}(t_i) a_i = v_{i-1}'^{\perp}(t_i) + W_{i-1}^{\perp}(t_i) a_{i-1} \,. \tag{4.9}$$

Apply equation (4.7) and (4.8), and notice that $v_i'(t_i) = v_i'^{\perp}(t_i)$, we get:

$$-Q_i b_i + Q_i a_i = Q_i R_i a_{i-1} \tag{4.10}$$

Since Q_i has orthonormal columns, $Q_i^T Q_i = I \in \mathbb{R}^{u \times u}$. Multiplying Q_i^T to the left of both sides, we have the continuity condition for v at t_i:

$$a_i = R_i a_{i-1} + b_i \,. \tag{4.11}$$

Due to the continuity condition, we can see that $v_i^\perp$ is continuous across segment. However, v_i is not continuous. So there remains the question of how to construct a continuous v and ξ from v_i and ξ_i, where ξ_i is defined as such that $\xi_i(t)f(t) = v_i(t) - v_i^\perp(t)$, so that we can apply equation (2.13) to compute derivatives. We give the following formula on the i-th segment:

$$\begin{aligned} \xi(t) &= \xi_i(t) + \sum_{i'=0}^{i-1} \xi_{i'}(t_{i'+1}) \,, \\ v(t) &= v_i^\perp(t) + \xi(t) f(t) \,. \end{aligned} \tag{4.12}$$

Readers may directly verify that above formula give continuous v and ξ; moreover, $v^\perp = v_i^\perp$ on all segments, v is an inhomogeneous tangent solution, and ξ and v satisfy the pairing condition $\xi(t)f(t) = v(t) - v^\perp(t)$.

A further remark is that, we choose here to keep the continuity of the projected affine space $v_i'^\perp + span\{w_{ij}^\perp\}_{j=1}^u$. We think it is also possible to choose to keep the continuity of the unprojected space $v_i' + span\{w_{ij}\}_{j=1}^u$, which should lead to easier derivation and programming. However, we can not get rid of the projection process completely, since it still exists in the minimization step. Moreover, this alternative approach would require to compute one more homogeneous tangent solution, since the neutral CLV is no longer projected out. We suggest interested readers to try this possible approach.

Procedure list of the finite-difference shadowing algorithm

We should first prescribe the following parameters for finite-difference shadowing:

- Number of homogeneous tangents u. Here we refer readers to [29] for how numerical discretization affects u.
- Perturbations $\Delta\gamma, \Delta\varphi_1, \ldots, \Delta\varphi_u$. Typically we set $\Delta\varphi_1 = \cdots = \Delta\varphi_u = \Delta\varphi$. For convenience, we further set $\Delta\gamma$ and $\Delta\varphi$ to be the same small positive number ϵ.
- length of each time segment ΔT.
- number of time segments A.

Consequently, the time length of the entire trajectory, $T = A\Delta T$ is also determined. The finite-difference shadowing algorithm is given by the following procedure list.

1. Integrate equation (2.7) for sufficiently long time so that the trajectory lands onto the attractor. Then, set $t = 0$.
2. Generate initial conditions of homogeneous and inhomogeneous tangent solutions.
   a) Generate a $M \times u$ random matrix $W^0 = [w_1^0, \cdots w_u^0]$.
   b) Compute $W^{0\perp} = [w_1^{0\perp}, \cdots w_u^{0\perp}]$, whose column vectors are orthogonal to $f(t = 0)$.

c) Perform reduced QR factorization: $W^{0\perp} = Q_0 R_0$, where $Q_0 = [q_{01}, \cdots q_{0u}]$. Since the span of columns in Q_0 is the same as that of $W_0^{\perp}$, columns in Q_0 are also orthogonal to $f(t=0)$. Q_0 will be the initial conditions for homogeneous tangent solutions.

d) Set the initial condition for the inhomogeneous tangent solution: $v_0'(t_0) = 0$.

3. For $i = 0$ to $A-1$, on segment i, where $t \in [t_i, t_{i+1}]$ do:

a) Compute primal solutions and their related quantities.

i. Compute the base trajectory $x(t)$ for $t \in [t_i, t_{i+1}]$ by integrating the primal system in equation (2.7).

ii. Compute the instantaneous objective function $\Phi(t)$ for the base trajectory.

iii. Compute and store averaged objective on this segment, denoted by Φ_i, and objective at the end of the segment $\Phi(t_{i+1})$.

b) Compute homogeneous tangent solutions and their related quantities.

i. For each $1 \le j \le u$, solve a solution $x_{ij}^w(t), t \in [t_i, t_{i+1}]$, of the primal system in equation (2.7), with initial condition $x_{ij}^w(t_i) = x(t_i) + \epsilon q_{ij}$. Here $x(t_i)$ is the base trajectory at the beginning of segment i; q_{ij} is given by step 2c for the 0-th segment and by step 3(b)v for later segments.

ii. The homogeneous tangent $w_{ij}(t)$ for $t \in [t_i, t_{i+1}]$ with initial condition $w_{ij}(t_i) = q_{ij}$ is approximated by:

$$w_{ij}(t) \approx \frac{x_{ij}^w(t) - x(t)}{\epsilon} \,. \tag{4.13}$$

Define an $M \times u$ matrix: $W_i(t) = [w_{i1}(t), \cdots w_{iu}(t)]$, $t \in [t_i, t_{i+1}]$.

iii. Compute orthogonal projection $W_i^{\perp}(t) = [w_{i1}^{\perp}(t), \cdots w_{iu}^{\perp}(t)]$ via:

$$w_{ij}^{\perp}(t) = w_{ij}(t) - \frac{f^T(t) w_{ij}(t)}{f^T(t) f(t)} f(t) \,, \tag{4.14}$$

iv. Compute and store the covariant matrix C_i on segment i, defined as:

$$C_i = \int_{t_i}^{t_{i+1}} (W_i^{\perp})^T W_i^{\perp} dt. \tag{4.15}$$

This C_i is the covariant matrix on the i-th segment, similar to that defined in equation (2.16).

v. Perform reduced QR factorization: $W_i^{\perp}(t_{i+1}) = Q_{i+1} R_{i+1}$, where Q_{i+1} can be written in column vectors: $[q_{i+1,1}, \cdots q_{i+1,u}]$.

vi. For each $1 \leq j \leq u$, compute and store ξ_{ij}^w:

$$\xi_{ij}^w = \frac{(w_{ij}(t_{i+1}))^T f(x(t_{i+1}))}{f(x(t_{i+1}))^T f(x(t_{i+1}))} \,. \tag{4.16}$$

Above term is $\psi(w_{ij})$ in equation (4.3) evaluated at t_{i+1}.

vii. For each $0 \leq j \leq u$, evaluate the instantaneous objective function on the trajectory with perturbed initial condition, $u_{ij}^w(t)$. We denote this perturbed objective function by $\Phi_{ij}^w(t), t \in [t_i, t_{i+1}]$. Compute and store the perturbation in the time integration of the objective function:

$$\tilde{\Phi}_{ij}^w = \frac{1}{\epsilon} \int_{t_i}^{t_{i+1}} \Phi_{ij}^w(t) - \Phi(t)\, dt \,. \tag{4.17}$$

c) Compute inhomogeneous tangent solutions and their related quantities.

i. Solve a solution $u_i'(t), t \in [t_i, t_{i+1}]$ of the primal system with parameter $\gamma + \epsilon$, and initial condition $x_i'(t_i) = x(t_i) + \epsilon v_i'(t_i)$. Here $x(t)$ is the base trajectory; $v_i'(t_i)$ is given by step 2d for 0-th segment and by step 3(c)v for later segments.

ii. The inhomogeneous tangent $v_i'(t)$ for $t \in [t_i, t_{i+1}]$ with initial condition $v_i'(t_i)$ is approximated by:

$$v_i' \approx \frac{x_i' - x}{\epsilon} \,. \tag{4.18}$$

iii. Compute the orthogonal projection $v_i'^{\perp}(t)$, $t \in [t_i, t_{i+1}]$ via:

$$v_i'^{\perp} = v_i' - \frac{f^T v_i'}{f^T f} f \,, \tag{4.19}$$

iv. Compute and store

$$d_i = \int_{t_i}^{t_{i+1}} W_i^{\perp T} v_i'^{\perp} dt. \tag{4.20}$$

This d_i is similar to that defined in equation (2.16).

v. Orthogonalize $v_i'^{\perp}(t_{i+1})$ with respect to $W_{i+1}^{\perp}(t_{i+1}) = Q_{i+1}$ to obtain the initial condition of the next time segment:

$$v_{i+1}'(t_{i+1}) = v_i'^{\perp}(t_{i+1}) - Q_{i+1} b_{i+1}, \tag{4.21}$$

where b_{i+1} is defined as:

$$b_{i+1} = Q_{i+1}^T v_i'^{\perp}(t_{i+1}) \,, \tag{4.22}$$

and b_{i+1} should be stored.

vi. Compute and store ξ_i':

$$\xi_i' = \frac{(v_i'(t_{i+1}))^T f(x(t_{i+1}))}{f(x(t_{i+1}))^T f(x(t_{i+1}))} \,. \tag{4.23}$$

Above term is $\psi(v_i')$ in equation (4.3) evaluated at t_{i+1}.

vii. Evaluate the instantaneous objective function on the perturbed trajectory $u_i'(t)$. We denote this perturbed objective function by $\Phi_i'(t), t \in [t_i, t_{i+1}]$. Compute and store the perturbation in the time integration of the objective function:

$$\tilde{\Phi}_i' = \frac{1}{\epsilon} \int_{t_i}^{t_{i+1}} \Phi_i'(t) - \Phi(t)\, dt\,. \tag{4.24}$$

4. Solve the non-intrusive shadowing problem:

$$\begin{aligned} \min_{\{a_i\}} \sum_{i=0}^{A-1} \frac{1}{2} a_i^T C_i a_i + d_i^T a_i \\ \text{s.t. } a_i = R_i a_{i-1} + b_i \quad i = 1, \ldots, A-1. \end{aligned} \tag{4.25}$$

This is a least-squares problem in $\{a_i\}_{i=0}^{i=A-1}$, where $a_i \in \mathbb{R}^u$ for each i. We give a suggestion on how to solve this least-squares problem in the next subsection.

5. Compute

$$\rho(\Phi) = \frac{1}{A} \sum_{i=0}^{A-1} \Phi_i\,.$$

The derivative can be computed by:

$$\delta\rho(\Phi) \approx \frac{1}{T} \sum_{i=0}^{A-1} \left[\tilde{\Phi}_i' + \sum_{j=1}^{u} a_{ij} \tilde{\Phi}_{ij}^w + \left(\xi_i' + \sum_{j=1}^{u} a_{ij} \xi_{ij}^w \right) (\rho(\Phi) - \Phi(t_{i+1})) \right]. \tag{4.26}$$

Here $\tilde{\Phi}_i'$ is defined in equation (4.24), $\tilde{\Phi}_{ij}^w$ is defined in equation (4.17), ξ_i' is defined in equation (4.23), and ξ_{ij}^w is defined in equation (4.16).

□

We first remark that there is no need to store u_i, v_i' or W_i on the entire trajectory if we are only interested in the sensitivity. The quantities that finite-difference shadowing needs are C_i, d_i, R_i, b_i used in the minimization problem equation (4.25), and $\tilde{\Phi}_i'$, $\tilde{\Phi}_{ij}^w$, ξ_i', ξ_{ij}^w, $\Phi(t_{i+1})$ and $\rho(\Phi)$ used in the sensitivity formula in equation (4.26): all of these quantities are either scalars, u-dimensional vectors or $u \times u$ matrices. We should also store u_i, v_i' or W_i at the end time of the last segment for resuming the algorithm or lengthening the trajectory.

The integrations in equation (4.15), (4.17), (4.20), and (4.24), can certainly be computed by summation over all time steps in the current time segment. Alternatively, these integrations can be approximated by summation over several snapshots. For example, the integration in equation (4.20) can be approximated by:

$$d_i \approx \frac{1}{2} \left(W_i^{\perp T} v_i'^{\perp}(t_i) + W_i^{\perp T} v_i'^{\perp}(t_{i+1}) \right) \Delta T\,. \tag{4.27}$$

Correspondingly, the finite-difference approximations in equation (4.13) and (4.18) and the orthogonal projection in equation (4.14) and (4.19), now need be done only at the beginning and the end of a time segment. Although taking snapshots does not reduce the computational complexity, it reduces data storage. The idea of taking snapshots was also used in the multiple-shooting shadowing method developed by Blonigan [9].

The large part of the finite-difference shadowing algorithm is to compute $\{a_i\}_{i=0}^{A-1}$, using which we can construct the shadowing direction as shown in [59]: this does not use any knowledge of the instantaneous objective function $\Phi(u, s)$. Hence the marginal cost for one more objective is almost negligible, provided that we have determined all objectives before we run finite-difference shadowing. If we stored all information generated during the computation, including all those primal solutions, then we may also add more objectives for very little cost, after the computation is done.

For one more parameter γ, $\delta f := \partial f/\partial\gamma$ is changed, hence v' is changed; thus we need to recompute $\{a_i\}_{i=0}^{A-1}$, and the shadowing direction is also changed. However, homogeneous tangents W does not depend on δf, hence the marginal cost for one more parameter in finite-difference shadowing is to compute another v', and to solve again the non-intrusive shadowing problem in equation (4.25), whose cost is typically much lower than computing tangent solutions. As a result, the marginal cost for one more parameter is about $1/u$ of the total cost, provided that all parameters are determined before we run finite-difference shadowing.

Solving the non-intrusive shadowing problem

Here we give a suggestion on how to solve the minimization problem in equation (4.25). The Lagrange function is:

$$\sum_{i=0}^{A-1}\left(\frac{1}{2}a_i^T C_i a_i + d_i^T a_i\right) + \sum_{i=1}^{A-1}\lambda_i^T\left(a_i - R_i a_{i-1} - b_i\right) \ . \tag{4.28}$$

The Lagrange multiplier method tells us the minimizer for the non-intrusive shadowing problem is achieved at the solution of the following linear equation systems:

$$\begin{bmatrix} C & B^T \\ B & 0 \end{bmatrix}\begin{bmatrix} a \\ \lambda \end{bmatrix} = \begin{bmatrix} -d \\ b \end{bmatrix} \ , \tag{4.29}$$

where the block matrices $C \in \mathbb{R}^{uA \times uA}$, $B \in \mathbb{R}^{(uA-u)\times uA}$, vectors $a, d \in \mathbb{R}^{uA}$, and $\lambda, b \in \mathbb{R}^{uA-u}$. More specifically,

$$C = \begin{bmatrix} C_0 & & & \\ & C_1 & & \\ & & \ddots & \\ & & & C_{A-1} \end{bmatrix}, \quad B = \begin{bmatrix} -R_1 & I & & & \\ & -R_2 & I & & \\ & & \ddots & \ddots & \\ & & & -R_{A-1} & I \end{bmatrix},$$
$$a = \begin{bmatrix} a_0 \\ \vdots \\ a_{A-1} \end{bmatrix}, \quad \lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{A-1} \end{bmatrix}, \quad d = \begin{bmatrix} d_0 \\ \vdots \\ d_{A-1} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_{A-1} \end{bmatrix}, \tag{4.30}$$

where matrices $C_i, R_i \in \mathbb{R}^{u\times u}$, and vectors $a_i, \lambda_i, d_i, b_i \in \mathbb{R}^u$.

We can solve the Schur complement of equation (4.29) for λ:

$$-BC^{-1}B^T\lambda = BC^{-1}d + b\,, \tag{4.31}$$

where C^{-1} can be computed via inverting each diagonal block in C. Then we compute a by:

$$a = -C^{-1}(B^T\lambda + d)\,. \tag{4.32}$$

The above formula for solving the least-squares problem in finite-difference shadowing can as well be used in non-intrusive shadowing [59], which solves the same least-squares problem. Moreover, if we use snapshots at the beginning of each time segment to replace the inner products between tangent solutions, then due to the orthonormalization procedures we have $C_i = I, d_i = 0$, which further eases implementation.

Remarks on implementations

We first notice readers that homogeneous tangent solutions computed in finite-difference shadowing can also be used to compute LEs. More specifically, Benettin showed in [5] that almost surely, in the long-time limit, the volume growth rate of the parallelepiped spanned by, say u, randomly initialized homogeneous tangent solutions, will be almost the same as the growth rate of the parallelepiped spanned by the first u CLVs. Now the $u+1$-th LE can be computed by subtracting the volume growth rate of the parallelepiped spanned by u homogeneous tangent solutions from the growth rate of the parallelepiped spanned by all previous u plus one new homogeneous tangent solutions.

A caveat in Benettin's result is that when applied to a finitely long trajectory, LEs may not show up in the exact descending order. For example, if the random initial condition of the first homogeneous tangent solution happens to have only very small component in the direction of the first CLV, then after finite time, we may still only observe the first tangent solution being dominated by the second CLV. The same concern applies to non-intrusive shadowing and finite-difference shadowing, that is, we should typically compute some more

homogeneous tangent solutions than exactly u, in case the random initial conditions does not contain enough unstable components to cancel the exponential growth in v'. A sufficient u for a particular set of initial conditions can be identified as that Benettin's algorithm has confidently given all positive LEs.

In fact, we only require that the number of homogeneous tangent to be u', for any $u' \geq u$; however, we do not need to know u a priori. First, we can add tangent solutions to finite-difference shadowing inductively. Assume that we currently have u tangent solutions, then for equation (4.25), when adding one more tangent solution, then coefficients arrays d_i and b_i should each be augmented by one more entry, while the old coefficient arrays are not changed inside the new arrays; C_i, R_i should be augmented by one row and one column; for equation (4.26), we should further compute $\tilde{\Phi}^w_{i,u+1}$ and $\xi^w_{i,u+1}$ for all i. Second, due to the last comment, we can run Benettin's algorithm each time a new homogeneous tangent solution is added, and stop when all positive LEs has appeared, at which time we would have a big enough u.

We discuss how to select perturbation coefficient ϵ and segment length ΔT. These two algorithm parameters are mainly constrained by the requirement that finite-differences adequately approximate tangent solutions. Large ϵ would immediately yield a perturbed trajectory out of the linear approximation region; on the other hand, too small an ϵ would lead to large computation error when subtracting the perturbed trajectory from the base trajectory. Large ΔT would also allow the perturbed trajectory eventually falling out of the linear approximation region; small ΔT would lead to high computational cost. The solution is to run a linearity test. uore specifically, on the segment $[t_0, t_1]$, first select an ϵ, then compute w_{01} by equation (4.13) from a random initial condition of unit length, and compute v'_0 by equation (4.18) from a zero initial condition. Then compute again using 2ϵ. We select ϵ when the $w_{01}(t_1)$ and $v'_0(t_1)$ computed using ϵ and 2ϵ relatively differ less than some small number, say $\delta = 0.01$. Now the finite-difference error leads to a δ relative error in v and so hence in the sensitivity. Notice we may need to adjust the unit of parameter s, or adjust $\Delta\gamma$ and $\Delta\varphi$ separately, if we can not find a common ϵ such that both w_{01} and v'_0 pass the linearity test.

The total time length T is determined by the convergence history of sensitivity or by the computational cost requirement. Typically, T is determined empirically as the time when the sensitivity computed by finite-difference shadowing converges to within the uncertainty bound we desire. Another possibility is to stop computation when the limited time or computation resource has passed. We choose the latter criteria later in this chapter, since we want to compare the cost of finite-difference shadowing to that of solving the primal system. We have found that typically shadowing methods require a shorter trajectory to compute sensitivity than that required by the primal solver to reflect average behavior.

For problems with a large number of unstable Lyapunov exponents, we suggest switching from finite-difference shadowing to non-intrusive shadowing or adjoint shadowing, where we can take advantage of the vectorization of linear solvers and further accelerate computing homogeneous tangent or adjoint solutions, as discussed in [58]. Still, the cost of non-intrusive shadowing methods can get larger when u is larger and the system becomes more chaotic. However, such cost increase is typical for many numerical methods, for example, even

computing long-time averages should take longer time to converge for more chaotic systems. For very chaotic systems, if we still have $u \ll M$, then non-intrusive shadowing and adjoint shadowing may still be competitive in computational efficiency; at least, the idea of the 'non-intrusive' formulation, that is, reducing the computation to unstable subspace, will still be important. Current investigation on some computer-simulated fluid systems all have $u \leq 0.1\%M$ [59, 29, 6, 57], but we do not yet have a good estimation for very chaotic systems. On the other hand, there are systems with $u \approx M$, such as Hamiltonian systems, which has equally many stable and unstable CLVs; for these systems, non-intrusive shadowing or adjoint shadowing may not be faster than other methods.

4.3 Application on a turbulent three-dimensional flow over a cylinder

Physical problem and numerical simulation

Before using finite-difference shadowing to compute sensitivities, we first describe the physical problem of the 3-D flow past a cylinder. The front view of the geometry of the entire flow field is shown in figure 4.2. The diameter of the cylinder is $D = 0.25 \times 10^{-3}$. The span-wise width is $Z = 2D$. The free-stream conditions are: density $\rho' = 1.18$ (do not confuse with SRB measure, which is denoted by ρ), pressure $P = 1.01 \times 10^5$, temperature $T = 298$, dynamic viscosity $\mu = 1.86 \times 10^{-5}$. The free stream flow is in the x-direction, with the velocity U being one of the system parameters, and for the base case $U_0 = 33.0$. The flow-through time t_0, defined as the time for U_0 flowing past the cylinder, is $t_0 = D/U_0 = 7.576 \times 10^{-6}$. The Reynolds number of the base case is $Re = 525$ and Mach number is 0.1. The cylinder can rotate around its center with rotational speed ω, which is the second system parameter for our problem. ω is measured in revolutions per unit time, and its positive direction is counter-clockwise, as shown in figure 4.2. For the cylinder to rotate one cycle per flow-through time, $\omega_0 = 1/t_0 = 1.32 \times 10^5$.

Then we look at settings for numerical simulations. We use a block-structured mesh with 3.7×10^5 hexahedra. 2-D slices of the mesh are shown in figure 4.3. The span-wise direction has 48 cells. The CFD solver we use is CharLES developed at Cascade Technologies [14], using which we perform the implicit large eddy simulation, where the numerical error of the discretization scheme serves as the sub-grid scale Reynolds stress model. The accuracy of the solver is formally 2nd order in space and 3rd order in time. The span-wise boundary uses periodic boundary conditions; the left boundary uses a convective boundary condition [22]; the right boundary uses the Navier-Stokes characteristic boundary conditions (NSCBC) boundary condition [64]. The time step size is $\Delta t = 9.8 \times 10^{-9} = 1.30 \times 10^{-3} t_0$. In order to trigger the 3-D flow faster in our numerical simulation, we add a small white noise to the initial condition, whose magnitude is about 0.1% of the inflow.

2-D snapshots of the flow field at $U = U_0$ are shown in figure 4.4. The flow is chaotic and 3-D. The same physical problem has been investigated through experiments by Williamson

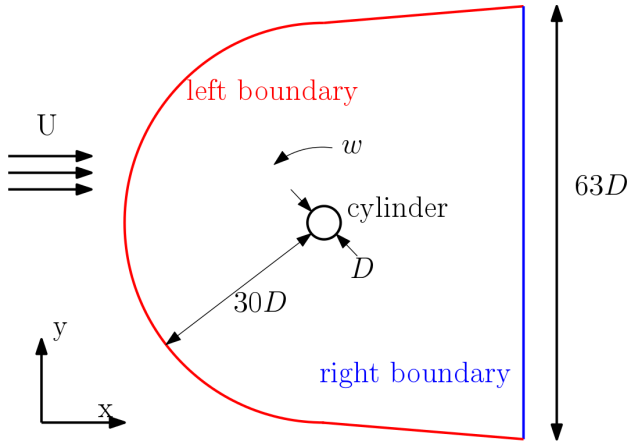


Figure 4.2: Geometry used in the simulation of a flow over a 3-D cylinder. The span-wise extent of the computational domain is $Z = 2D$. The positive direction of the cylinder rotational speed ω is counter-clockwise.

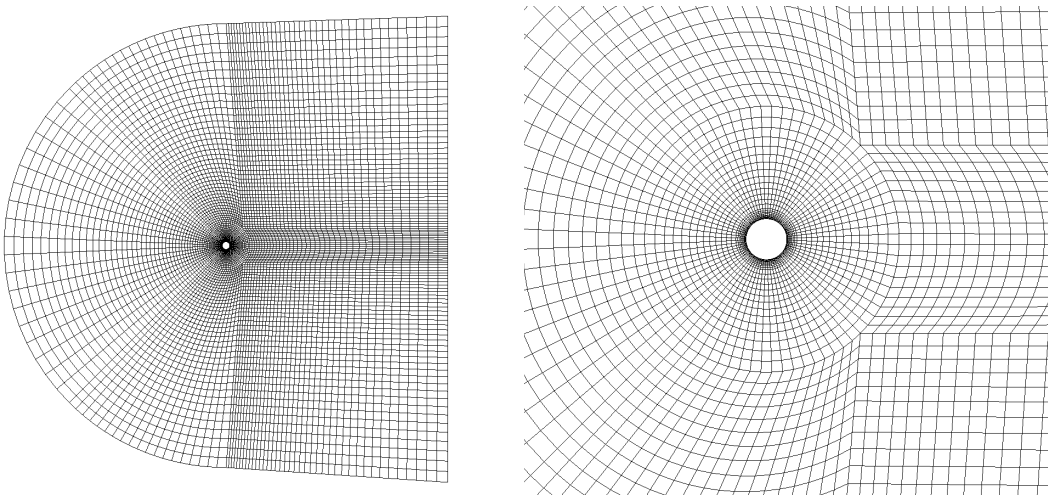


Figure 4.3: Left: 2-D slice of the mesh over the entire computational domain. Right: zoom around the cylinder. This is a block-structured mesh with 3.7×10^5 hexahedra. The span-wise direction has 48 cells.

and Roshko [84], and through numerical simulations by Mittal and Balachandar [51]. The comparison of the Strouhal number and the averaged drag coefficient is shown in table 4.1. Here the Strouhal number is defined by $S_t = fD/U$, where f is the main frequency of the vortex shedding, selected as the location of the peak in the Fourier transformation of the lift history; the drag coefficient $C_D = D_r/(0.5\rho' U^2 DZ)$, where D_r is the drag. As we can see, our simulation matches previous experimental and numerical results.

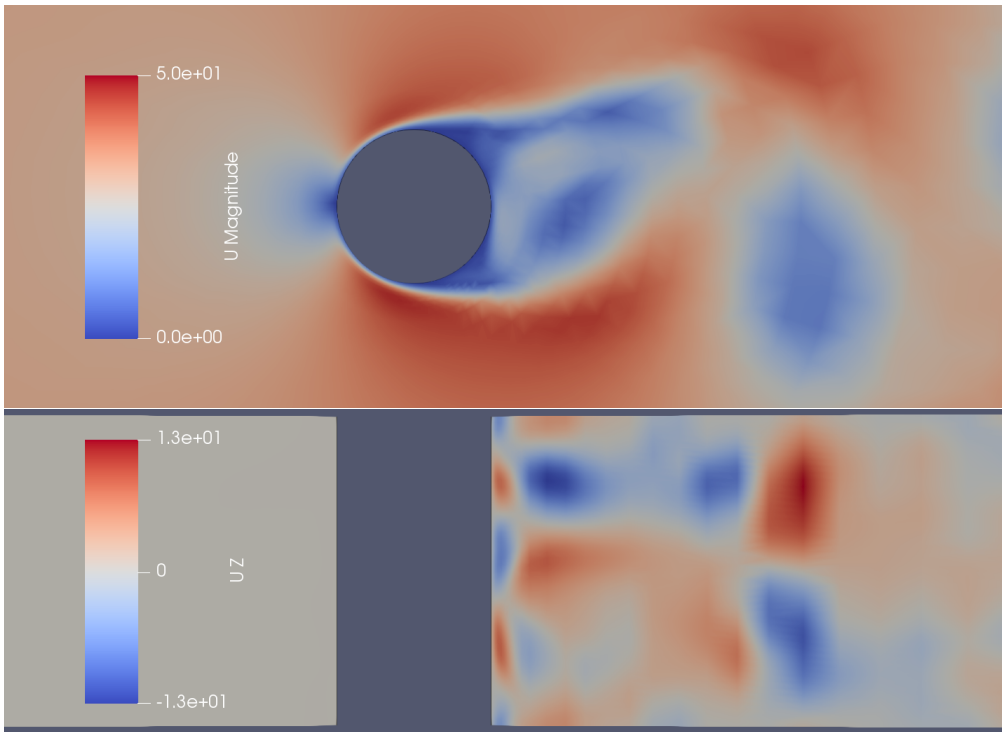


Figure 4.4: A typical snapshot of the flow field. Top: cross-section along the x-z plane, plotted by magnitude of velocity. Bottom: cross-section along the x-y plane, plotted by the z-component of velocity. The bottom picture shows the flow is 3-D.

	S_t	C_D
Current work	0.21	1.22
Previous 2-D simulation [51]	0.22	1.44
Previous 3-D simulation [51]	0.22	1.24
Previous experiment [84]	0.21	1.15

Table 4.1: Comparison of our simulation with previous results in literatures by the Strouhal number S_t and the averaged drag coefficient C_D.

Results

We apply finite-difference shadowing to this 3-D chaotic flow past a cylinder. [1] We consider two system parameters: free-stream velocity U and the rotational speed of the cylinder ω.

[1]The python package 'fds' implementing finite-difference shadowing is available at https://github.com/qiqi/fds. The particular files related to the application in this section are in fds/apps/charles_cylinder3D.

We will normalize U by U_0, time by t_0, and ω by ω_0. We investigate the effect of U on two objectives: averaged drag force $\rho(D_r)$, and averaged base suction pressure $\rho(S_b)$, which is defined as the pressure drop at the base of the cylinder in comparison to the free stream. We will normalize $\rho(D_r)$ by $F_0 = 0.5\rho' U_0^2 DZ = 8.031 \times 10^{-5}$, and $\rho(S_b)$ by $P_0 = 0.5\rho' U_0^2 = 642.5$. For ω, we look at its effect on averaged lift $\rho(L)$ and averaged lift square $\rho(L^2)$. We will normalize $\rho(L)$ by F_0, and $\rho(L^2)$ by $F_0^2 = 6.450 \times 10^{-9}$.

Each objective $\rho(\Phi)$ is approximated by $\Phi_{T'}$, which is averaged over $T' = 8.7 \times 10^{-3} = 1148t_0$. In figure 4.7, we compute each objective with 7 different parameters in order to reflect the trend between the parameter and the objective: this trend will help us validate the sensitivities computed by finite-difference shadowing. For the 7 primal simulations, a total number of 6.1×10^6 steps of primal solutions are computed. As we will see later, T' is chosen so that costs of finite-difference shadowing and primal simulations are similar, and we will show that the sensitivity computed by finite-difference shadowing matches the trend suggested by the primal simulation.

To get the uncertainty of averaging objectives over finite time, we divide the history of $\Phi(t)$ into 5 equally long parts. Denote the objectives averaged over each of the five parts by $\Phi_1, ...\Phi_5$. The corrected sample standard deviation between them are:

$$\sigma' = \sqrt{\frac{1}{4}\sum_{k=1}^{5}(\Phi_k - \Phi_{T'})^2}. \tag{4.33}$$

We assume that the standard deviation of $\Phi_{T'}$ is proportional to $T'^{-0.5}$. Thus, we use $\sigma = \sigma'/\sqrt{5}$ as the standard deviation of $\Phi_{T'}$. We further assume $\pm 2\sigma$ yields the 95% confidence interval for $\Phi_{T'}$. Objectives for different parameters are shown in figure 4.7, where the bars indicate the 95% confidence intervals.

Each segment in non-intrusive shadowing has 200 time steps, thus the segment length $\Delta T = 1.96 \times 10^{-6} = 0.259t_0$. We set $\epsilon = 10^{-4}$ and the number of segments $A = 400$. Here ϵ and ΔT have been checked by the linearity test we discussed in section 4.2, and A is chosen such that the cost of finite-difference shadowing is similar to the primal solver.

Our current implementation can not yet inductively add tangent solutions as we discussed in section 4.2. Currently, we can only do trial and error to find a large enough u, and we selected 40 as our initial guess. To verify that we have used a large enough u, we use the algorithm given by Benettin [5] to find the number of unstable CLVs. Confidence intervals of LEs are estimated by the smallest interval which bounds the history of an LE and whose size shrinks as $T^{-0.5}$: this method is the same as in [59]. Figure 4.5 shows that there are about 17 unstable CLVs, indicating u is large enough. The LEs, CLVs and shadowing directions of the same physical problem, on both the current mesh and a finer mesh with twice as many cells, are studied with more details in [57], which shows that for both meshes, (1) there are only a few unstable CLVs, (2) CLVs are active at different area in the flow field, indicating angles are large between CLVs whose indices are far-apart, and (3) shadowing directions exists and can give pretty accurate sensitivities. Moreover, [57] also plots snapshots of CLVs and shadowing directions.

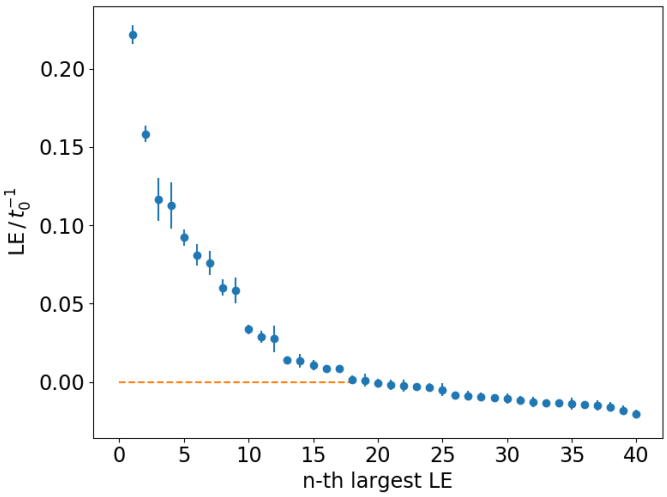


Figure 4.5: Confidence intervals of the largest 40 Lyapunov exponents (LE), normalized by t_0^{-1}. The largest LE is $0.22t_0^{-1}$, meaning in one flow-through time t_0, the norm of the first CLV becomes $e^{0.22} = 1.25$ times larger.

Using above settings, the cost of finite-difference shadowing is from integrating the primal solution over $400 \times 200 \times 42 = 3.36 \times 10^6$ time steps. Here $A = 400$ is the number of segments, 200 is the number of time steps in each segment. $u + 2 = 42$ is the number of primal solutions computed: in the finite-difference shadowing we need one inhomogeneous tangent and $u = 40$ homogeneous tangents. Each tangent solution is approximated by the difference between a perturbed trajectory and the same base trajectory: those are 42 primal solutions in total. The total cost of finite-difference shadowing is smaller than computing averaged objectives for the 7 parameters in figure 4.7. We also remind readers that the marginal cost for a new objective in finite-difference shadowing is negligible, and the marginal cost for a new parameter is about 1/40 of the total cost.

The confidence intervals of sensitivities computed by finite-difference shadowing are estimated by the smallest interval which bounds the history of the sensitivity and whose size shrinks as $T^{-0.5}$: this method is given in more detail in [59]. Figure 4.6 shows history plots of sensitivities for different pairs of parameter and objective. In figure 4.7 the green wedges are confidence intervals of sensitivities. Notice that $\rho(L^2)$ attains minimum at $\omega = 0$, thus the sensitivity should be almost zero: this is why the last plot in figure 4.6 appears not to converge, since the sensitivity is already very small.

Figure 4.7 validates the sensitivities computed with finite-difference shadowing, since the sensitivities matches the trend between objectives and parameters. Moreover, the cost of computing sensitivities by finite-difference shadowing is similar to revealing the trend by evaluating objectives at 7 different parameters.

Another way to compute sensitivities is to perform some function regression among

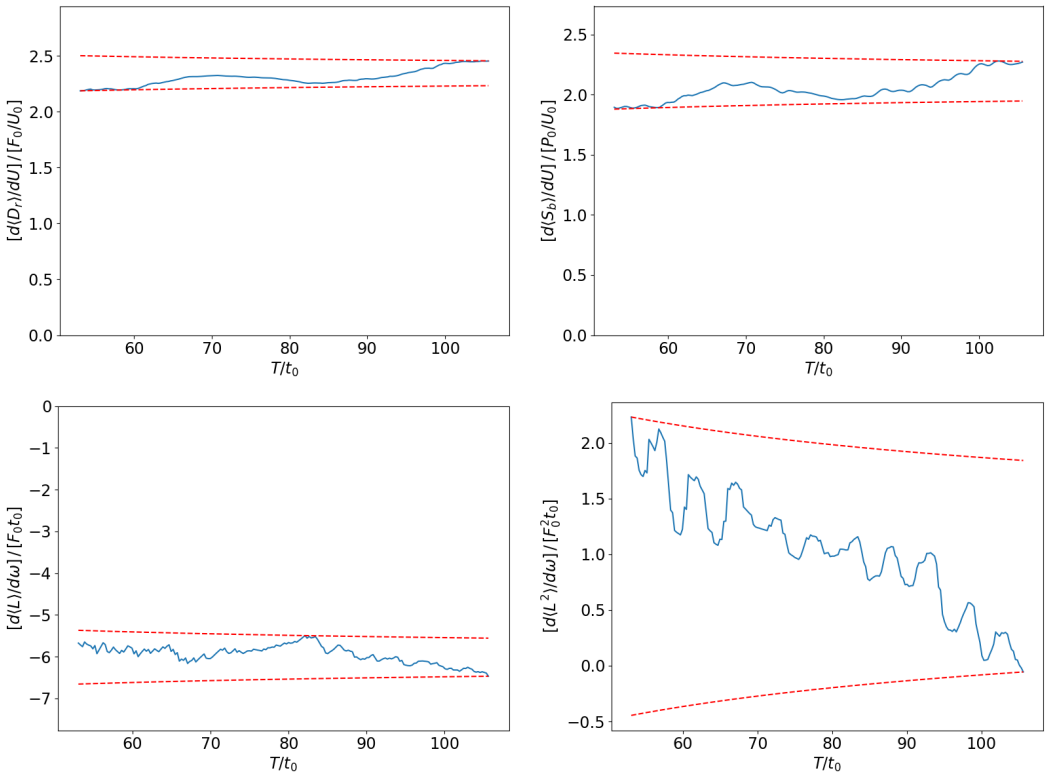


Figure 4.6: History plots of sensitivities computed by finite-difference shadowing. All axes are normalized. The $\langle \cdot \rangle$ in the annotation of y-axis means to take average over SRB measure ρ. The dashed lines indicate the smallest encompassing interval whose size shrinks as $T^{-0.5}$.

objectives evaluated with different parameters. However, this regression method requires prescribing a function prototype, the choice of which is typically not obvious. Even worse, giving confidence intervals to sensitivities computed via regression requires prescribing on the space of function prototypes a probability measure, the choice of which is even less obvious.

4.4 Conclusions

This chapter presents the finite-difference shadowing algorithm for computing sensitivities of chaotic dynamical systems. Unlike non-intrusive shadowing, finite-difference shadowing does not require tangent solvers, and it can be implemented with little modification to existing numerical simulation software. Numerical results show finite-difference shadowing can compute pretty accurate sensitivity for the 3-D chaotic flow over a cylinder under Reynolds number 525. This result also indicates that for real-life engineering problems, finite-difference shadowing can be an affordable method to compute the sensitivity.

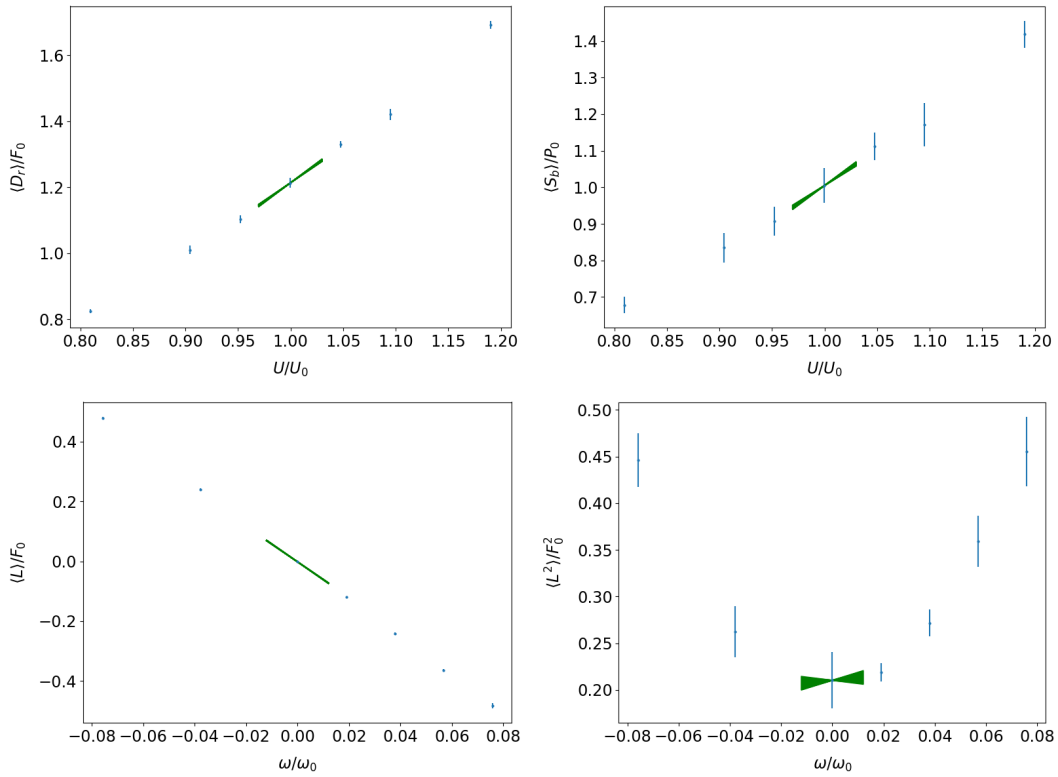


Figure 4.7: 95% confidence intervals of sensitivities computed by finite-difference shadowing, indicated by the green wedge. Blue vertical bars indicate 95% confidence intervals of averaged objectives. The $\langle \cdot \rangle$ in the annotation of y-axis means to average over ρ. Here all objectives and parameters are normalized.

There are several possible future research for the finite-difference shadowing algorithm. First, we may investigate the magnitude of the error induced by the finite-difference approximation. We may also investigate if the convergence of the finite-difference shadowing depends on mesh sizes, time step size, and the finite-difference coefficient ϵ. We can as well experiment different ways of using snapshots to approximate integrations. For readers who are convinced that finite-difference shadowing is useful, we suggest to further implement non-intrusive shadowing and adjoint shadowing with vectorized linear solvers, which are faster than finite-difference shadowing, and could be applied to more chaotic problems with acceptable cost.

Chapter 5

Non-intrusive adjoint shadowing algorithm

We develop the adjoint version of the non-intrusive shadowing algorithm, also known as the Non-Intrusive Least Squares Adjoint Shadowing (NILSAS) algorithm, which performs adjoint sensitivity analysis of chaotic systems via computing the adjoint shadowing direction. Non-intrusive adjoint shadowing constrains its minimization to the adjoint unstable subspace, and can be implemented with little modification to existing adjoint solvers. The computational cost of non-intrusive adjoint shadowing is independent of the number of parameters. We demonstrate non-intrusive adjoint shadowing on the Lorenz 63 system and a turbulent three-dimensional flow over a cylinder.

The marginal cost for a new parameter in non-intrusive shadowing is only computing one extra inhomogeneous tangent solution. Yet for cases where there are many parameters and only a few design objectives, an adjoint version of non-intrusive shadowing is desired. A continuous adjoint version appeared in the first publication of non-intrusive shadowing [60]; however, this version of lacks the constraint on the neutral subspace, which will be explained in this chapter. Blonigan [7] developed a discrete adjoint version of non-intrusive shadowing, which was later implemented by Chandramoorthy et al. [18] using automatic differentiation. In comparison to this discrete adjoint non-intrusive shadowing, non-intrusive adjoint shadowing does not require tangent solvers, and requires less modification to existing adjoint solvers, and the simplicity of the formula of non-intrusive adjoint shadowing should also give it more robustness and perhaps better convergence.

Recently, we defined the adjoint shadowing direction for both hyperbolic flows and diffeomorphisms, which is a bounded inhomogeneous adjoint solution with several other properties [54]. We showed that the adjoint shadowing direction exists uniquely on a given trajectory, and can be used for adjoint sensitivity analysis. Adjoint shadowing direction is defined using only adjoint flows, giving us a chance to get rid of tangent solvers in our algorithm, and arrive at a neat formula.

This chapter presents the non-intrusive adjoint shadowing algorithm, where we construct a least squares problem to approximate the adjoint shadowing direction and then compute

adjoint sensitivity. The 'non-intrusive' formulation allows it be built using existing adjoint solvers, and more importantly, it allows the minimization be constrained to the unstable adjoint subspace. The main body of this chapter will be about continuous dynamical systems, and we briefly discuss non-intrusive adjoint shadowing for discrete systems in 5.6.

We organize the rest of this chapter as follows. First, we prepare our study by defining our problem and reviewing adjoint flows and adjoint shadowing directions; we also provide some intuitions to help understanding adjoint shadowing directions. Then we derive the non-intrusive adjoint shadowing algorithm. Then we present a detailed procedure list for our algorithm and give several remarks on the algorithm. Finally, we demonstrate non-intrusive adjoint shadowing on the Lorenz 63 system and a weakly turbulent three-dimensional (3D) flow over a cylinder.

5.1 Review on adjoint flow and adjoint shadowing

In this chapter we mainly deal with continuous-time systems, and leave the discrete case to the appendix of this chapter. For now we assume there is only one parameter γ; and in section 5.3 we will explain how we can compute sensitivities with respect to several parameters with almost no additional cost. Our goal of this chapter is to develop an algorithm computing the sensitivity, $\delta\rho(\Phi)$, whose marginal cost for a new parameter is negligible.

Adjoint flow

Definition 1. A homogeneous adjoint solution $w(t) : \mathbb{R} \to \mathbb{R}^m$ is a function which solves the homogeneous adjoint equation:

$$\frac{dw}{dt} + f_x^T w = 0, \tag{5.1}$$

where $f_x := \partial f/\partial x$ is the Jacobian matrix, and $\cdot^T$ is the matrix transpose. An inhomogeneous adjoint solution is a function $v(t) : \mathbb{R} \to \mathbb{R}^m$ which solves:

$$\frac{dv}{dt} + f_x^T v = g(t), \tag{5.2}$$

where $g(t) : \mathbb{R} \to \mathbb{R}^m$ is a vector-valued function of time.

In numerical implementations, we typically solve adjoint equations backward in time. This is because, as shown in [54], when solving backward in time, the dimension of the unstable adjoint subspace is the same as the unstable tangent subspace, which is typically much lower than m. On the other hand, if we solve the adjoint equation forward in time, the unstable subspace has much higher dimension, causing strong numerical instability.

Definition 2. In this chapter, an adjoint covariant Lyapunov vector (CLV) with adjoint Lyapunov exponent (LE) λ is a homogeneous adjoint solution $\overline{\zeta}(t)$ such that there is a constant C, for any $t_1, t_2 \in \mathbb{R}$,

$$\|\overline{\zeta}(t_1)\| \leq Ce^{\lambda(t_2-t_1)}\|\overline{\zeta}(t_2)\| \,. \tag{5.3}$$

Note that the time direction in the above definition is reversed: if the adjoint CLV grows exponentially backward in time, its exponent is positive. Adjoint CLVs with positive exponents are called unstable, those with negative exponents are stable, and those with zero exponent is are neutral. In this chapter, we sort adjoint CLVs by descending order of their exponents. The earliest mention of adjoint CLVs was by Kuptsov and Parlitz in [44]. For the purpose of defining adjoint shadowing directions, and deriving the non-intrusive algorithm, we recently proved the existence of adjoint CLVs for uniform hyperbolic systems, and found some relation between CLVs and adjoint CLVs.

Adjoint CLVs are homogeneous adjoint solutions whose norm grows exponentially, and the adjoint LE is measured backward in time. This is similar but also different from tangent CLVs, which are tangent solutions measured forward in time. The CLV structure for the adjoint flow is the same as the tangent flow. That is, the adjoint LE spectrum is the same as the tangent LE spectrum. Moreover, the subspace of CLVs with an exponent λ is perpendicular to the subspace of all adjoint CLVs with exponents not λ, and vice versa. If we can write the full set of CLVs as a matrix valued function of time, $W(t)$, then $W^{-T}(t)$, where $\cdot^{-T}$ is the inverse of transpose, is a matrix whose columns are adjoint CLVs: readers can verify that $W^{-T}(t)$ satisfies the properties listed above. Note that we do not know if CLVs and adjoint CLVs with the same exponent are perpendicular or parallel.

We assume our system is uniform hyperbolic and it has a bounded global attractor. Definition of hyperbolicity can be found in most textbook on dynamical system such as [43], and readers may also refer to [54] for a definition using the same notation as this chapter. Uniform hyperbolicity requires that the tangent space can be split into stable subspace, unstable subspace, and a neutral subspace of dimension one. Together with the boundedness of the attractor, we can show the angles between two subspaces of different sets of tangent CLVs are always larger than some positive angle. Since the adjoint equations have the same structure as the tangent ones, there is only one neutral adjoint CLV, and adjoint CLVs are always bounded away from each other [54].

Adjoint shadowing directions

In [54], the author defined adjoint shadowing directions, proved their unique existence on a given trajectory, and showed how to use them for adjoint sensitivity analysis. We briefly restate the main results in this subsection.

Definition 3. On a trajectory $u(t)$ on the attractor, for $t \geq 0$, the adjoint shadowing direction $\overline{v}^\infty : \mathbb{R}_+ \to \mathbb{R}^m$ is defined as a function with the following properties:

1. $\overline{v}^\infty$ solves the inhomogeneous adjoint equation:

$$\frac{d\overline{v}^\infty}{dt} + f_x^T \overline{v}^\infty = -\Phi_x \,, \tag{5.4}$$

where subscripts are partial derivatives, that is, $f_x = \partial f/\partial x$, $\Phi_x = \partial \Phi/\partial x$.

2. $\overline{v}^{\infty}(t=0)$ has zero component in the unstable adjoint subspace.
3. $\|\overline{v}^{\infty}(t)\|$ is bounded by a constant for all $t \in \mathbb{R}_+$.
4. The averaged inner-product of $\overline{v}^{\infty}$ and f is zero:

$$\rho(\langle \overline{v}^{\infty}, f \rangle) := \lim_{T\to\infty} \frac{1}{T} \int_0^T \langle \overline{v}^{\infty}(t), f(t) \rangle = 0\,, \tag{5.5}$$

where $\langle \cdot, \cdot \rangle$ is the inner-product on the Euclidean space.

We remind readers to distinguish the three different kinds of adjoint solutions we mentioned: homogeneous adjoint solutions, inhomogeneous adjoint solutions and adjoint shadowing directions. Homogeneous adjoint solutions are different from inhomogeneous ones, since homogeneous adjoint equations must have zero right-hand-sides. The adjoint shadowing direction is an inhomogeneous adjoint solution, but not any inhomogeneous adjoint solution: it must in extra have three more properties listed in the definition. In fact, one way to view non-intrusive adjoint shadowing is that we search the space of all inhomogeneous adjoint solutions to find one such that it mimics the other three properties. More specifically, we minimize the L^2 norm, and constrain the inner product with f: this derivation will be revealed in later sections.

Theorem 5. *For a uniform hyperbolic system with a global compact attractor, on a trajectory on the attractor, there exists a unique adjoint shadowing direction. Further, we have the adjoint sensitivity formula:*

$$\delta\rho(\Phi) \approx S.C. = \lim_{T\to\infty} \frac{1}{T} \int_0^T \langle \overline{v}^{\infty}, f_\gamma \rangle + \Phi_\gamma \, dt\,. \tag{5.6}$$

We explain the assumption of the adjoint shadowing theorem. First, if a dynamical system has a compact global attractor, it means there is a bounded set of states, or the attractor, such that no matter what initial condition the system starts from, the trajectory will eventually enter the attractor and never leave. Second, uniform hyperbolicity here mainly means that there is only one neutral CLV. Third, by the compactness, the angles between all CLVs are larger than a positive angle, regardless of where we are on the attractor.

Why do we make above assumptions in theories for shadowing methods? The main reason for assuming only one neutral CLV in shadowing methods is to prevent linear growth in inhomogeneous tangent/adjoint solutions. The main reason for global attractability is to ensure that shadowing trajectories are representative of the averaged behavior of the system. The main reason for compactness is because we want a bound for the projection operators projecting onto a particular subspace. Still, we remind readers that, in practice, shadowing methods may be effective beyond above assumptions, as to be discussed in section 5.3.

Rather than giving an explicit expression of adjoint shadowing directions, which can be found as well in [54], the definition is stated as a criterion, where we check several properties to determine if a function is indeed the adjoint shadowing direction. In fact, we forged this definition for designing the non-intrusive adjoint shadowing algorithm, which will be revealed in the next section.

Interpreting adjoint shadowing directions

We give several different perspectives to help readers build intuitions of adjoint shadowing directions. To start with, we revisit some formal descriptions of shadowing operators. The shadowing operator, denoted as $\mathcal{S}$, can be viewed roughly as mapping a vector-valued function $f_\gamma(t)$ to another vector-valued function $v^\infty(t)$, Here $f_\gamma(t)$ is the perturbation on f due to parameter perturbations; v^∞ is the tangent shadowing direction, which is first order approximation of the difference between the shadowing and the base trajectory. Note that $\mathcal{S}$ is a linear operator, and both $f_\gamma(t)$ and $v^\infty(t)$ are linear approximations. If we neglect the neutral CLV, we have roughly

$$\delta\rho(\Phi) \approx S.C. = \rho \left\langle v^\infty, \Phi_x \right\rangle = \rho \left\langle \mathcal{S}(f_\gamma), \Phi_x \right\rangle . \tag{5.7}$$

First we provide a utility point of view for adjoint shadowing directions, which is also an algebraic point of view. Riesz's representation theorem tells us that there is an adjoint operator $\overline{\mathcal{S}}$ such that

$$\rho \left\langle \mathcal{S}(f_\gamma), \Phi_x \right\rangle = \rho \left\langle f_\gamma, \overline{\mathcal{S}}(\Phi_x) \right\rangle . \tag{5.8}$$

The adjoint shadowing direction, $\overline{v}^\infty$, is $\overline{\mathcal{S}}(\Phi_x)$. Suppose now we have a computer program which approximately functions as $\overline{\mathcal{S}}$. If we have two parameters s_1, s_2, then they can perturb the governing equation by f_{s_1} and f_{s_2}, whereas Φ_x keeps the same. This means that we only need to run our adjoint program once, and use the result to inner-product with both f_{s_1} and f_{s_2}. For cases where there are many parameters s and a few objectives Φ, we only need to run our program a few times.

Then an implementation point of view. The adjoint operator of a matrix is simply its transpose. For many other cases, we can see that matrix transposition often appears as a crucial step in the formulation of adjoint operators. So once we derive an new adjoint formula of something, we may ask if it can be presented as neatly as transposing a matrix. Adjoint shadowing directions satisfy inhomogeneous adjoint equations, which is a linear ODE whose matrix is the transpose of the Jacobian matrix. Hence there is chance that algorithms, such as non-intrusive adjoint shadowing, do not differ too much from existing adjoint solvers. In fact, giving a neat recipe for adjoint shadowing direction is the main contribution of both the adjoint shadowing theorem and the non-intrusive adjoint shadowing algorithm, since the existence is already given by Riesz's representation theorem.

Finally, we provide a physical point of view. Denote the trajectory by x_t, assume the initial condition of our trajectory is fixed, and we perturb the state at t by δx_t, then the trajectory at a later time, $\tau \geq t$, is changed by $D_t^\tau \delta x_t$. Notice here we use δ to denote infinitesimal perturbations, which may also be thought as caused by perturbing some parameter γ, which controls the state of the trajectory at t. Now the change in $\Phi_{avg} := \frac{1}{T}\int_0^T \Phi dt$ satisfies:

$$T(\Phi_{avg} + \delta\Phi_{avg}) = \int_0^T \Phi(x(\tau) + \delta x(\tau), s) d\tau = \int_0^T \Phi(x, \gamma) d\tau + \int_t^T \Phi_x^T D_t^\tau \delta x_t d\tau .$$

Here Φ_x is transposed to comply with our notation that Φ_x is a column vector. Canceling the first term on both sides and diving both side by δx_t, we get

$$T\frac{\delta\Phi_{avg}}{\delta x_t} = \int_t^T \Phi_x^T D_t^\tau d\tau \,. \tag{5.9}$$

This is exactly $\overline{v}^*(t)$, the conventional adjoint solution at t.

Instead of assuming initial condition being fixed, we let it change according to that prescribed by the shadowing direction. Now $T\delta\Phi_{avg}/\delta x_t$ is the adjoint shadowing solution $\overline{v}^\infty(t)$. This means that the adjoint shadowing direction can be viewed as how a perturbation at t in the trajectory affects the objective, with the caveat that now both the past and future trajectory are perturbed to match the perturbation at t.

5.2 Deriving non-intrusive adjoint shadowing algorithm

The non-intrusive formulation

On a finite trajectory of time span $[0, T]$, the non-intrusive adjoint shadowing algorithm computes a $\overline{v}$ which approximates $\overline{v}^\infty$. Since the definition of the adjoint shadowing direction is similar to the tangent shadowing direction, it is not surprising that we can reuse the 'non-intrusive' formulation in non-intrusive shadowing, that is, we can find adjoint shadowing direction by a minimization in the unstable adjoint subspace.

In non-intrusive adjoint shadowing, we strictly enforce the first property of adjoint shadowing directions by constraining our solutions to inhomogeneous adjoint solutions. The second property is changed to a symmetric statement that stable component in $\overline{v}(T)$ should be $O(1)$, which can be easily satisfied. The third property is approximated by minimizing the L^2 norm of the inhomogeneous adjoint solution. The fourth property is strictly enforced by adding a constraint to our minimization problem. In this subsection, we explain why the $\overline{v}$ given by this reverse-engineering is a good approximation of $\overline{v}^\infty$.

Our algorithm strictly enforces the first property of $\overline{v}^\infty$. To do this, we represent the solution set of equation (5.4) as a particular solution plus the space of homogeneous solutions. We select the particular solution as the conventional inhomogeneous adjoint solution $\overline{v}^*$, which is defined as the solution of:

$$\frac{d\overline{v}^*}{dt} + f_x^T\overline{v}^* = -\Phi_x \,,\ \overline{v}^*(T) = 0 \,. \tag{5.10}$$

Then we select the collection of all adjoint CLVs, $\overline{Z} = [\overline{\zeta}_1, \cdots, \overline{\zeta}_m]$, all of which have terminal condition $\|\overline{\zeta}_j(T)\| = 1$, as the basis of the space of homogeneous solutions. Hence we can enforce the first property by considering candidates only in the following form for some $\{a_j\}_{j=1}^m$:

$$\overline{v} = \overline{v}^* + \sum_{j=1}^m a_j\overline{\zeta}_j = \overline{v}^* + \overline{Z}a \,, \tag{5.11}$$

where the coefficients $a = [a_1, ..., a_m]^T$ is a column vector. Another interpretation of our way of enforcing the first property is that, we want to start $\overline{v}$ from $\overline{v}^*$, and modify by adding adjoint CLVs to approximate $\overline{v}^\infty$; in other words, the coefficients a should be such that $Za \approx \overline{v}^\infty - \overline{v}^*$. We then use other properties to determine the coefficients for stable, unstable and neutral CLVs.

When defining adjoint shadowing directions in [54], the author was considering functions defined starting from time zero, whereas in our case here, adjoint solutions are solved from T backward in time. Hence, in order to keep the sensitivity formula, we change the second property to a symmetric statement, that is, we want the stable component in $\overline{v}(T)$ be in the order $O(1)$, meaning be bounded by a constant independent of T. Equivalently, we require coefficients for stable CLVs be $O(1)$. Another way to interpret is that, if this $O(1)$ condition is true, then since $\overline{v}^\infty(T)$ is $O(1)$, the stable component in $\overline{v}(T) - \overline{v}^\infty(T)$ is $O(1)$; now since stable CLVs decay exponentially fast, their contribution in $\overline{v} - \overline{v}^\infty$ can be neglected. This $O(1)$ condition is a loose requirement and, as we will see, it can be easily satisfied.

To mimic the boundedness in the third property of adjoint shadowing directions, we minimize $\|\overline{v}\|_{L^2}$, which determines the coefficients for the unstable CLVs. Indeed, this minimization removes significant unstable CLVs from $\overline{v} - \overline{v}^\infty$, since otherwise this difference would grow exponentially, and since $\overline{v}^\infty$ is bounded, $\overline{v}$ would have large L^2 norm.

We strictly enforce the fourth property of $\overline{v}$. This determines the coefficient for the neutral adjoint CLV, since as shown in [54], f is always orthogonal to non-neutral adjoint CLVs. Note also that the norm of the neutral adjoint CLV is bounded, unlike neutral tangent CLV, which can have linear growth. Hence we can allow its coefficient be $O(1)$ without jeopardizing the boundedness property we used earlier. In fact, the adjoint non-intrusive shadowing in [60] lacks exactly this constraint on the neutral adjoint CLV.

To summarize, we determine coefficients for unstable adjoint CLVs via a minimization, the coefficient for the neutral adjoint CLV via equation (5.5), and we do not care coefficients for stable adjoint CLVs too much. Hence there is no need to provide stable CLVs to our algorithm; it is even unnecessary to provide accurate non-stable CLVs, they can contain some stable components at T. Further, we care not individual CLVs but only their span. Hence we can replace $\overline{Z}$ by $\overline{W} = [\overline{w}_1, \cdots, \overline{w}_{u'}]$, with $u' \geq u + 1$, u being the number of unstable CLVs, and $\{\overline{w}_j\}_{j=1}^{u'}$ are homogeneous adjoint solutions whose non-stable components at T span the entire non-stable subspace. Such a set of solutions can be obtained by solving homogeneous adjoint equations from almost all terminal conditions of u' randomized unit vectors.

With above discussions, we see that our algorithm should solve the non-intrusive adjoint shadowing problem on one segment:

$$\begin{aligned} \min_{a \in \mathbb{R}^{u'}} \ & \frac{1}{2} \int_0^T \left\langle \overline{v}^* + \overline{W}a, \overline{v}^* + \overline{W}a \right\rangle \ , \\ \text{s.t.} \ & \int_0^T \left\langle \overline{v}^* + \overline{W}a, f \right\rangle = 0 \ . \end{aligned} \tag{5.12}$$

This is simply a least squares problem with arguments $a \in \mathbb{R}^{u'}$. Note that all adjoint solutions can be computed after making little modifications to existing adjoint solvers, and

our minimization is constrained to essentially only the unstable adjoint subspace: these are the benefits of the non-intrusive formulation. Letting $\overline{v} = \overline{v} + \overline{W}a$, we can compute sensitivity via equation (5.6) on a finite trajectory:

$$\delta\rho(\Phi) \approx S.C. = \lim_{T\to\infty} \frac{1}{T} \int_0^T \langle \overline{v}, f_\gamma \rangle + \Phi_\gamma \, dt \,. \tag{5.13}$$

Dividing trajectory into segments

An issue in numerical stability is that, as the trajectory gets longer, all adjoint solutions become dominated by the fastest growing adjoint CLV; as a result, the minimization problem in equation (5.12) becomes ill-conditioned. This issue also happened in non-intrusive shadowing [59, 60] and in the algorithm for computing CLVs [35], and here we use a similar technique to resolve it, that is, dividing the whole trajectory into multiple segments, and rescaling at interfaces.

Roughly speaking, at the end of each segment, we orthogonalize and rescale adjoint solutions $\overline{W}$ and $\overline{v}^*$ so that they are no longer dominated by the first CLV. Note here since adjoint solutions are integrated backward in time, the initial condition is at the end of a segment. Despite that now $\overline{W}$ and $\overline{v}^*$ are discontinuous across segments, we can still construct $\overline{v}$ as their linear combinations, and keep $\overline{v}$ continuous across all segments. This continuous $\overline{v}$ computed from multiple segments should be identical to that solved on one large segment containing the entire trajectory.

We first define some notations, as shown in figure 5.1. Let T be the time length of the entire trajectory, and A the total number of segments. We denote the time span of the i-th segment by $[t_i, t_{i+1}]$, where $t_0 = 0, t_A = T$. For quantities defined on a entire segment such as $\overline{W}_i, \overline{v}_i^*$, C_i, d_i and a_i, their subscripts are the same as the segment they are defined on. For quantities defined only at the interfaces between segments such as Q_i, R_i, b_i, p_i and λ_i, their subscripts are the same as the time point they are defined at. Some of the notations are used immediately below, the others will be used in section 5.3 and 5.6.

At time t_i, we perform QR factorization to $\overline{W}_i(t) = [\overline{w}_{i1}(t), \cdots \overline{w}_{iu'}(t)]$, which is a $M \times u'$ matrix whose column vectors are homogeneous adjoints on segment i. We use the Q-matrix, the matrix with orthonormal columns, as the terminal condition for $\overline{W}_{i-1}$ on segment $i-1$. More specifically,

$$\overline{W}_i(t_i) = Q_i R_i, \quad \text{and} \quad \overline{W}_{i-1}(t_i) = Q_i \,. \tag{5.14}$$

At time t_i, we also rescale $\overline{v}_i^*(t)$, which is the particular inhomogeneous adjoint solution on segment i. To do this, we subtract from $\overline{v}_i^*$ its orthogonal projection onto homogeneous adjoint solutions. More specifically,

$$p_i := \overline{v}_i^*(t_i) - Q_i b_i \,, \text{ where } b_i = Q_i^T \overline{v}_i^*(t_i) \,, \text{ and } \overline{v}_{i-1}^*(t_i) = p_i \,. \tag{5.15}$$

This rescaling maintains the continuity of the affine space $\overline{v}_i^* + span\{\overline{w}_{ij}\}_{j=1}^{u'}$ across different segments.

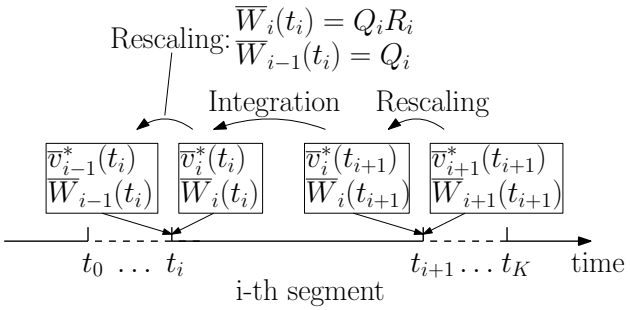


Figure 5.1: Notations for multiple segments. $\overline{W}_i(t), \overline{v}^*_i(t)$ are defined on the i-th segment, which spans $t \in [t_i, t_{i+1}]$; Q_i, R_i are defined at t_i. 'Integration' refers to integrating adjoint equations for $\overline{W}_i(t), \overline{v}^*_i(t)$: after this procedure we move from end to the start within one segment. 'Rescaling' refers to renormalize adjoint solutions at the interface between segments: after this procedure we move to another time segment.

The continuity of affine space allows us to impose continuity condition for $\overline{v}_i$, which is the adjoint shadowing direction on segment i. On each segment, $\overline{v}_i = \overline{v}^*_i + \overline{W}_i a_i$ for some $a_i \in \mathbb{R}^{u'}$. The continuity condition can now be expressed via a relation between a_i and a_{i-1}:

$$\overline{v}^*_i(t_i) + \overline{W}_i(t_i) a_i = \overline{v}^*_{i-1}(t_i) + \overline{W}_{i-1}(t_i) a_{i-1} \ . \tag{5.16}$$

Apply equation (5.14) and (5.15), cancel $\overline{v}^*_i(t_i)$ on each side, we get:

$$Q_i R_i a_i = -Q_i b_i + Q_i a_{i-1} \tag{5.17}$$

Since Q_i has orthonormal columns, $Q_i^T Q_i = I \in \mathbb{R}^{u' \times u'}$. Multiplying Q_i^T to the left of both sides, we have the continuity condition for $\overline{v}$:

$$a_{i-1} = R_i a_i + b_i \ . \tag{5.18}$$

5.3 The non-intrusive adjoint shadowing algorithm

Procedure list of the algorithm

Now we give a procedure list of the non-intrusive adjoint shadowing algorithm. To start with, we need to have an inhomogeneous adjoint solver and a homogeneous adjoint solver, both can take arbitrary terminal conditions. The inhomogeneous adjoint equation we solve in non-intrusive adjoint shadowing has right-hand-side $-\Phi_x$, which is the same as many existing adjoint solvers. Hence for inhomogeneous adjoint solvers in non-intrusive adjoint shadowing, we only need to change existing solvers to be able to take arbitrary terminal conditions. For homogeneous adjoint solvers, we only need to further change the right-hand-side of existing solvers to zero.

We provide the following data to non-intrusive adjoint shadowing: 1) the number of homogeneous adjoint solutions, $u' \geq u+1$, where u is the unstable dimension; 2) the total number of segments, A; 3) for convenience, we assume that the length of all time segments are the same, denoted by ΔT. The total time length is determined by $T = A\Delta T$. Moreover, in the procedure list below, inner products are written in matrix notations, and by default vectors are in column forms.

1. Integrate the primal system for sufficiently long time before $t = 0$ so that $u(t = 0)$ is on the attractor.

2. Compute the trajectory $u(t), t \in [0, T]$, by integrating the primal system.

3. Generate terminal conditions for $\overline{W}_i$ and $\overline{v}_i^*$ on the last segment $i = A - 1$:

   a) Randomly generate a $M \times u'$ full rank matrix, Q'. Perform QR factorization: $Q_A R_A = Q'$.

   b) Set $p_A = 0$.

4. Compute $\overline{W}_i$ and v_i^* on all segments. For $i = A - 1$ to $i = 0$ do:

   a) To get $\overline{W}_i(t)$, whose columns are homogeneous adjoint solutions on segment i, solve:
   $$\frac{d\overline{W}_i}{dt} + f_x^T \overline{W}_i = 0, \quad \overline{W}_i(t_{i+1}) = Q_{i+1}\,. \tag{5.19}$$
   To get $\overline{v}_i^*(t)$, solve the inhomogeneous adjoint equation:
   $$\frac{d\overline{v}_i^*}{dt} + f_x^T \overline{v}_i^* = -\Phi_x\,, \quad \overline{v}_i^*(t_{i+1}) = p_{i+1}\,. \tag{5.20}$$

   b) Compute the following integrations.
   $$\begin{aligned}
   C_i &= \int_{t_i}^{t_{i+1}} \overline{W}_i^T \overline{W}_i dt\,, \quad d_i^{wv^*} = \int_{t_i}^{t_{i+1}} \overline{W}_i^T \overline{v}^* dt\,,\\
   d_i^{wf} &= \int_{t_i}^{t_{i+1}} \overline{W}_i^T f dt\,, \quad d_i^{v^*f} = \int_{t_i}^{t_{i+1}} \overline{v}^{*T} f dt\,,\\
   d_i^{wf_\gamma} &= \int_{t_i}^{t_{i+1}} \overline{W}_i^T f_\gamma dt\,, \quad d_i^{v^*f_\gamma} = \int_{t_i}^{t_{i+1}} \overline{v}^{*T} f_\gamma dt\,,\\
   d_i^{\Phi_\gamma} &= \int_{t_i}^{t_{i+1}} \Phi_\gamma dt\,,
   \end{aligned} \tag{5.21}$$
   where $d_i^{wv^*}$, d_i^{wf}, $d_i^{wf_\gamma} \in \mathbb{R}^{u'}$; $d_i^{v^*f}$, $d_i^{v^*f_\gamma}$, $d_i^{\Phi_\gamma} \in \mathbb{R}$; $C_i \in \mathbb{R}^{u' \times u'}$ is the covariant matrix.

   c) Orthonormalize homogeneous adjoint solutions via QR factorization:
   $$Q_i R_i = \overline{W}_i(t_i) \tag{5.22}$$

d) Rescale the inhomogeneous adjoint solution using Q_i:

$$p_i = \overline{v}_i^*(t_i) - Q_i b_i\,, \quad \text{where } b_i = Q_i^T \overline{v}_i^*(t_i)\,. \tag{5.23}$$

5. Compute the adjoint shadowing direction $\{\overline{v}_i\}_{i=0}^{A-1}$.

a) Solve the non-intrusive adjoint shadowing problem on multiple segments:

$$\begin{aligned} &\min_{a_0,\cdots,a_{A-1}\in\mathbb{R}^{u'}} \sum_{i=0}^{A-1} \frac{1}{2}(a_i)^T C_i a_i + (d_i^{wv^*})^T a_i, \quad \text{s.t.} \\ &a)\ \ a_{i-1} = R_i a_i + b_i\,, \quad i = 1,\cdots,A-1\,, \\ &b)\ \ \sum_{i=0}^{A-1} (d_i^{wf})^T a_i + \sum_{i=0}^{A-1} d_i^{v^*f} = 0\,. \end{aligned} \tag{5.24}$$

This is a least squares problem in $\{a_i\}_{i=0}^{A-1} \subset \mathbb{R}^{u'}$. In 5.6 we suggest one way to solve this problem.

b) On each time segment i, $\overline{v}_i$ is given by

$$\overline{v}_i(t) = \overline{v}_i^*(t) + \overline{W}_i(t) a_i. \tag{5.25}$$

6. Compute the derivative by:

$$\delta\rho(\Phi) \approx \frac{1}{T}\sum_{i=0}^{A-1} \int_{t_i}^{t_{i+1}} \left(\overline{v}_i^T f_\gamma + \Phi_\gamma\right) dt = \frac{1}{T}\sum_{i=0}^{A-1} \left(d_i^{v^*f_\gamma} + a_i^T d_i^{wf_\gamma} + d_i^{\Phi_\gamma}\right) \tag{5.26}$$

Remarks about non-intrusive adjoint shadowing

Miscellaneous

We remark that if we are not interested in obtaining $\overline{v}(t)$ for all t, there is no need to store adjoint solutions $\overline{W}$ and $\overline{v}^*$ in computers, which is typically lots of data. To compute sensitivity, we only need to store $d_i^{wv^*}$, d_i^{wf}, $d_i^{wf_\gamma}$, $d_i^{v^*f}$, $d_i^{v^*f_\gamma}$, $d_i^{\Phi_\gamma}$, C_i given in equation (5.21), R_i given in equation (5.22), and b_i given in equation (5.23).

Moreover, when computing quantities in equation (5.21), we can estimate the integration using particular values of the integrands evaluated at several snapshots, to further reduce the storage management cost. For example, similar to [7], we can estimate C_i by the terminal value of W_i, which is Q_i, and thus $C_i = I \in \mathbb{R}^{u'\times u'}$. We suggest further research be done to determine which estimation is best practice.

Non-intrusive adjoint shadowing has the benefit of typical adjoint algorithms, that is, for a new parameter γ, $\overline{v}$ does not change, so we only need to give new f_γ, Φ_γ, and recompute equation (5.21) and (5.26). Hence the extra cost for a new parameter is only performing

an L^2 inner product, which is negligible in comparison with the total cost of the algorithm. More specifically, assume there are n parameters, $\gamma = [\gamma_1, \cdots, \gamma_n]$, we can define

$$\begin{aligned} \frac{d\rho(\Phi)}{d\gamma} &= [\frac{\partial \rho(\Phi)}{\partial \gamma_1}, \cdots, \frac{\partial \rho(\Phi)}{\partial \gamma_n}] \in R^{1\times n}\,; \\ \Phi_\gamma &= [\frac{\partial \Phi}{\partial \gamma_1}, \cdots, \frac{\partial \Phi}{\partial \gamma_n}] \in R^{1\times n}\,; \\ f_\gamma &= [\frac{\partial f}{\partial \gamma_1}, \cdots, \frac{\partial f}{\partial \gamma_n}] \in R^{M\times n}\,. \end{aligned} \tag{5.27}$$

With these definitions, the non-intrusive adjoint shadowing algorithm, in particular equation (5.21) and (5.26), extend to several parameters with almost no extra cost. An extreme example is where f_γ is unknown a priori and we can now use $\overline{v}$ to design an optimal control, f_γ.

The assumptions in theorem 5 are made for theoretically proving the unique existence of adjoint shadowing directions and convergence of non-intrusive adjoint shadowing. In practice, it is possible that non-intrusive tangent/adjoint shadowing are still valid on a chaotic system which fails these assumptions. For example, the 3D cylinder flow we investigate later in this chapter has at least two neutral CLVs, corresponding to translations in time and in the span-wise directions, due to the periodic boundary condition. In fact, in [57] we also showed that the smallest angle between tangent CLVs depends on meshes and may fall below a threshold value: this further violates our assumptions. However, we did found the trend that angles between tangent CLVs gets larger when their indices are further apart: this property is related to hyperbolicity, but has not been well investigated yet. As we shall see, both non-intrusive shadowing and adjoint shadowing compute correct sensitivities on this 3D flow. The generality of shadowing methods is as suggested by the chaotic hypothesis [33, 32], that is, theoretical tools may still valid for non-uniform hyperbolic chaotic systems, even though those tools can only be rigorously proved with a stricter assumption. We do not expect non-intrusive shadowing and adjoint shadowing be valid for all chaotic systems; however, they are valid somewhere beyond our current assumptions. We call for more research to identify the limit of shadowing methods, especially in real-life problems.

In 5.6, we discuss in detail non-intrusive adjoint shadowing for discrete systems, more specifically, hyperbolic diffeomorphisms. Adjoint shadowing directions for diffeomorphisms were also defined in [54]. Because of the absence of the neutral subspace, the non-intrusive adjoint shadowing algorithm for hyperbolic diffeomorphisms is easier than flows. To obtain non-intrusive adjoint shadowing for diffeomorphisms, we no longer compute d_i^{wf}, $d_i^{v^*f}$, and no longer impose the second constraint in the non-intrusive adjoint shadowing problem. Of course, we should change integrations to summations, and adjoint equations to their discrete counterparts.

Number of homogeneous adjoint solutions

Since the number of homogeneous adjoint solutions should be strictly larger than then number of unstable adjoint CLVs, which equals the number of unstable tangent CLVs, we first discuss the number of unstable CLVs, about which there are two questions: (1) whether the absolute number can be large; (2) whether the number is significantly lower than the dimension of the system. We are interested in (1) because we want to estimate the cost of non-intrusive adjoint shadowing. We are interested in (2) because we want to determine whether computational efficiency can benefit from the 'non-intrusive' formulation, which restricts minimization to unstable subspaces. Roughly, the efficiency improvement due to the non-intrusive formulation is proportional to the ratio of the system dimension to unstable subspace dimension.

First, the absolute number of unstable CLVs can be large and the cost of non-intrusive adjoint shadowing increases. We think maybe this is the price to pay for chaos, that is, for more chaotic systems, numerical methods should be more expensive, not only for non-intrusive shadowing and adjoint shadowing, but also for other common methods such as computing long-time averages, which should take longer time to converge for more chaotic systems. Second, in a recent paper [57], based on observations on flow past a 3D cylinder, we conjectured that for open flows, CLVs active in the freestream or less turbulent regions are stable. At least for these open flows, where there are large areas of freestream, a large fraction of CLVs should be stable. For these cases, we can benefit from the non-intrusive formulation by restricting minimization to unstable subspaces.

We provide some examples on the number of unstable CLVs and the dimension of the system. The number of unstable CLVs for general 3D turbulent flows is bounded by the 'necessary' degree of freedom for qualitatively describe the flow, which is estimated by $Re^{3/4}$ [45]; [23] gives a pessimistic bound of Re^3, under weaker assumptions. The point is, although the dimension of the phase space for fluid flows is infinite, there is only a finite number of unstable dimension. For numerically simulated fluids, due to numerical dissipation and turbulence models, there should be less unstable CLVs than analytic solutions. For example, for a 2D incompressible channel flow over a backward facing step at Reynolds number $Re = 2.5 \times 10^4$, there are 13 unstable CLVs in a system of dimension 4×10^4 [59]. For a 2D NACA 0012 airfoil at Mach number $Ma = 0.2$, angle of attack 20 deg, $Re = 2400$, there are less than 5 unstable CLVs for different implementations with system dimension ranging from 7×10^3 to 8×10^5 [29]. For a 3D turbulent channel flow with $Ma = 0.3$ and $Re_\tau = 180$ on a domain of size $4\pi \times 2 \times 2\pi$, there are about 1.5×10^3 unstable CLVs out of a system of dimension 2.2×10^6 [6]. For a 3D weakly turbulent flow over a cylinder at $Re = 5.2 \times 10^2$, there are 20 unstable out of 1.9×10^6 [57]. For a weather model, PUMA, there are about 65 unstable out of more than 10^5 [26]. All of the above numerical fluid problems have unstable CLVs less than 0.1% of the system dimension. On the other hand, there are cases where significant part of all CLVs are unstable, such as Hamiltonian systems. To conclude, we believe that although the cost of non-intrusive tangent/adjoint shadowing can be high, for many cases, the idea of non-intrusive formulation is still important for achieving the highest possible computational efficiency.

In our procedure list we listed u' in the setting of non-intrusive adjoint shadowing and required it strictly larger than u, the number of unstable CLVs. How should we know u before running non-intrusive adjoint shadowing? And what if we chose initial u' smaller than required? First, the number of unstable modes is roughly positively related to how chaotic the flow is. This is not a rigorous criterion but readers can look at some test cases to have a rough sense. But there is not any precise method that allows us to know the exact number at the first glance. Second, even if we started with an insufficient u', we can add adjoint solutions inductively, rather than recomputing everything all over again. More specifically, in the non-intrusive adjoint shadowing problem in equation (5.24) and the sensitivity formula in equation (5.26), say we want to add k more adjoint solutions, then coefficients arrays $d_i^{wv^*}$, d_i^{wf}, $d_i^{wf_\gamma}$ and b_i, should be augmented by k more entries, while the old coefficient arrays are not changed inside the new arrays; similarly, the coefficient matrices C_i, R_i should be augmented by k rows and k columns.

The headache of choosing an initial u' is further relieved by the fact that adjoint solutions can be more efficiently computed in batches. Within each segment, we can accelerate non-intrusive adjoint shadowing by taking advantage of the fact that all adjoint solutions, both homogeneous and inhomogeneous, use the same Jacobian f_x. If the numerical integration is vectorized, we can integrate all adjoint solutions simultaneously without repeatedly loading f_x into the computer CPU, which is the most time-consuming procedure in the numerical integration. At each time step, instead of several matrix-vector products, we can perform one matrix-matrix products, where the second matrix is composed of several adjoint solutions; then we add the right-hand-side to the inhomogeneous adjoint solution. For example, for a 4th order IEDG solver, the marginal cost for one more adjoint solution can be only, say 0.037, of the first adjoint solution. In this scenario we should start non-intrusive adjoint shadowing and then add adjoint solutions by batches on the order of $1/0.037 \approx 27$ adjoint solutions per batch. This should be further faster than adding adjoint solutions one by one. [1] However, our later implementation in this chapter does not yet have this vectorized feature.

Other settings of non-intrusive adjoint shadowing

It is required by the algorithm that we run primal system long enough before the main part of non-intrusive adjoint shadowing, so that our initial condition is on the attractor. In general, we can not know very well what is 'long enough' before we do any computations, and this run-up time is determined empirically as the time when the flow field starts to repeat itself. In a typical scenario, we would run a primal simulation before taking interest in any sensitivities. When running that primal simulation, there is the same question of when we reach the stage that enough long-time behavior has been captured: typically this is indicated by that several objective functions began to oscillate around some averaged values.

[1] The ideas of taking advantage of vectorized integration and the estimation on IEDG were both given during private discussion by Pablo Fernandez, who co-authored with us on finite-difference non-intrusive shadowing.

We should also determine the time length T on which we run non-intrusive adjoint shadowing. In practice T is determined empirically as the time when the sensitivity computed by non-intrusive adjoint shadowing converges to within the uncertainty bound we desire. However, there is one caveat that the adjoint solutions are computed backwards in time. Now if we find T insufficient, we can not add time after T without recomputing all adjoint solutions, since integration adjoint solutions forward in time is very unstable. Rather, we should add time before our current trajectory. In practice, we should run our primal simulation till enough long-time behavior has been captured, then start computing adjoint solutions from the end of that primal trajectory. We have found that typically non-intrusive adjoint shadowing requires a shorter trajectory to compute sensitivity than that required to reflect average behavior.

Then we discuss the choice of segment length ΔT. Similar to non-intrusive shadowing, ΔT is determined by that within one segment, the leading adjoint CLV does not dominate the u'-th adjoint CLV. This is because otherwise we would have covariant matrices, $\{C_i\}_{i=0}^{A-1}$, with small condition number, which would lead to eventually the poor condition of the non-intrusive adjoint shadowing problem in equation (5.24). We recommend $\Delta T(\lambda_1 - \lambda_{u'})$ to be $O(1)$, in which case within one segment, the leading CLV would grow to be about $e^1 = 2.7$ time larger than the u'-th CLV.

A related question is that if the leading LE is large and we select a small ΔT, will the cost of frequent rescaling offset the cost reduction due to non-intrusive formulation? First, as we discussed in section 5.3, there are a lot of fluid systems whose CLVs are mostly stable, in which case the non-intrusive formulation is beneficial. Second, our understanding is that the numerical methods should have smaller time steps for more chaotic systems, to capture accurate motions on all scales. Hence one segment, although is shorter in physical time, may still contain many small time steps. As a result, the rescaling may not be more frequent for more chaotic systems. Again, we call for more research on numerical schemes and LE spectrum, especially for systems other than fluid or extremely chaotic systems.

Comparison with other shadowing algorithms

There are currently several variants of non-intrusive shadowing [59], such as the finite-difference version [61] and the discrete adjoint version [7], Non-intrusive adjoint shadowing, as well as these variants, bears part of the merit of 'non-intrusive' formulation, that is, the minimization problems in these algorithms are constrained to the unstable subspaces. Hence, for many real-life problems, where the unstable subspaces have significantly lower dimension than the dynamical systems, these algorithms should be significantly faster than previous shadowing algorithms.

We compare non-intrusive adjoint shadowing with variants of non-intrusive shadowing in table 5.1. In particular, we want to compare in more detail the two adjoint algorithms: discrete adjoint versus non-intrusive adjoint shadowing. Non-intrusive adjoint shadowing should be easier to implement than the discrete adjoint version since it does not require tangent solvers, and requires less modification to existing adjoint solvers. Furthermore, unlike

discrete adjoint version, Non-intrusive adjoint shadowing does not explicitly depend on the fact that inner-products between adjoint and tangent homogeneous solutions are constants: since this property holds true only for analytic solutions but is typically false for numerical solutions, we think non-intrusive adjoint shadowing should be more robust to implementations of tangent and adjoint solvers, and should typically have better convergence. We suggest more numerical comparison be done to compare the two methods.

	NI shadowing	FD version	DA version	NI adjoint
needs prm solvers	yes	yes	yes	yes
needs tan solvers	yes	no	yes	no
needs adj solvers	no	no	yes	yes
cost increases with parameter numbers	yes	yes	no	no
cost increases with objective numbers	no	no	yes	yes
cost for 1 parameter and 1 objective	1 prm + 1 ihm tan + (u'-1) hm tan	(u'+1) prm	1 prm +1 ihm adj + (u'-1) hm tan	1 prm +1 ihm adj + u' hm adj

Table 5.1: Comparison of non-intrusive (NI) adjoint shadowing with the original version, Finite Difference (FD) version, and discrete adjoint (DA) version of non-intrusive (NI) shadowing. Here 'prm', 'tan', 'adj', 'ihm' and 'hm' are short for primal, tangent, adjoint, inhomogeneous and homogeneous, respectively. u' is a number strictly larger than the number of unstable CLVs. For item 4 and 5, we assume that all objectives and parameters are determined before the computation, rather than adding more objectives and parameters after the computation is done.

5.4 Applications

Application on Lorenz 63 system

In this subsection we apply non-intrusive adjoint shadowing to the Lorenz 63 system as an illustration. [2] Lorenz 63 is an ordinary differential equations system with three states $u = [x, y, z]$:

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho' - z) - y, \quad \frac{dz}{dt} = xy - \beta z. \tag{5.28}$$

Notice that here ρ' is one of the parameters, not to be confused with the SRB measure, ρ. We fix $\beta = 8/3$. This system models the heat transfer in a fluid layer heated from below

[2]The python code used for this section is at: https://github.com/niangxiu/nilsas.

and cooled from above. In particular, x is the convection rate, y the horizontal temperature variation, and z the vertical temperature variation. The parameters σ and ρ' are proportional to the Prandtl number and Rayleigh number. We select the instantaneous objective function as $\Phi(x) = z$, and hence our objective $\rho(\Phi)$ is the averaged vertical temperature variation.

The primal system and adjoint equations are integrated via the explicit time-stepping scheme:

$$\begin{aligned} x_{k+1} &= x_k + f(x_k)\Delta t \\ \overline{w}_k &= \overline{w}_{k+1} + f_x(x_k)^T \overline{w}_{k+1}\Delta t \\ \overline{v}^*_k &= \overline{v}^*_{k+1} + f_x(x_k)^T \overline{v}^*_{k+1}\Delta t + \Phi_x(x_k)\Delta t \end{aligned} \tag{5.29}$$

where the subscript k denotes the time step number in numerical integration. The time step size is $\Delta t = 0.001$. For non-intrusive adjoint shadowing, time segment length is $\Delta T = 0.2$, thus there are 200 time steps per segment.

We want to determine the number of unstable CLVs for the Lorenz system. The Lyapunov exponents, $\lambda_1, \lambda_2, \lambda_3$, satisfy the following constraints [10]:

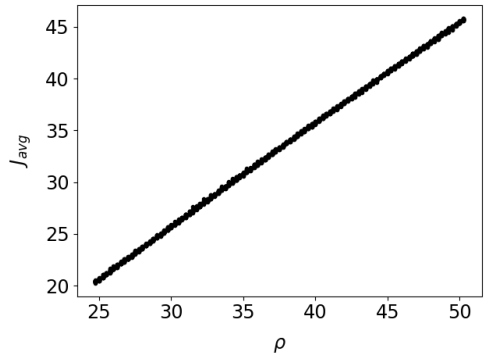
$$\lambda_1 + \lambda_2 + \lambda_3 = -(1 + \sigma + \beta) < 0 \; . \tag{5.30}$$

Moreover, one of these exponents corresponds to the neutral CLV so it is zero. Hence there is at most one positive exponent, so in non-intrusive adjoint shadowing we set the number of homogeneous solutions $u' = 2$.

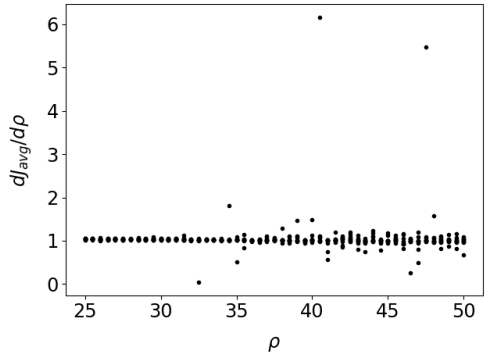
We verify that non-intrusive adjoint shadowing gives correct sensitivities by computing $\rho(\Phi)$ and $\partial\rho(\Phi)/\partial\rho'$ for different ρ', while fixing $\sigma = 10$. The Lorenz system has one quasi-hyperbolic strange attractor when $25 \leq \rho' < 31$, and one non-hyperbolic attractor when $31 \leq \rho' \leq 50$: none of these cases strictly satisfies our uniform hyperbolic assumption. As shown in figure 5.2, as ρ' becomes larger, the system becomes non-hyperbolic, and the sensitivity results given by non-intrusive adjoint shadowing begin to oscillate. Nevertheless, non-intrusive adjoint shadowing gives that $\partial\rho(\Phi)/\partial\rho'$ is approximately 1 for all ρ', which matches the trend between $\rho(\Phi)$ and ρ': this again shows that non-intrusive adjoint shadowing can be effective for systems not satisfying assumptions of theorem 5, as we discussed in section 5.3.

Then we show that both $\rho(\Phi)$ and the sensitivities computed by non-intrusive adjoint shadowing converge as the trajectory length T gets larger, while fixing $\sigma = 10$ and $\rho' = 28$. Figure 5.3 shows that the standard deviation of $\rho(\Phi)$ reduces at the rate of $T^{-0.5}$. Figure 5.4 shows that the sensitivities computed by non-intrusive adjoint shadowing, with respect to both ρ' and σ, converge faster than the rate of $T^{-0.5}$.

Non-intrusive adjoint shadowing computes sensitivities with respect to multiple parameters with almost no additional cost, since the adjoint shadowing solution $\overline{v}$ does not depend on the choice of parameters. Figure 5.5 illustrates the contour of $\rho(\Phi)$ with respect to ρ' and σ, and the gradient, $[\partial\rho(\Phi)/\partial\rho', \partial\rho(\Phi)/\partial\sigma]$, is computed by non-intrusive adjoint shadowing. Since we use the same length unit for both parameters, gradients should be perpendicular

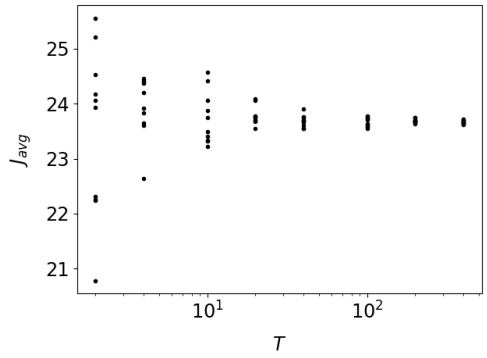


(a) For each value of ρ', $\rho(\Phi)$ is computed 20 times on randomly initialized trajectories of length 100.

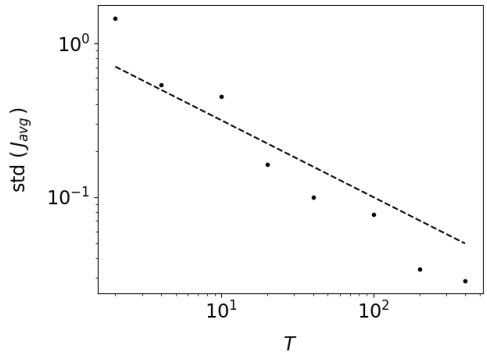


(b) For each ρ', $\partial\rho(\Phi)/\partial\rho'$ is computed 10 times by non-intrusive adjoint shadowing on randomly initialized trajectories of length 40.

Figure 5.2: $\rho(\Phi)$ and $\partial\rho(\Phi)/\partial\rho'$ versus ρ' for the Lorenz 63 system. Here $\sigma = 10$ is fixed.



(a) For each value of trajectory length T, $\rho(\Phi)$ is computed 10 times on randomly initialized trajectories.

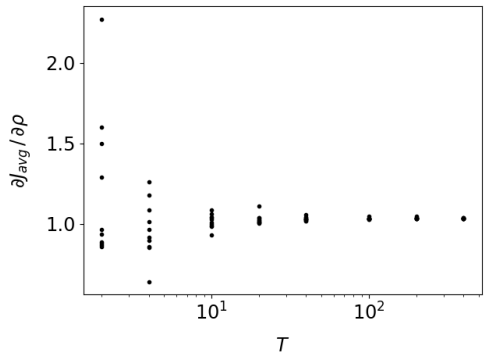


(b) The sample standard deviation of the 10 $\rho(\Phi)$'s computed at each T. The dashed line is $T^{-0.5}$.

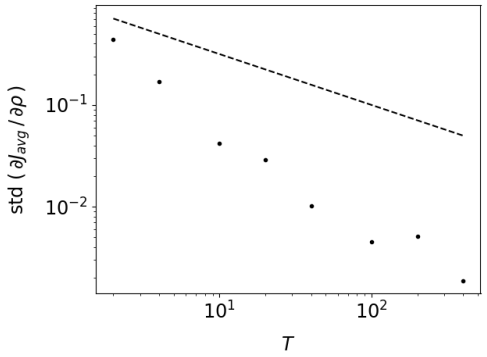
Figure 5.3: Convergence of the averaged objective $\rho(\Phi)$ with respect to the trajectory length T. Here $\rho' = 28$ and $\sigma = 10$ are fixed.

to the level sets of the objective: this is indeed the case, and it shows non-intrusive adjoint shadowing gives correct gradient information.

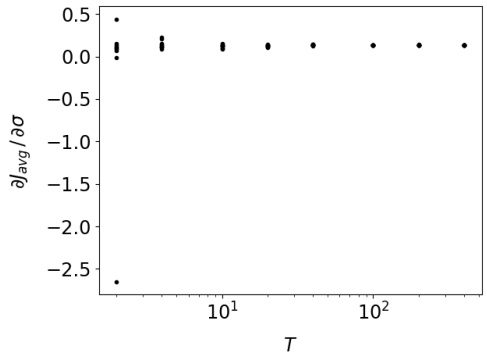
Finally, we draw the norm of an adjoint shadowing direction in figure 5.6. As we can see from the left plot, the norm of the adjoint shadowing direction does not grow exponentially, satisfying the third property of adjoint shadowing directions. Moreover, as shown in the right plot, the adjoint shadowing direction computed by non-intrusive adjoint shadowing is



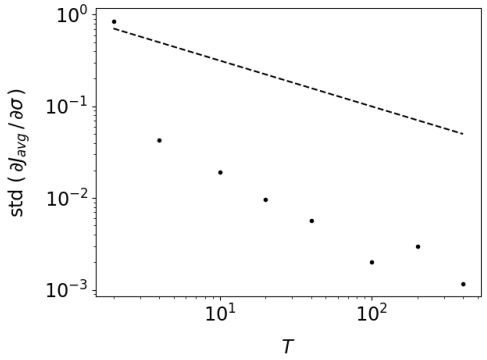
(a) For each value of T, $\partial\rho(\Phi)/\partial\rho'$ is computed by non-intrusive adjoint shadowing 10 times on randomly initialized trajectories.



(b) The sample standard deviation of the 10 $\partial\rho(\Phi)/\partial\rho'$'s computed at each T. The dashed line is $T^{-0.5}$.



(c) For each value of T, $\partial\rho(\Phi)/\partial\sigma$ is computed by non-intrusive adjoint shadowing 10 times on randomly initialized trajectories.



(d) The sample standard deviation of the 10 $\partial\rho(\Phi)/\partial\sigma$'s computed at each T. The dashed line is $T^{-0.5}$.

Figure 5.4: Convergence of sensitivities computed by non-intrusive adjoint shadowing with respect to the trajectory length T. Here $\rho' = 28$ and $\sigma = 10$ are fixed.

continuous. This shows that our dividing trajectory technique indeed allows us to recover a continuous adjoint shadowing direction.

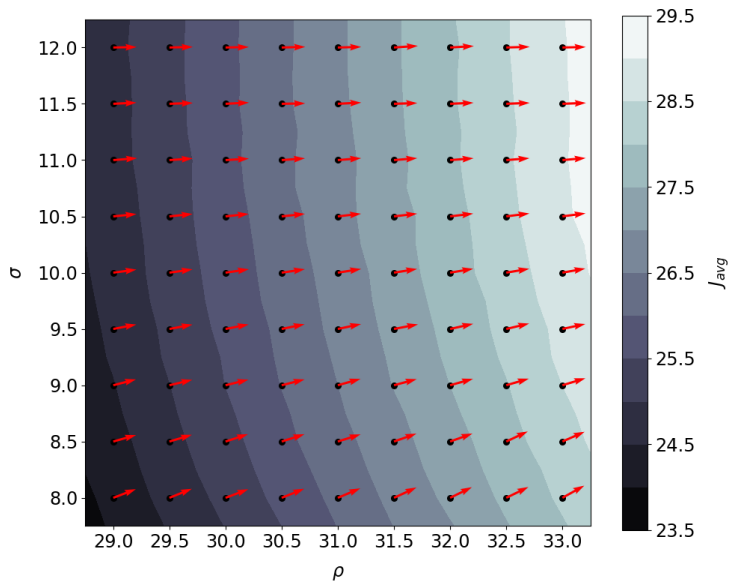


Figure 5.5: Gradients computed by non-intrusive adjoint shadowing. The contour is of $\rho(\Phi)$ with respect to ρ' and σ, and arrows are gradient vectors. Here $\rho(\Phi)$'s are averaged over 20 randomly initialized trajectories of length 100, while gradients computed by non-intrusive adjoint shadowing are averaged over 10 randomly initialized trajectories of length 40. The arrow length is 0.2 times the gradient norm. non-intrusive adjoint shadowing computes one gradient, composed of two sensitivities to two parameters, in one run.

Application on a turbulent flow past a three-dimensional cylinder

In this subsection, we apply non-intrusive adjoint shadowing to a 3D subsonic flow over a cylinder at Reynolds number $Re = 1100$ and Mach number $Ma = 0.093$. [3] The flow-wise length of the domain is $60d$, where $d = 0.25mm$ is the diameter of the cylinder. The Reynolds number is defined using the diameter of the cylinder and the density, velocity and viscosity of inflow. The span-wise extent, at $z = 2d$, is sufficient to capture most of the important flow features, like a turbulent wake and flow separation. The front view of our fluid problem is shown in figure 5.7.

We use compressible Navier-Stokes equations with the ideal gas law approximating the thermodynamic state equation [34]. The gas is assumed to be air. More specifically, the

[3]The flow solver, adFVM, used for this section is at: https://github.com/chaitan3/adFVM, the particular file that implements the non-intrusive adjoint shadowing algorithm used in this case is apps/nilsas.py.

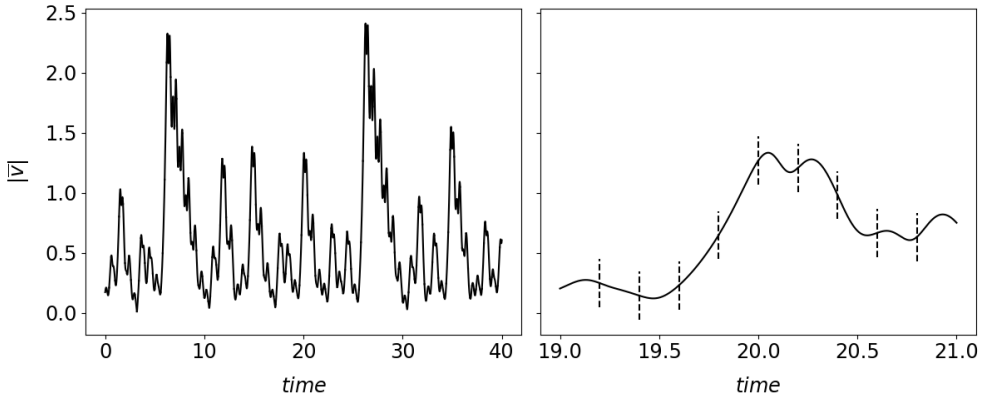


Figure 5.6: Norm of the adjoint shadowing direction computed by non-intrusive adjoint shadowing for the Lorenz system, with $\rho' = 28$ and $\sigma = 10$. Left: plot on the entire trajectory time span. Right: zoom onto time span from 19 to 21. The vertical dashed lines marks different time segments.

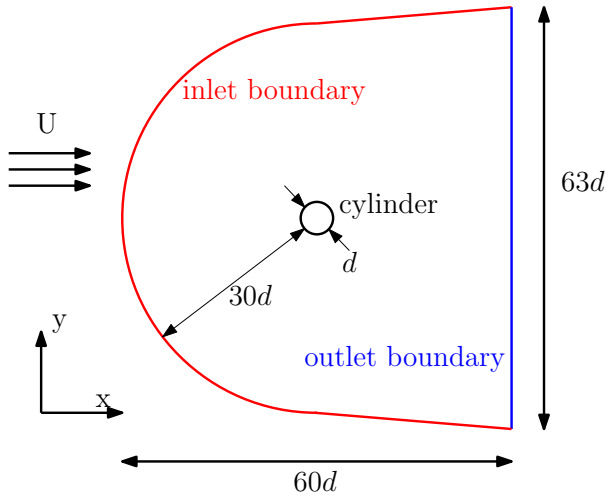


Figure 5.7: Geometry used in the simulation of a 3D flow past a cylinder. The span-wise extent of the computational domain is 2d.

governing equations are:

$$
\begin{aligned}
&\frac{\partial \rho'}{\partial t} + \nabla \cdot (\rho' \mathbf{u}) = 0\,, \\
&\frac{\partial (\rho' \mathbf{u})}{\partial t} + \nabla \cdot (\rho' \mathbf{u}\mathbf{u}) + \nabla p = \nabla \cdot \sigma\,, \\
&\frac{\partial (\rho' E)}{\partial t} + \nabla \cdot (\rho' E \mathbf{u} + p\mathbf{u}) = \nabla \cdot (\mathbf{u} \cdot \sigma + \alpha \rho' \gamma \nabla e)\,, \\
&\sigma = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) - \frac{2\mu}{3}(\nabla \cdot \mathbf{u})\mathbf{I}\,, \quad c = \sqrt{\frac{\gamma p}{\rho'}}\,, \\
&p = (\gamma - 1)\rho' e\,, \quad e = E - \frac{\mathbf{u} \cdot \mathbf{u}}{2}\,.
\end{aligned}
\tag{5.31}
$$

Here ρ' is the density, $\mathbf{u}$ is the velocity vector, $\rho' E$ is the total energy, p is pressure, e is internal energy of the fluid, c is the speed of sound, $\gamma = 1.4$ is the isentropic expansion factor and μ is the viscosity field modeled using Sutherland's law for air

$$\mu = \frac{C_s T^{3/2}}{T + T_s} \tag{5.32}$$

where $T_s = 110.4\,A$ and $C_s = 1.458 \times 10^{-6} kg/m\,s\,\sqrt{A}$. α is the thermal diffusivity modeled using

$$\alpha = \frac{\mu}{\rho' Pr} \tag{5.33}$$

where $Pr = 0.71$ is the Prandtl number.

We use an unstructured hexahedral mesh with approximately 7×10^5 cells, with 50 cells in the span-wise direction. The front view of our mesh is shown in figure 5.8.

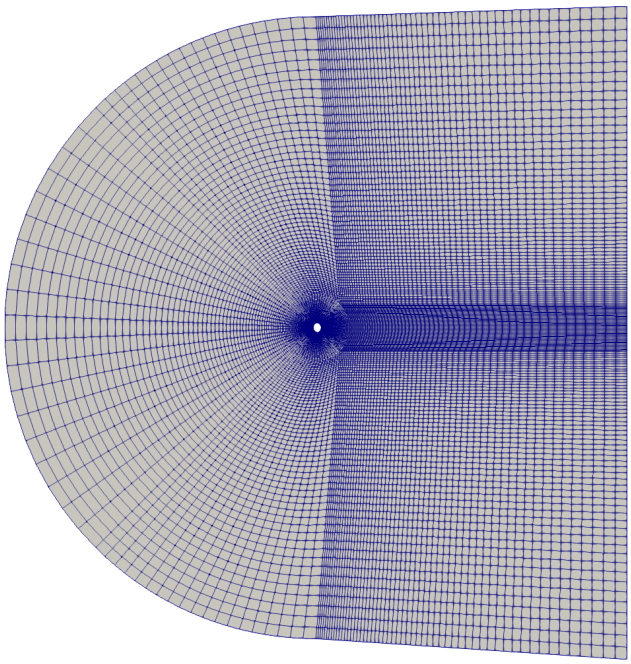


Figure 5.8: Front view of the mesh for the flow over cylinder problem. This is an unstructured hexahedral mesh with approximately 7×10^5 cells, with 50 cells in the span-wise direction.

We use a second order finite volume method (FVM) [48] for unstructured hexahedral meshes. The central differencing scheme is used to interpolate cell averages of the flow solution onto faces of the mesh [81]. The numerical fluxes for the conservative flow variables are computed using the Roe approximate Riemann solver [66]. An explicit time integration scheme, the strong stability preserving third order Runge-Kutta method [50], is used for

time marching the numerical flow solution. The size of the time step is determined using the acoustic Courant-Friedrichs-Lewy (CFL) condition [25], and we choose our CFL number to be 1.2. The flow solver is implemented in Python using the adFVM [78] library, which provides a high-level abstract application programming interface for writing efficient CFD applications. The flow solver is parallelized using the Message Passing Interface (MPI) library.

We use implicit Large Eddy Simulations (LES) in our numerical simulation. In an LES, the large scale eddies of the flow are resolved by the grid, while the contribution from the small scale eddies to the filtered Navier-Stokes equations are modeled using a sub-grid scale Reynolds stress model [34]. In this chapter, the numerical error of the discretization scheme serves as the LES model. It has been shown that when using a relatively dissipative discretization method, the numerical viscosity from the grid can be of the same order of magnitude as the sub-grid scale viscosity [52, 30], and thus can be regarded as an implicit LES model.

On the inlet boundary, we specify stagnation pressure and temperature, corresponding to a fixed Reynolds number $Re = 1100$ and a Mach number which we choose to be the system parameter. For the base case, we choose Mach number $Ma = 0.093$. Periodic boundary condition is used in the span-wise direction. The surface of the cylinder is maintained at a constant temperature of $300K$. Static pressure of 1 atmosphere unit is prescribed on the outlet boundary.

A snapshot of the flow field simulated with above settings is shown in figure 5.9. As we can see, this flow exhibits weak turbulence in the wake. In particular, the top view shows that this flow is 3D.

We choose our system parameter as the Mach number of the incoming flow. The objective function is the time-averaged normalized drag over the cylinder. More specifically,

$$\rho(\Phi) = \frac{2}{\rho_r' u_r^2 z d} \lim_{T\to\infty} \int_0^T \int (p n_x - \mu \boldsymbol{n} \cdot \nabla u_x)\, dS\, dt, . \tag{5.34}$$

Here the second integral is over the surface of the cylinder; $u_r = 31.4 m/s$ and $\rho_r' = 1.3 kg/m^3$ are the reference velocity and density of the base case, where $Ma = 0.093$. For the base case, the normalized drag and the drag coefficient are the same, whereas for other Mach numbers they are different.

We run the flow simulation for 10^6 time steps, which corresponds to approximately $720 t_r$. Here the time unit t_r is the amount of time that the flow takes to traverse the length of the cylinder, that is, $t_r = d/u_r$. This time interval is sufficient to obtain a statistically converged estimate of the design objective. The standard deviation of the time-averaged objective is computed using the autoregressive time series analysis techniques described in [79] and [62], and we use one standard deviation as the confidence interval. The normalized drag for the base case is 1.2 ± 0.03. Our results reasonably matches the results from experiments [83, 67, 80], which is approximately 1.0 ± 0.15. Figure 5.10 shows different objectives for different incoming Mach number.

We first estimate the sensitivity by the linear least-squares regression method using 5 data points with different parameter values. To use this linear regression method, we need

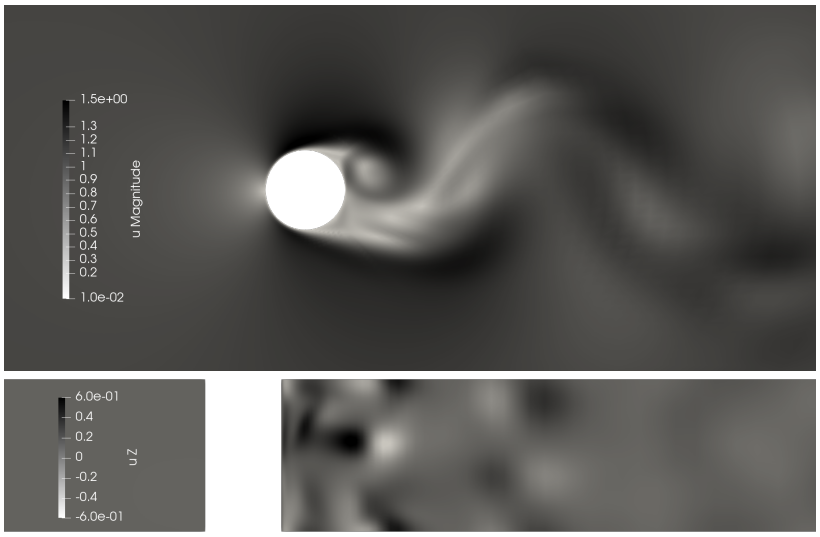


Figure 5.9: Instantaneous visualization of the flow field. Top: vertical cross-section, plotted by the magnitude of velocity. Bottom: horizontal cross-section, plotted by the span-wise velocity. The bottom picture shows the flow is 3D. All velocities are normalized by the reference velocity u_r.

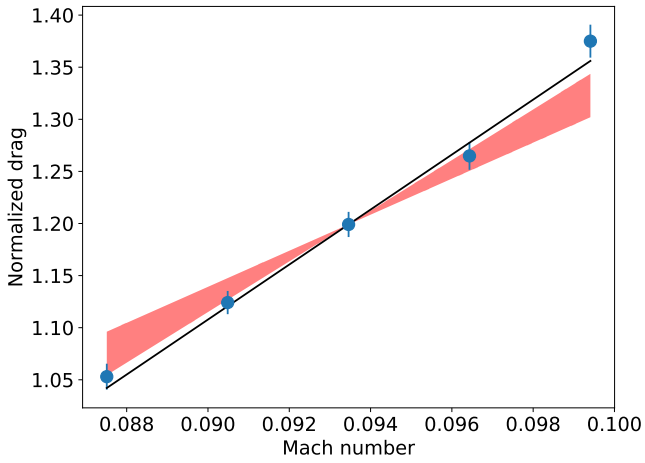


Figure 5.10: Normalized drag as a function of inlet Mach number. Blue bars denote the confidence interval of the averaged normalized drag. The black line denotes the sensitivity estimated using linear regression. The red shaded region denotes the confidence interval of the sensitivity estimated using non-intrusive adjoint shadowing.

to make the assumption that the relation between parameters and objectives is linear. We select one standard deviation of the relevant estimator as our confidence interval. Note that one shortcoming of the linear regression method is that the linear assumption may not be true when parameters are spaced far apart such that a linear approximation no longer holds on the dataset; on the other hand, when parameters are too close, the uncertainty in the objectives will lead to large error in the sensitivity. In the base case, the sensitivity of drag with respect to the inflow Mach number, given by linear regression, is 25.0 ± 2.1. This sensitivity is visualized in figure 5.10.

Adjoint LEs are shown in figure 5.11, where the confidence interval is also selected as one standard deviation given by autoregressive time series analysis. The subsonic flow over a 3D cylinder has $u = 9$ unstable adjoint CLVs. In non-intrusive adjoint shadowing, the number of homogeneous adjoint solutions computed is set to $u' = 20$. We suggest more research be done to investigate how the number of unstable CLVs grow as the flow becomes more turbulent, not only for this particular open flow problem, but for wall-bounded flows as well.

The number of segments in non-intrusive adjoint shadowing is $A = 100$ and the number of time steps per segment is 500. Each segment roughly corresponds to $0.4t_r$. Consequently, the time length of trajectory used in non-intrusive adjoint shadowing is $40t_r$, which is much lower than that required to obtain a reasonably accurate sensitivity using the linear regression method. In this particular implementation, corresponding to the discussion in section 5.3, on segment $[t_i, t_{i+1}]$, we approximate integrations in equation (5.21) using snapshots at t_{i+1}. As a result, we have $C_i = I$, $d_i^{wv^*} = 0$: this approximation eases the implementation responsible for storing adjoint solutions.

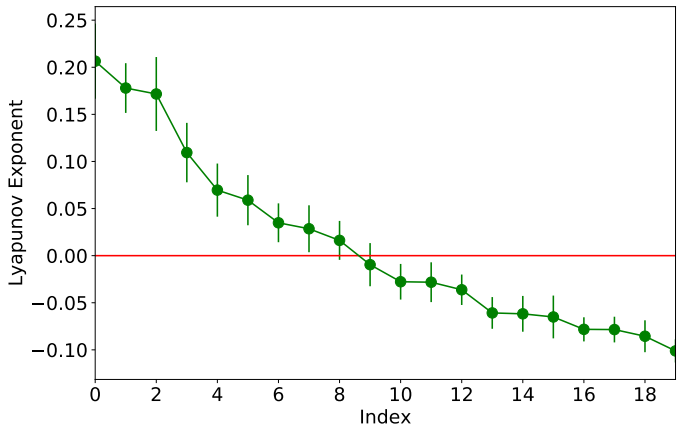


Figure 5.11: Spectrum of the first 20 adjoint Lyapunov Exponents (LE). The time unit for LEs is t_r^{-1}. The largest LE is $0.21t_r^{-1}$, meaning in one time unit t_r, the norm of the first adjoint CLV becomes $e^{0.21} = 1.23$ times larger.

The sensitivity computed by non-intrusive adjoint shadowing is 20.8 ± 3.5, which costs

about a week on a computer with 64 cores, is visualized in figure 5.10. Here the confidence interval is also selected as one standard deviation given by autoregressive time series analysis. Comparing to the sensitivity estimated via linear regression methods, the relative difference is less than 20%. As we can see, the sensitivity computed by non-intrusive adjoint shadowing correctly reflects the trend between parameters and objectives.

We remark that non-intrusive adjoint shadowing may work for systems do not strictly satisfy the assumptions in theorem 5, and our fluid problem is such an example. First, our system has at least two neutral CLVs: the first one corresponds to the common time translation of continuous dynamical systems, and the second corresponds to span-wise translations due to the periodic boundary conditions. Second, due to the similarity of this fluid problem with the one investigated in [57], whose tangent CLVs appear to have occasional tangencies, it is reasonable to assume that adjoint CLVs in our current system may also have occasional tangencies. Still, like the finite-difference shadowing in [57], non-intrusive adjoint shadowing computes a correct sensitivity: this encourages us to test non-intrusive tangent/adjoint shadowing on more general chaotic systems.

We compare computational cost for sensitivity analysis via the linear regression method and non-intrusive adjoint shadowing. The linear regression method runs the primal solver for a total of 5×10^6 steps. Non-intrusive adjoint shadowing runs the primal solver and 21 adjoint solvers, each for 5×10^4 steps, which leads to 1.1×10^6 steps in total. To build more favor towards non-intrusive adjoint shadowing, note that, first, adjoint solvers can be further accelerated due to the vectorization we discussed in section 5.3; second, non-intrusive adjoint shadowing has no additional cost for sensitivities to multiple parameters. For chaotic problems with a higher number of positive LEs, the cost of non-intrusive adjoint shadowing increases; however, if the percentage of positive LEs is still low, the non-intrusive formulation can still be a key technique for designing fast sensitivity algorithms.

5.5 Conclusions

To compute the gradient of long-time averaged objectives in chaotic systems, we develop the non-intrusive adjoint shadowing algorithm, which approximates the adjoint shadowing direction by a 'non-intrusive' formulation, which is a least squares problem in the unstable adjoint subspace. Non-intrusive adjoint shadowing is demonstrated on the Lorenz 63 system and a turbulent 3D flow over a cylinder, where it gives pretty accurate sensitivities for both cases.

Similar to non-intrusive shadowing [59], non-intrusive adjoint shadowing can be implemented with little modification to existing adjoint solvers, and its minimization is carried out only in the unstable adjoint subspace. Unlike the original tangent version, non-intrusive adjoint shadowing has the benefit of adjoint approaches that its cost does not increase with the number of parameters; thus making non-intrusive adjoint shadowing ideal for applications where there are many parameters, or where f_γ is unknown a priori. non-intrusive adjoint shadowing does not require tangent solvers, and is easy to implement.

5.6 Appendix

Solving the non-intrusive adjoint shadowing problem

We discuss one way to solve the non-intrusive adjoint shadowing problem in equation (5.24). The corresponding Lagrange function is:

$$\begin{aligned} &\sum_{i=0}^{A-1} \frac{1}{2}(a_i)^T C_i a_i + (d_i^{wv^*})^T a_i \\ &+ \sum_{i=1}^{A-1} \lambda_i^T \left(a_{i-1} - R_i a_i - b_i\right) + \lambda' \left(\sum_{i=0}^{A-1} (d_i^{wf})^T a_i + \sum_{i=0}^{A-1} d_i^{v^* f} \right) , \end{aligned} \tag{5.35}$$

where λ_i is the Lagrange multiplier for the continuity condition at t_i. By the Lagrange multiplier method, the minimizer for the non-intrusive adjoint shadowing problem is at the solution of the following linear equation systems:

$$\begin{bmatrix} C & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} a \\ \lambda \end{bmatrix} = \begin{bmatrix} -d \\ b \end{bmatrix} , \tag{5.36}$$

where the block matrices $C \in \mathbb{R}^{uA \times uA}$, $B \in \mathbb{R}^{(uA-u+1)\times uA}$, vectors $a, d \in \mathbb{R}^{uA}$ and $\lambda, b \in \mathbb{R}^{uA-u+1}$. More specifically,

$$\begin{aligned} C &= \begin{bmatrix} C_0 & & & \\ & C_1 & & \\ & & \ddots & \\ & & & C_{A-1} \end{bmatrix}, \quad B = \begin{bmatrix} I & -R_1 & & & \\ & I & -R_2 & & \\ & & \ddots & \ddots & \\ & & & I & -R_{A-1} \\ (d_0^{wf})^T & & \cdots & & (d_{A-1}^{wf})^T \end{bmatrix}, \\ a &= \begin{bmatrix} a_0 \\ \vdots \\ a_{A-1} \end{bmatrix}, \quad \lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_{A-1} \\ \lambda' \end{bmatrix}, \quad d = \begin{bmatrix} d_0^{wv^*} \\ \vdots \\ d_{A-1}^{wv^*} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_{A-1} \\ -\sum_{i=0}^{A-1} d_i^{v^* f} \end{bmatrix}, \end{aligned} \tag{5.37}$$

where $\{C_i\}_{i=0}^{A-1}$, $\{R_i\}_{i=1}^{A-1} \subset \mathbb{R}^{u\times u}$; $\{a_i\}_{i=0}^{A-1}$, $\{d_i^{wf}\}_{i=0}^{A-1}$, $\{d_i^{wv^*}\}_{i=0}^{A-1}$, $\{\lambda_i\}_{i=1}^{A-1}$, $\{b_i\}_{i=1}^{A-1} \subset \mathbb{R}^u$; λ', $\{d_i^{v^* f}\}_{i=0}^{A-1} \subset \mathbb{R}$.

We can solve the Schur complement of equation (5.36) for λ:

$$-BC^{-1}B^T\lambda = BC^{-1}d + b \, , \tag{5.38}$$

where C^{-1} can be computed via inverting each diagonal block in C. Then we can compute a by:

$$a = -C^{-1}(B^T\lambda + d) \, . \tag{5.39}$$

Non-intrusive adjoint shadowing on discrete systems

Backgrounds and notations

We provide a brief introduction on discrete dynamical systems, in particular, hyperbolic diffeomorphisms. More details are provided in [54]. First, the governing equation for a discrete dynamical system is:

$$x_{l+1} = f(x_l, s), \quad l \geq 0\,. \tag{5.40}$$

The objective is:

$$\rho(\Phi) := \lim_{N\to\infty} \frac{1}{N} \sum_{l=0}^{N-1} \Phi(x_l, \gamma). \tag{5.41}$$

Similar to flows, we assume $x_l \in \mathbb{R}^m$, and $f(x,\gamma)$ and $\Phi(x,\gamma)$ are smooth. We also assume f is a diffeomorphism in x, that is, for each fixed γ, f has a smooth inverse. Also, for simplicity of notations, we assume there is only one parameter $\gamma \in \mathbb{R}$.

We first look at the tangent equations. The homogeneous tangent diffeomorphism is:

$$w_{l+1} = f_{xl} w_l\,. \tag{5.42}$$

where the second subscript of f_{xl} indicate where the partial derivative is evaluated, that is, $f_{xl} := \partial f/\partial x(x_l, \gamma)$. A tangent CLV with exponent λ is a homogeneous tangent solution $\{\zeta_l\}_{i=0}^{\infty}$ such that there is constant C, for any integer l_1, l_2, $\|\zeta_{l_2}\| \leq Ce^{\lambda(l_2-l_1)}\|\zeta_{l_1}\|$. The uniform hyperbolicity for diffeomorphisms is defined as that all LEs are not 1.

On the adjoint side, the homogeneous adjoint diffeomorphism is defined as:

$$\overline{w}_l = f_{xl}^T \overline{w}_{l+1}\,, \tag{5.43}$$

where $\cdot^T$ is the matrix transpose. The particular inhomogeneous adjoint diffeomorphism we will be using is:

$$\overline{v}_l = f_{xl}^T \overline{v}_{l+1} + \Phi_{xl}. \tag{5.44}$$

On a trajectory $\{x_l\}_{l=0}^{\infty}$ on the attractor, the adjoint shadowing direction $\{\overline{v}_l\}_{l=0}^{\infty}$ is a sequence with the following properties:

1. $\{\overline{v}_l\}_{l=0}^{\infty}$ solves an inhomogeneous adjoint equation:

$$\overline{v}_l = f_{xl}^T \overline{v}_{l+1} + \Phi_{xl}\,, \tag{5.45}$$

2. $\overline{v}_0$ has zero component in the unstable adjoint subspace.

3. $\|\overline{v}_l\|$ is bounded by a constant for all $l \geq 0$.

It was proved in [54] that for a uniform hyperbolic diffeomorphism with a global compact attractor, on a trajectory on the attractor, there exists a unique adjoint shadowing direction. Further, we have the adjoint formula for the shadowing contribution, which is an approximation of the entire linear response:

$$\delta\rho(\Phi) \approx S.C. := \lim_{N\to\infty} \frac{1}{N} \sum_{l=0}^{N-1} \left(\langle \overline{v}_{l+1}, f_{\gamma l}\rangle + \Phi_{\gamma l}\right)\,. \tag{5.46}$$

Procedure list of non-intrusive adjoint shadowing

We provide a procedure list for the non-intrusive adjoint shadowing algorithm on discrete chaotic systems, more specifically, hyperbolic diffeomorphisms. To start with, we need an inhomogeneous adjoint solver and a homogeneous adjoint solver, both can take arbitrary terminal conditions. The inhomogeneous adjoint equation still has right-hand-side $-\Phi_x$, same as many existing adjoint solvers for discrete systems. We provide the following data: 1) the number of unstable adjoint CLVs, u, note that because the lack of neutral CLV, we can use one less homogeneous adjoint solution; 2) the total number of segments, A; 3) number of steps in one segment, L.

We can have three subscripts, the first, typically being x or γ, indicates this term is a partial derivative; the second, typically being i, 0, or A, indicates the segment number; the third, typically being l, 0, or L, indicates the step number inside a segment. Disappearance of the first subscript means that term is not a partial derivative. Disappearance of the third subscript means either we are considering all steps in a segment, or that term is defined only once per segment interface.

1. Integrate the primal system for sufficiently many steps so that the initial condition, x_{00}, is on the attractor.

2. Compute the trajectory x_{il} for $0 \le i \le A-1$ and $0 \le l \le L$. Here we assume the step at end of each segment overlaps with the start of next segment, that is, $x_{iL} = x_{i+1,0}$.

3. Generate terminal conditions for $\overline{W}_i$ and $\overline{v}_i^*$ on the last segment $i = A-1$:

   a) Randomly generate a $M \times u$ full rank matrix, Q'. Perform QR factorization: $Q_A R_A = Q'$.

   b) Set $p_A = 0$.

4. Compute $\overline{W}_i$ and v_i^* on all segments. For $i = A-1$ to $i = 0$ do:

   a) To get $\overline{W}_{il}$, whose columns are homogeneous adjoint solutions on segment i, solve:

   $$\overline{w}_{il} = f_{xil}^T \overline{w}_{i,l+1}\,, \quad \overline{W}_{iL} = Q_{i+1}\,. \tag{5.47}$$

   To get $\overline{v}_i^*(t)$, solve the inhomogeneous adjoint equation:

   $$\overline{v}_{il} = f_{xil}^T \overline{v}_{i,l+1} + \Phi_{xil}\,, \quad \overline{v}_{iL}^* = p_{i+1}\,. \tag{5.48}$$

   b) Compute the following integrations.

   $$\begin{aligned} &C_i = \sum_{l=1}^{L} \overline{W}_{il}^T \overline{W}_{il} dt\,, \quad d_i^{wv^*} = \sum_{l=1}^{L} \overline{W}_{il}^T \overline{v}_{il}^* dt\,, \quad d_i^{\Phi_\gamma} = \sum_{l=1}^{L} \Phi_{\gamma il} dt\,, \\ &d_i^{wf_\gamma} = \sum_{l=1}^{L} \overline{W}_{il}^T f_{\gamma i,l-1} dt\,, \quad d_i^{v^* f_\gamma} = \sum_{l=1}^{L} \overline{v}_{il}^{*T} f_{\gamma i,l-1} dt\,, \end{aligned} \tag{5.49}$$

where $d_i^{wv^*}$, $d_i^{wf_\gamma} \in \mathbb{R}^u$; $d_i^{v^*f_\gamma}$, $d_i^{\Phi_\gamma} \in \mathbb{R}$; $C_i \in \mathbb{R}^{u\times u}$ is the covariant matrix. Note that when multiplying adjoint solutions with f_γ, their time steps are not the same: this asymmetry is the same as that in equation (5.46). We are not sure yet if this technical detail can be neglected in practice.

c) Orthonormalize homogeneous adjoint solutions via QR factorization:

$$Q_i R_i = \overline{W}_{i0} \tag{5.50}$$

d) Rescale the inhomogeneous adjoint solution using Q_i:

$$p_i = \overline{v}^*_{i0} - Q_i b_i\,, \quad \text{where } b_i = Q_i^T \overline{v}^*_{i0}\,. \tag{5.51}$$

5. Compute the adjoint shadowing direction $\{\overline{v}_{il}\}$ for $0 \le i \le A-1$ and $0 \le l \le L$.

a) Solve the non-intrusive adjoint shadowing problem on multiple segments:

$$\begin{aligned} \min_{a_0,\cdots,a_{A-1}\in\mathbb{R}^u} \sum_{i=0}^{A-1} \frac{1}{2}(a_i)^T C_i a_i + (d_i^{wv^*})^T a_i, \quad \text{s.t.} \\ a_{i-1} = R_i a_i + b_i\,, \quad i = 1,\cdots,A-1\,. \end{aligned} \tag{5.52}$$

This is a least squares problem in $\{a_i\}_{i=0}^{A-1} \subset \mathbb{R}^u$. Note we do not have the other constraint as non-intrusive adjoint shadowing in the continuous case.

b) On each time segment i, $\overline{v}_{il}$ is given by

$$\overline{v}_{il} = \overline{v}^*_{il} + \overline{W}_{il} a_i. \tag{5.53}$$

6. Compute the shadowing contribution of the linear response by:

$$\delta\rho(\Phi) \approx S.C. \approx \frac{1}{AL} \sum_{i=0}^{A-1} \left(d_i^{v^*f_\gamma} + a_i^T d_i^{wf_\gamma} + d_i^{\Phi_\gamma} \right) \tag{5.54}$$

Chapter 6

Fast linear response algorithm

A precise and computable formula for the unstable divergence was missing in previous work. As a result, an accurate exact and affordable algorithm for the linear response for SRB measures, especially when the attractor is fractal, has been an open problem. Our definition of 'computability' is to achieve a formula with a small integrand, and whose terms are all functions rather than distributions (our definition of 'computability' is different from some other contexts). For chaotic systems, it is almost impossible to use integration-by-parts to compute the unstable divergence, because that is where we came from, the original linear response formula, which has a very large integrand growing exponentially fast to some step number W. This is quite often unaffordable, since averaging a large integrand to get a small integration result is very expensive. Moreover, in general, on fractal attractors, the unstable divergence can not be computed from summing individual directional derivatives, which are only distributions, and their values can be infinite. This chapter answers the open problem by the following results.

The first result of this chapter is the first computable formula of the unstable divergence on fractal attractors, We follow Ruelle's roadmap originally intended for proving the regularity of the unstable divergence, which includes three steps [70]. Roughly speaking, they are:

- First, transform the unstable divergence under the conditional SRB measure to the divergence under the Lebesgue measure. The difference between the two divergence can be expanded by differentiating the conditional SRB measure, which is the infinite pushforward of the Lebesgue measure.
- Then, transform the Lebesgue unstable divergence to the derivative of the ratio between two volumes, one given by the projection along stable manifolds, the other by projection along X.
- The projection along stable manifolds is still rough, so we replace it using the fact that it collapses after infinite pushforward. Hence, we can expand the volume ratio. Together with the previous step, we have expanded the Lebesgue unstable divergence.

We make Ruelle's (sketchy) roadmap coordinate-free, and derive detailed formulas. This process uses somewhat heavy differential geometry, and the expansion formula is given in theorem 7. Directly computing this formula already gives an exact algorithm, which is typically much faster than algorithms based on the original linear response formula. However, computing the expansion formula still requires solving at least u^2 many second-order tangent equations, and also computing oblique projections. Here u is the dimension of the unstable subspace. It turns out that we can get rid of oblique projection while computing only u many second-order tangent solutions.

The second result is a new characterization of the expansion formula using second-order tangent equations. This theorem is proved in section 6.2.

Theorem 6 ('fast' characterization)**.** *Make same assumptions as theorem 1. Let* $\{x_n := f^n x_0\}_{n\geq 0}$ *be a trajectory on the attractor, for any* $r_0 \in \mathcal{D}_e(x_0)$*, define the sequence* $\{r_n\}_{n\geq 0}$*,*

$$r_n \in \mathcal{D}_e(x_n), \quad r_{n+1} := P^{\perp}\tilde{\beta} r_n.$$

Then for almost all x_0 *according to the SRB measure, the unstable contribution,*

$$U.C.^W = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \left\langle \tilde{\beta} r_n, \tilde{e}_{n+1} \right\rangle.$$

Here theorem 1 is the original linear response result by Ruelle, $\langle \cdot, \cdot \rangle$ is the Riemannian metric. Roughly speaking, e is a u-dimensional cube spanned by unstable vectors, $\mathcal{D}_e$ is the space of derivatives of cubes (see Appendix 6.5); $\tilde{\beta}$ is the renormalized second-order tangent equation, which describes the propagation of derivatives of cubes (definition 4 in section 6.2); $P^{\perp}$ is the orthogonal projection operator on $\mathcal{D}_e$, projecting to the orthogonal complement of unstable subspace Appendix 6.5). Here all oblique projections are summarized into a modified shadowing direction, which is the second direction of differentiation in $\tilde{\beta}$. We have thus gotten rid of oblique projections, since shadowing directions can be expressed without them. This characterization is of theoretical interest, since it is a new form of the linear response formula, combining the advantages of both the original formula and the integrated-by-parts formula: it does not explicitly involve the oblique projection operators, and it has a small integrand. It could perhaps help generalizing linear response to less hyperbolic situations, where the oblique projection is problematic.

More importantly, this 'fast' characterization is of applied interest, since it allows the unstable contribution, hence the entire linear response, be computed even more efficiently than directly using the expansion formula. The adjective 'fast' has a specific meaning: famous fast algorithms, such as the fast Fourier transformation [24] and the fast multipole method [36, 37], found a non-obvious 'fast' structure, which allows us to combine many small terms into a few big terms, and because the rules of propagation on these small terms are similar, we can then apply some averaged rule of propagation on big terms only once a few times, instead of many times on each small term. In our case, we have found a 'embarrassingly fast' structure, where all the small terms are combined into only one summation using linearity

of derivatives of vectors, and we only need one uniform rule of propagation, given by the second-order tangent equations. We have thus achieved even better performance than directly using the expansion formula: computing the new characterization requires solving only u many second-order tangent equations.

The third result is an accurate exact, efficient, robust, and easy-to-implement algorithm for the linear response of SRB measures, which is called the fast linear response algorithm. We first show that the renormalization in $\tilde{\beta}$ only needs to be done after a number of steps. We further write everything in matrix notation, which is more suitable for programming, and should be understandable without differential geometry knowledge. Readers mainly interested in applications can jump to the detailed procedure list in section 6.3. The algorithm is exact; it is robust, since it does not involve oblique projections; it requires little additional coding to first-order and second-order tangent solvers; more importantly, it is very efficient.

The last result of this chapter is several basic geometry and algebra tools for second-order tangent equations, which governs the propagation of derivatives of unstable u-vectors. These tools are useful when considering u-dimensional invariant submanifolds in M-dimensional manifolds, for any $1 \leq u \leq M$. In particular, appendix 6.5 defines the derivative of the pushforward operator, ∇f_*. Appendix 6.5 defines the linear space, $\mathcal{D}_e$, spanned by derivatives of unstable u-vectors. Appendix 6.5 extends the projection operators on single-vectors to derivatives of u-vectors.

This chapter is organized as follows. Section 6.1 derives the expansion formula of the unstable divergence. Section 6.2 derives the fast characterization. Section 6.3 gives more details of the fast linear response algorithm, with a procedure list. Section 6.4 shows a numerical application on a modified solenoid map, which is a three-dimensional system with a two-dimensional unstable subspace, whose direction is unknown beforehand. Such an example is difficult for previous algorithms, but our algorithm is even faster than finite difference, which is typically regarded as an approximate algorithm.

6.1 Expanding unstable divergence

We first write out a computable expansion formula for the unstable divergence, following the (sketchy) roadmap given by Ruelle for proving its regularity. Here computable means that all terms are functions rather than distributions. Since the stable and unstable foliations are not differentiable, the unstable divergence, defined via directional derivatives, is *a priori* only a distribution but not a function. This difficulty is resolved via three steps of transformations. First, we expand the difference between the unstable divergence under the SRB measure and the Lebesgue measure. Then, we transform the Lebesgue unstable divergence to a ratio between volumes given by two projections, one of which is still non-differentiable. Finally, we express the non-differentiable projection as an infinite pushforward, which expands the Lebesgue unstable divergence. In general, unstable divergence can not be computed by summing individual directional derivatives, since they are only distributions but not

functions. For this general situation, our expansion is the first computable formula for unstable divergence.

Comparing to the original roadmap given by Ruelle, we make the definitions coordinate-free, and derive the detailed formulas, which are necessary for numerics, but were previously missing. Deriving such a formula was also attempted by some earlier works. The S3 algorithm did not give a computable formula of the Lebesgue unstable divergence, which is still a distribution. Hence, S3 works only on essentially one dimensional systems, where the unstable divergence can be directly computed from directional derivatives; however, this is not the case for typical fractal attractors [16, 76].

Integration by parts and measure change

In order to obtain a smaller integrand for the unstable contribution, we integrate-by-parts on the unstable manifold under the conditional SRB measure. This yields unstable divergence under SRB measure, which relates to the Lebesgue unstable divergence by a measure change. This subsection presents this classical treatment with some previously missing formulas.

Throughout this chapter, we make the assumptions in theorem 1 in our proofs. Recall that the conditional SRB measure on an unstable manifold $\mathcal{V}^u$ has a density, or the Radon-Nikodym derivative with respect to the u-dimensional Lebesgue measure, which is denoted by σ. Let ω be the volume form on $\mathcal{V}^u$, ρ be the SRB measure. Integrations to ρ can be done by first integrating to $\sigma\omega$ on the unstable manifold, then in the transversal direction.

We first explain the integration by parts. For any $\varphi \in C^1$,

$$X^u(\varphi)\sigma + \varphi X^u(\sigma) = X^u(\varphi\sigma) = \operatorname{div}^u(\varphi\sigma X^u) - \varphi\sigma \operatorname{div}^u X^u.$$

Here $X^u(\cdot)$ means to differentiate a function in the direction of X^u, div^u is the divergence on the unstable manifold under Riemannian metric. For now, we only know that X^u is Holder continuous, and its derivatives are distributions; later, we will prove the regularity of the unstable divergence. Integrate on a piece of unstable manifold $\mathcal{V}^u$, we have the integration-by-parts formula

$$\int_{\mathcal{V}^u} X^u(\varphi)\sigma\omega = \int_{\mathcal{V}^u} \operatorname{div}^u(\varphi\sigma X^u)\omega - \int_{\mathcal{V}^u} \left(\frac{\varphi}{\sigma} X^u(\sigma)\right) \sigma\omega - \int_{\mathcal{V}^u} (\varphi \operatorname{div}^u X^u)\, \sigma\omega. \tag{6.1}$$

We will deal with the first two terms on the right hand side in this subsection, and leave the last term, which involves the unstable divergence, to the next two subsections.

When further integrating over the entire attractor with SRB measure ρ, the first term on the right of equation (6.1) becomes zero. To see this, first notice that the divergence theorem reduces it to boundary integrals. Intuitively, since unstable manifolds always lie within the attractor, and they do not have boundaries, the boundary integral would never appear when integrating over ρ, and hence this term becomes zero. More rigorously, first choose a Markov

partition with a small diameter, then notice that the boundary integrals cancel between two adjacent rectangles. To conclude, we have now

$$\rho(X^u(\varphi)) = -\rho\left(\varphi \operatorname{div}^u_\sigma X^u\right), \text{ where } \operatorname{div}^u_\sigma X^u := \frac{1}{\sigma} X^u(\sigma) + \operatorname{div}^u X^u.$$

In the rest of this chapter, let

$$e := e_1 \wedge \cdots \wedge e_u \in (V^u)^{\wedge u} \tag{6.2}$$

be a rough u-vector field, which is differentiable on each unstable manifold, but not necessarily continuous in all directions. Also assume that e and $\nabla_{(\cdot)} e$ are bounded on the attractor K under Riemannian metric, where $\nabla_{(\cdot)} e$ is the Riemannian connection operating on vectors in V^u. In fact, we may regard e as C^1 u-vector fields on individual unstable manifolds. We can make it continuous across the foliation, for example,

$$\tilde{e} := \frac{e}{\|e\|}$$

is unique and continuous modulo an orientation, and it satisfies our boundedness assumption, because the unstable manifold theorem states that, in our case, unstable manifolds are continuous in C^3 topology [40]. Here the tensor norm $\|\cdot\|$ is induced by the Riemannian metric, which is the u-dimensional volume of the hyper-cube spanned by $\{e_i\}_{i=1}^u$. We use e instead of $\tilde{e}$ if the statement holds more generally.

SRB measure is the weak limit of pushing-forward the Lebesgue measure. Roughly speaking, pushing-forward is like a matrix multiplication. Hence, by the Leibniz rule, the measure change term, $X^u(\sigma)/\sigma$, can be expanded into an infinite summation, with detailed formula given below.

Lemma 6 (expression for measure change)**.** *The following formula converges uniformly on K*

$$\frac{1}{\sigma} X^u(\sigma) = \sum_{k=1}^{\infty} -\frac{\left\langle \nabla_{f_*^{-k+1} X^u} f_* e_{-k}, f_* e_{-k} \right\rangle}{\langle f_* e_{-k}, f_* e_{-k} \rangle} + \frac{\left\langle \nabla_{f_*^{-k} X^u} e_{-k}, e_{-k} \right\rangle}{\langle e_{-k}, e_{-k} \rangle}.$$

Here e_{-k} can be any differentiable u-vector field on $f^{-k}(\mathcal{V}^u)$. For example, we can take it as the one such that e and $\nabla_{(\cdot)} e$ are bounded.

Remark. (1) If we evaluate both side of the equation at x, then $e_{-k} = e(f^{-k}x)$, and when being differentiated, e_{-k} is a restriction of e to a neighborhood of $x_{-k} = f^{-k}x$. (2) Due to uniform convergence, $X^u(\sigma)/\sigma$ is uniform continuous over K. (3) Moreover, an algorithm computing this expansion would converge to the true solution. (4) A more careful analysis would show that this term is in fact Holder, as claimed in [70]; we will not pursue Holder continuity here, for it does not directly affect the algorithm.

Proof. The conditional SRB measure σ on $\mathcal{V}^u$ is the result of evolving the Lebesgue measure starting from the infinite past. More specifically,

$$\sigma = \prod_{k=1}^{\infty} C_k \left(J^u_{-k}\right)^{-1},$$

where C_k is constant over $\mathcal{V}^u_k$ to keep the total conditional measure at 1, and J^u is the unstable Jacobian computed with respect to the Riemannian metric on $\mathcal{V}^u$, more specifically, for any e,

$$J^u := \frac{\|f_* e\|}{\|e\|} = \left(\frac{\langle f_* e, f_* e\rangle}{\langle e, e\rangle}\right)^{0.5}. \tag{6.3}$$

Notice J^u does not depend on the particular choice of e. By the Leibniz rule of differentiation, and the notation convention explained in equation (2.2),

$$\frac{1}{\sigma} X^u(\sigma) = -\sum_{k=1}^{\infty} \frac{X^u(J^u_{-k})}{J^u_{-k}} = -\sum_{k=1}^{\infty} \frac{(f_*^{-k} X^u) J^u_{-k}}{J^u_{-k}}. \tag{6.4}$$

To get the equation in the lemma, substitute (6.3) into (6.4). For any vector $Y \in V^u$,

$$\begin{aligned} Y(J^u) &= Y\left(\frac{\langle f_* e, f_* e\rangle}{\langle e, e\rangle}\right)^{0.5} = \frac{1}{2}(J^u)^{-1} Y\left(\frac{\langle f_* e, f_* e\rangle}{\langle e, e\rangle}\right) \\ &= \frac{1}{2}(J^u)^{-1} \frac{\langle f_* e, f_* e\rangle}{\langle e, e\rangle}\left[\frac{f_* Y \langle f_* e, f_* e\rangle}{\langle f_* e, f_* e\rangle} - \frac{Y\langle e, e\rangle}{\langle e, e\rangle}\right] = J^u\left[\frac{\langle \nabla_{f_* Y} f_* e, f_* e\rangle}{\langle f_* e, f_* e\rangle} - \frac{\langle \nabla_Y e, e\rangle}{\langle e, e\rangle}\right]. \end{aligned}$$

In the last equality we used the rule of differentiating Riemannian metric.

To see uniform convergence, take e be the one such that both e and $\nabla_{(\cdot)} e$ are bounded on K, then the series is controlled by the exponentially shrinking term, $\|f_*^{-k} X^u\| \le C\lambda^{-k}\|X^u\|$. □

We have expanded the difference between two unstable divergence under two measures, however, the Lebesgue unstable divergence, $\operatorname{div}^u X^u$, is still not computable, since it is defined via directional derivatives, which are distributions. The S3 algorithm has something somewhat similar to lemma 6; however, the Lebesgue unstable divergence was still computed by summing up directional derivatives approximated by finite difference [16, 76]. Hence, S3 did not give a computable formula by our definition, and may suffer from numerical instabilities, because X^u is in fact not differentiable; moreover, finite difference is not efficient, since the computation at one step can not be reused for other steps. We proceed to obtain a computable expansion formula of the unstable divergence, which was missing from previous works.

Transforming to derivative of volume ratio

The main difficulty, in both proving the regularity and computing the unstable divergence, is that V^s, hence X^u, are typically not differentiable along any directions in the unstable manifold; hence, directional derivatives are distributions but not functions. However, $\mathrm{div}^u X^u$ can be more regular. In fact, Ruelle proved that $\mathrm{div}^u X^u$ is Holder continuous over K [70]. The key step is to show that the unstable divergence equals the derivative of a ratio between two volumes given by two projections, η_* and ξ_*. Here η_* is the projection along X; ξ_* is the projection along stable manifolds. This subsection presents this classical treatment in a coordinate-free manner, and derives the detailed formulas which were previously missing.

Without loss of generality, we assume that the angle between the vector field X and $\mathcal{V}^u$ is uniformly away from zero. If not, then we can find a vector field in the stable subspace, mollify it, then multiply by a large constant: denote the resulting vector field by X', which is always non-parallel to $\mathcal{V}^u$. We can compute the sensitivity caused by X as the sum of those caused by $X + CX'$ and $-CX'$.

Fix an unstable manifold $\mathcal{V}^u$, let $q \in \mathbb{R}$ be a small parameter. for any $y \in \mathcal{V}^u$, define $\eta^q(y) : \mathcal{V}^u \to \mathcal{M}$ as the unique curve such that $\partial \eta^q / \partial q = X$ and $\eta^0(y) = y$. For fixed q, $\mathcal{V}^{uq} := \{\eta^q(y) : y \in \mathcal{V}^u\}$ is a u-dimensional C^3 manifold; for a small interval of q, $\hat{\mathcal{V}}^u := \cup_q \mathcal{V}^{uq}$ is a $u+1$ dimensional manifold. For any $y \in \mathcal{V}^u$, denote the stable manifold that goes through it by $\mathcal{V}^s(y)$, which does not vary differentiably with y in C^3. Define $\xi^q(y)$ as the unique intersection point of $\mathcal{V}^s(y)$ and $\mathcal{V}^{uq}$.

Define $\pi_\eta, \pi_\xi : \hat{\mathcal{V}}^u \to \mathcal{V}^u$ such that $\pi_\eta(x) = (\eta^q)^{-1} x$, $\pi_\xi(x) = (\xi^q)^{-1} x$, for any $x \in \mathcal{V}^{uq}$ and small $q \in \mathbb{R}$. Denote η^q_*, ξ^q_* as the pushforward operator of η^q, ξ^q. For any small $q \in \mathbb{R}$, define

$$\eta_* e(x) := \eta^q_* e(\pi_\eta(x)), \quad \xi_* e := \xi^q_* e(\pi_\xi(x)), \quad \text{for } x \in \mathcal{V}^{uq}.$$

Then $\eta_* e$ and $\xi_* e$ are two parallel u-vector fields on $\hat{\mathcal{V}}^u$. (We may equivalently define $\eta_* e := \cup_q \eta^q_* e$, $\xi_* e := \cup_q \xi^q_* e$.) Define a function on $\hat{\mathcal{V}}^u$, ϖ, as the volume ratio,

$$\frac{\eta_* e}{\|e \circ \pi_\eta\|} = \varpi \frac{\xi_* e}{\|e \circ \pi_\xi\|}, \tag{6.5}$$

By transversal absolute continuity, ϖ is a well-defined measurable function. The definitions are illustrated in figure 6.1. Comparing to Ruelle's original definitions, ours are coordinate-free.

Lemma 7 (expression of $\nabla_{X^s} \eta_* e$)**.** *Denote* $\nabla_e X := \sum_i e_1 \wedge \cdots \wedge \nabla_{e_i} X \wedge \cdots \wedge e_u$. *On* $\mathcal{V}^u$,

$$\nabla_{X^s}(\eta_* e_i) = \nabla_{e_i} X - \nabla_{X^u} e_i \,, \quad \nabla_{X^s}(\eta_* e) = \nabla_e X - \nabla_{X^u} e.$$

Remark. $\nabla_e X$ is a function, not a distribution, because X is differentiable on $\mathcal{M}$. $\nabla_{X^u} e$ also is a function, since it only requires e be differentiable along the direction of X^u. Differentiability of X^u is not and should not be required, since X^u is not differentiable.

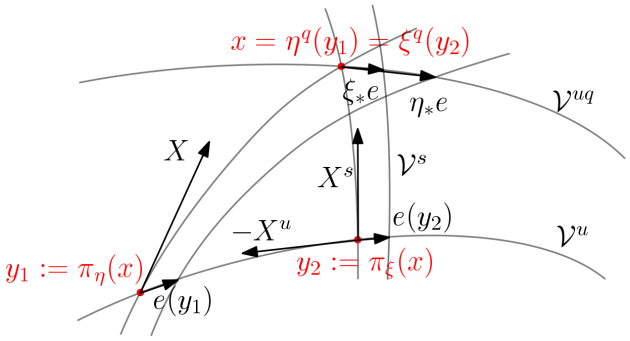


Figure 6.1: Definitions of projections.

Proof. With our notation, the proof below works for both e and e_i. First decompose,

$$\nabla_{X^s}(\eta_* e) = \nabla_X(\eta_* e) - \nabla_{X^u}(\eta_* e)$$

Since η is the flow of X, the Lie derivative $L_X(\eta_* e) = 0$, hence, on $\mathcal{V}^u$,

$$\nabla_X(\eta_* e) = \nabla_{\eta_* e} X = \nabla_e X,$$

where $\eta_* e = e$ on $\mathcal{V}^u$. Since X^u is a vector field on $\mathcal{V}^u$, $\nabla_{X^u}(\eta_* e) = \nabla_{X^u} e$. □

Lemma 8. *Further assume that the stable foliation is C^1 with C^3 leaves, then on $\mathcal{V}^u$,*

$$\nabla_{X^s}(\xi_* e_i) = \nabla_{e_i} X^s, \quad \nabla_{X^s}(\xi_* e) = \nabla_e X^s.$$

Remark. The extra regularity assumption is for $\xi_* e$ to be a differentiable u-vector field on $\mathcal{V}^{uq}$. Without this assumption, we can still make sense of this equation as distributional derivatives. The roughness of $\xi_* e$ hints us to further replace it in section 6.1.

Proof. On $\hat{\mathcal{V}}^u$, $\partial \xi^q / \partial q$ is a C^1 vector field. We claim that $\partial \xi^q / \partial q = X^s$ on $\mathcal{V}^u$. To see this, define $y^q : \mathcal{V}^u \to \mathcal{V}^u$ as

$$y^q(x) = \pi_\eta(\xi^q(x)), \quad \text{or equivalently,} \quad \xi^q(x) = \eta^q(y^q(x)).$$

Fix x, differentiate the second equation to q, evaluate at $q = 0$, where $\partial \eta^q / \partial y = Id$,

$$\frac{\partial \xi^q}{\partial q} = \frac{\partial \eta^q}{\partial y}\frac{\partial y^q}{\partial q} + \frac{\partial \eta^q}{\partial q} = \frac{\partial y^q}{\partial q} + X.$$

Since $\partial y^q / \partial q \in V^u$, $\partial \xi^q / \partial q \in V^s$, and that X is uniquely decomposed in $V^u \bigoplus V^s$ into $X = X^u \oplus X^s$, we see that

$$\frac{\partial \xi^q}{\partial q} = X^s, \quad \frac{\partial y^q}{\partial q} = -X^u \quad \text{at} \quad q = 0. \tag{6.6}$$

Hence, on $\mathcal{V}^u$,

$$\nabla_{X^s}(\xi_* e) = \nabla_{\partial\xi^q/\partial q}(\xi_* e) = \nabla_{\xi_* e}\partial\xi^q/\partial q = \nabla_e X^s.$$

Here second equality is because the Lie derivative $L_{\partial\xi^q/\partial q}(\xi_* e) = 0$ by definitions. □

Lemma 9. *Assume that the stable foliation is C^1, then $X^s(\varpi) = \operatorname{div}^u X^u$ on $\mathcal{V}^u$.*

Remark. In [70], Ruelle went on to show that this equivalence persists into general cases, where the stable foliation is not C^1. The main technique is to approximate the stable foliation by evolving a smooth foliation backward in time; we will not reproduce that proof here, for it does not directly help the computation.

Proof. By definitions,

$$y^q(x) = \pi_\eta(\xi^q(x)), \quad x = \pi_\xi(\xi^q(x)).$$

Differentiate to q, and use equation (6.6), we get

$$-X^u = \frac{\partial y^q}{\partial q} = \pi_{\eta *}\frac{\partial \xi^q}{\partial q} = \pi_{\eta *}X^s, \quad 0 = \pi_{\xi *}\frac{\partial \xi^q}{\partial q} = \pi_{\xi *}X^s \quad \text{on} \quad \mathcal{V}^u.$$

Hence,

$$X^s\|e\circ\pi_\eta\|^2 = (\pi_{\eta *}X^s)\|e\|^2 = \nabla_{-X^u}\|e\|^2 = 2\left\langle \nabla_{-X^u}e, e\right\rangle, \quad X^s\|e\circ\pi_\xi\|^2 = 0.$$

Take inner product of each side of equation (6.5) with itself,

$$\|\eta_* e\|^2\,\|e\circ\pi_\xi\|^2 = \varpi^2\|\xi_* e\|^2\|e\circ\pi_\eta\|^2.$$

Differentiate in the direction of X^s, notice that $\varpi = 1$, $\eta_* = \xi_* = Id$ on $\mathcal{V}^u$, hence

$$\left\langle \nabla_{X^s}(\eta_* e), e\right\rangle = X^s(\varpi)\left\langle e, e\right\rangle + \left\langle \nabla_{X^s}(\xi_* e), e\right\rangle + \left\langle \nabla_{-X^u}e, e\right\rangle.$$

By lemma 7 and 8, we have

$$X^s(\varpi) = \frac{1}{\langle e, e\rangle}\left\langle \nabla_e X - \nabla_{X^u}e - \nabla_e X^s + \nabla_{X^u}e, e\right\rangle = \frac{1}{\langle e, e\rangle}\left\langle \nabla_e X^u, e\right\rangle.$$

Since the Riemannian connection within a submanifold $\mathcal{V}^u$ is the orthogonal projection of that on the background manifold M,

$$X^s(\varpi) = \frac{1}{\langle e, e\rangle}\left\langle \nabla^u_e X^u, e\right\rangle_u = \frac{1}{\langle e, e\rangle}\sum_{i=1}^{u}\left\langle e_1\wedge\cdots\wedge\sum_j e^j_u(\nabla^u_{e_i}X^u)e_j\wedge\cdots\wedge e_u, e\right\rangle_u.$$

Here ∇^u, $\langle\cdot,\cdot\rangle_u$, and e^j_u are the Riemannian connection, metric, and dual basis of e_j within $\mathcal{V}^u$. Terms with $j\neq i$ vanish due to that the same direction appears twice in the exterior product, hence

$$X^s(\varpi) = \sum_{i=1}^{u} e^i_u(\nabla^u_{e_i}X^u).$$

This is a contraction of $\nabla^u X^u$ within $\mathcal{V}^u$, which is the definition of $\operatorname{div}^u X^u$. □

Expanding volume ratio

We show that $X^s(\varpi)$ is continuous by expanding the volume ratio into a converging summation. The main problem now is that $\xi_* e$ is not yet differentiable. Hence, we further write $\xi_* e$ as an infinite pushfoward, whose derivative is an infinite summation. This finally expands the derivative of the volume ratio, and hence Lebesgue unstable divergence.

Lemma 10 (expansion of Lebesgue unstable divergence)**.** *The following formula converges uniformly on K*

$$\operatorname{div}^u X^u = \frac{\langle \nabla_e X, e\rangle}{\|e\|^2} + \sum_{k=0}^{\infty} \frac{\left\langle \nabla_{f_*^{k+1} X^s} f_*^{k+1}\eta_* e, f_*^{k+1} e\right\rangle}{\|f_*^{k+1} e\|^2} - \frac{\left\langle \nabla_{f_*^k X^s} f_*^k \eta_* e, f_*^k e\right\rangle}{\|f_*^k e\|^2}.$$

Remark. Due to uniform convergence, $\operatorname{div}^u X^u$ is uniform continuous over K. A more careful analysis would show that it is Holder continuous over K, as claimed in [70].

Proof. By definition of the stable manifold, for $x \in \hat{\mathcal{V}}^u$, $\lim_{k\to\infty} f^k(\pi_\xi x) = \lim_{k\to\infty} f^k(x)$. Hence,

$$\lim_{k\to\infty} \|f_*^k \xi_* e\|^2 / \|f_*^k (e \circ \pi_\xi)\|^2 = 1.$$

Because $\xi_* e$ is parallel to $\eta_* e$,

$$\frac{\|\eta_* e\|}{\|\xi_* e\|} = \lim_{k\to\infty} \frac{\|f_*^k \eta_* e\|}{\|f_*^k \xi_* e\|} = \lim_{k\to\infty} \frac{\|f_*^k \eta_* e\|}{\|f_*^k (e\circ\pi_\xi)\|}$$

We can use this to replace $\xi_* e$ in the definition of ϖ in equation (6.5), which yields an infinite pushforward:

$$\varpi^2 = \frac{\|e\circ\pi_\xi\|^2}{\|e\circ\pi_\eta\|^2} \lim_{k\to\infty} \frac{\|f_*^k \eta_* e\|^2}{\|f_*^k (e\circ\pi_\xi)\|^2} = \frac{\|\eta_* e\|^2}{\|e\circ\pi_\eta\|^2} \prod_{k=0}^{\infty} \frac{\|f_*^{k+1}\eta_* e\|^2}{\|f_*^{k+1} e\circ\pi_\xi\|^2} \frac{\|f_*^k e\circ\pi_\xi\|^2}{\|f_*^k \eta_* e\|^2}.$$

Further differentiating in direction X^s would yield an infinite summation. More specifically, notice that at $\mathcal{V}^u$, $\eta_* e = e\circ\pi_\xi = e$, $\varpi = 1$, and apply equation (2.2),

$$X^s(\varpi) = \frac{\langle \nabla_{X^s}\eta_* e, e\rangle}{\|e\|^2} - \frac{\langle \nabla_{-X^u} e, e\rangle}{\|e\|^2} + \sum_{k=0}^{\infty} \frac{\left\langle \nabla_{f_*^{k+1}X^s} f_*^{k+1}\eta_* e, f_*^{k+1} e\right\rangle}{\|f_*^{k+1} e\|^2} - \frac{\left\langle \nabla_{f_*^k X^s} f_*^k \eta_* e, f_*^k e\right\rangle}{\|f_*^k e\|^2}.$$

The equality in the lemma is obtained by lemma 7 and 9.

For the uniform convergence, using the Leibniz rule in appendix 6.5 lemma 20, and the projection operators defined in appendix 6.5, we can decompose the k-th term in the summation of the lemma, S_k, into $S_k = S_{k1} + S_{k2} + S_{k3}$, where

$$S_{k1} := \frac{1}{\|f_*^{k+1} e\|^2} \left\langle (\nabla_{f_*^k X^s} f_*) f_*^k e, f_*^{k+1} e\right\rangle \le C\lambda^k \|X^s\|.$$

since X^s decays exponentially via pushforwards. Here ∇f_* is the Riemannian connection of f_* (appendix 6.5 definition 6). Also,

$$S_{k2} := \frac{1}{\|f_*^{k+1}e\|^2}\Big\langle f_* P^u \nabla_{f_*^{k+1}X^s} f_*^k \eta_* e, f_*^{k+1} e\Big\rangle - \frac{1}{\|f_*^k e\|^2}\Big\langle P^u \nabla_{f_*^k X^s} f_*^k \eta_* e, f_*^k e\Big\rangle = 0,$$

This is because $(V^u)^{\wedge u}$ is 1-dimensional, so $P^u \nabla_{\frac{\partial}{\partial q}} f_*^k \eta_* e(y)$ and $f_*^k e$ increase by same amounts via the pushforward. Finally,

$$S_{k3} := \frac{1}{\|f_*^{k+1}e\|^2}\Big\langle f_* P^s \nabla_{f_*^{k+1}X^s} f_*^k \eta_* e, f_*^{k+1} e\Big\rangle - \frac{1}{\|f_*^k e\|^2}\Big\langle P^s \nabla_{f_*^k X^s} f_*^k \eta_* e, f_*^k e\Big\rangle.$$

This term also converges uniformly on K, since

$$\begin{aligned} P^s \nabla_{f_*^k X^s} f_*^k \eta_* e &= f_*^k P^s(\nabla_e X - \nabla_{X^u} e) + \sum_{n=0}^{k-1} f_*^{k-n-1} P^s(\nabla_{f_*^n X^s} f_*)\, f_*^n e \\ &\le C\lambda^k \|X\| + C\sum_{n=0}^{k-1} \lambda^{k-n-1} \|f_*^n X^s\| \|f_*^n e\| \le \frac{C\|X\|}{1-\lambda}. \end{aligned} \tag{6.7}$$

Hence, $S_{k3} \le C\lambda^k \|X\|/(1-\lambda)$, and $\sum_{k\ge 0} S_k$ uniformly converges. □

Theorem 7 (expansion of unstable divergence)**.** *Define*

$$\psi := \sum_{m=-W}^{W} (\Phi_m - \rho(\Phi)), \quad \Psi := \psi X.$$

By lemma 6 and lemma 10, the unstable contribution is $U.C. = \lim_{W\to\infty} U.C.^W$*, where*

$$\begin{aligned} U.C.^W &:= \rho(\psi \operatorname{div}^u_\sigma X^u) = \rho\left(\psi \operatorname{div}^u X^u + \frac{\psi}{\sigma} X^u(\sigma)\right) \\ =\rho\Bigg[&\frac{\langle \psi \nabla_e X, e\rangle}{\|e\|^2} + \sum_{k=0}^{\infty}\left(\frac{\Big\langle \nabla_{f_*^{k+1}\Psi^s} f_*^{k+1}\eta_* e, f_*^{k+1} e\Big\rangle}{\|f_*^{k+1}e\|^2} - \frac{\Big\langle \nabla_{f_*^k \Psi^s} f_*^k \eta_* e, f_*^k e\Big\rangle}{\|f_*^k e\|^2}\right) \\ &- \sum_{k=1}^{\infty}\left(\frac{\Big\langle \nabla_{f_*^{-k+1}\Psi^u} f_* e_{-k}, f_* e_{-k}\Big\rangle}{\|f_* e_{-k}\|^2} - \frac{\Big\langle \nabla_{f_*^{-k}\Psi^u} e_{-k}, e_{-k}\Big\rangle}{\|e_{-k}\|^2}\right)\Bigg], \end{aligned}$$

Here σ is the density of the conditional SRB measure, e_{-k} is a u-vector field on $f^{-k}\mathcal{V}^u$ as defined in equation (6.2)*, and $(\cdot)^u, (\cdot)^s$ are the unstable and stable projections of a vector.*

Remark. (1) The uniform convergence in lemma 6 and 10 shows that this formula also converges uniformly. In fact, we did not use the full strength of uniform hyperbolicity to prove uniform convergences; moreover, weaker forms of convergence could also suffice our computational purpose. (2) Note that adding a constant to Φ does not change the linear response, but it helps to reduce numerical errors.

Since the integrand is much smaller than the original linear response formula, directly computing the expansion formula in theorem 7 would have much faster convergence than algorithms based on the original linear response formula. However, that would require at least solving at least u^2 second-order tangent equations, which shall be defined later. It also requires the oblique projections, which can be computed via 'little-intrusive' algorithm, whose cost is twice of non-intrusive shadowing [54, 55]; moreover, it might not be very robust, since it requires artificially deciding whether a vector is stable or unstable. In the next section we further give the expansion formula a 'fast' characterization which is even faster, and do not involve oblique projections.

6.2 Fast characterization of unstable divergence

There are several well-known fast algorithms, such as the fast Fourier transformation [24] and the fast multipole method [36, 37]. Roughly speaking, a common idea in these fast algorithms is to combine many small terms into a few big terms, and then apply some expensive operation on the big terms only a few times, instead of many times on individual small terms. In this section, we find an 'embarrassingly fast' structure for the expansion formula of the unstable divergence. In our case, the most expensive operation is the renormalized second-order tangent equation, which governs the propagation of derivatives of vectors. We use the linearity of ∇e to define only one big term, which is a summation, p; hence, the second-order tangent equation only need to be applied to p once, instead many times on each ∇e.

In this section, we show that both the unstable contribution and the inductive relation of p can be expressed via this 'first combine then propagate' strategy. Finally, we show that p is uniquely determined by the inductive relation it satisfies, hence, we obtain a 'fast' characterization of the unstable divergence. It also turns out that, in the fast characterization, the oblique projection is replaced by an orthogonal projection, hence, the fast characterization is also non-intrusive, which means that it requires only first and second order tangent solutions.

Definitions of p, β, and U

The expression in theorem 7 computes too many times ∇f_*, which is the Riemannian connection of the pushforward tensor defined in appendix 6.5. To save computational efforts, we seek to combine terms with ∇f_* at the same step. To achieve this, first let e_{-k} be $\tilde{e}_{-k}$ in

the second summation, where $\tilde{e} = e/\|e\|$, then, by the invariance of the SRB measure,

$$\begin{aligned} U.C.^W = \rho \Bigg[\sum_{k=0}^{\infty} \left(\frac{\left\langle \nabla_{f_*^{k+1}\Psi^s_{-k}} f_*^{k+1}\eta_* e_{-k}, f_*^{k+1} e_{-k} \right\rangle}{\|f_*^{k+1} e_{-k}\|^2} - \frac{\left\langle \nabla_{f_*^k \Psi^s_{-k}} f_*^k \eta_* e_{-k}, f_*^k e_{-k} \right\rangle}{\|f_*^k e_{-k}\|^2} \right) \\ + \frac{\langle \psi_1 \nabla_{e_1} X_1, e_1 \rangle}{\|e_1\|^2} - \sum_{k=1}^{\infty} \left(\frac{\left\langle \nabla_{f_*^{1-k}\Psi^u_k} f_* \tilde{e}, f_* \tilde{e} \right\rangle}{\|f_* \tilde{e}\|^2} - \left\langle \nabla_{f_*^{-k}\Psi^u_k} \tilde{e}, \tilde{e} \right\rangle \right) \Bigg], \end{aligned} \tag{6.8}$$

where the subscript $(\cdot)_1$ labels steps, $\Psi_k := \psi_k X_k$, e is the u-dimensional hyper-cube defined in equation (6.2), $\tilde{e}$ is the normalized hyper-cube, η is the projection map along $X = \delta f \circ f^{-1}$ (see section 6.1), and the definition of $\nabla_{(\cdot)_k}(\cdot)_k$ is in equation (2.1).

Because $\wedge^u V^u$ is one-dimensional, and that the sign generated by manifold orientation is canceled within the inner products, we may assume without loss of generality that $f_*^k e_{-k}/\|f_*^k e_{-k}\| = \tilde{e}$ for all k. Hence, we may pull out $\tilde{e}$ from the second term of each summand in equation (6.8), and define

$$\begin{aligned} p &:= \sum_{k=0}^{\infty} \frac{1}{\|f_*^k e_{-k}\|} P^{\perp} \nabla_{f_*^k \Psi^s_{-k}} f_*^k \eta_* e_{-k} - \sum_{k=1}^{\infty} \nabla_{f_*^{-k}\Psi^u_k} \tilde{e} = \sum_{k\geq 0} p^{\perp}_{(k)} - \sum_{k \geq 1} p'_{(k)}\,, \\ &\text{where} \quad p_{(k)} := \frac{\nabla_{f_*^k \Psi^s_{-k}} f_*^k \eta_* e_{-k}}{\|f_*^k e_{-k}\|}\,, \quad p^{\perp}_{(k)} := P^{\perp} p_{(k)}\,, \quad p'_{(k)} := \nabla_{f_*^{-k}\Psi^u_k} \tilde{e}\,. \end{aligned} \tag{6.9}$$

Here $P^{\perp}$ is the orthogonal projection operator (appendix 6.5 definition 8), and notice that $p_{(k)} \neq p_k$. We will see later that $P^{\perp}$ takes out keeps only a convergent component in the first summation, while the normalized $\tilde{e}$ makes the second summation in $\mathcal{D}^{\perp}_e := P^{\perp}\mathcal{D}_e$, which is the orthogonal projection of the space of derivative-like u-vectors (appendix 6.5).

Lemma 11. *p is a convergent summation in $\mathcal{D}^{\perp}_e$.*

Proof. Since $\tilde{e}$ has constant volume, we have $\langle \nabla \tilde{e}, \tilde{e} \rangle = 0$. By the linearity of the projection operator, $p \in \mathcal{D}^{\perp}_e$ should it converge. To see the convergence of the first summation, use the estimation in equation (6.7) and that $P^{\perp} = P^{\perp} P^s$ from lemma 23. The convergence of the second summation is because $f_*^{-k}\Psi^u_k$ decays exponentially. □

To simplify our writing, we define two maps and show some of their properties. The map β's are renormalized second-order tangent equations, which governs the propagation of the derivatives of u-vectors, and U's are the integrand in the unstable contribution.

Definition 4. For any $r \in \mathcal{D}_e, Y \in T_x M$, define

$$\begin{aligned} \beta_Y(r) &:= \left(f_* r + (\nabla_Y f_*)\tilde{e}\right)/\|f_* \tilde{e}\|. \\ \tilde{\beta}(r) &:= \beta_{\tilde{v}} r + \psi_1 \nabla_{\tilde{e}_1} X_1, \end{aligned}$$

$$U_Y(r) := \langle \beta_Y(r), \tilde{e}_1 \rangle - \langle r, \tilde{e} \rangle ,$$
$$\tilde{U}(r) := \left\langle \tilde{\beta}(r), \tilde{e}_1 \right\rangle - \langle r, \tilde{e} \rangle = U_{\tilde{v}}(r) + \langle \psi_1 \nabla_{\tilde{e}_1} X_1, \tilde{e}_1 \rangle .$$

Here ∇f_* is the Riemannian connection of f_* (appendix 6.5 definition 6), $\tilde{v}$ is the shadowing direction of Ψ,

$$\tilde{v} := \sum_{k=0}^{\infty} f_*^k \Psi_{-k}^s - \sum_{k=1}^{\infty} f_*^{-k} \Psi_k^u.$$

Remark. (1) Similar to our convention on pushforward operators, $\beta_Y(r(x)), \tilde{\beta}(r(x)) \in T_{fx}M$. (2) The oblique projections are summarized into the modified shadowing direction $\tilde{v}$. By section 2.3, $\tilde{v}$ can be efficiently computed by non-intrusive shadowing algorithms, which does not require computing oblique projections. (3) We refer to β_Y as the renormalized homogeneous second-order tangent equation, and $\tilde{\beta}$ as the renormalized inhomogeneous equation. (4) The second-order tangent equation is in fact also an inhomogeneous first-order tangent equation, with a second-order inhomogeneous term, ∇f_*.

Lemma 12 (properties of β and U)**.** *For any $r, r' \in \mathcal{D}_e$, and $X, Y \in T_x\mathcal{M}$,*

1. *$\beta_X r' \pm \beta_Y r = \beta_{X \pm Y}(r' \pm r)$, $U_X r' \pm U_Y r = U_{X \pm Y}(r' \pm r)$;*
2. *for both β_Y and $\tilde{\beta}$, we have $\beta r \in \mathcal{D}_e$, $P^\perp \beta r = P^\perp \beta r^\perp$, where $r^\perp := P^\perp r$;*
3. *for both U_Y and $\tilde{U}$, $U(r) = U(r^\perp)$.*

Proof. (1) By definition. (2) By definition, then lemma 24 in appendix 6.5. (3) Since $\wedge^u V^u$ is one-dimensional, all of its u-vectors are increased by same amounts by the pushforward. Hence, $\left\langle f_* r^\parallel, \tilde{e}_1 \right\rangle / \| f_* \tilde{e} \| = \left\langle r^\parallel, \tilde{e} \right\rangle$, and

$$U(r) - U(r^\perp) = \left\langle \beta(r) - \beta(r^\perp), \tilde{e}_1 \right\rangle - \left\langle r - r^\perp, \tilde{e} \right\rangle$$
$$= \left\langle \beta_0(r^\parallel) \tilde{e}_1 \right\rangle - \left\langle r^\parallel, \tilde{e} \right\rangle = \left\langle \frac{f_* r^\parallel}{\| f_* \tilde{e} \|}, \tilde{e}_1 \right\rangle - \left\langle r^\parallel, \tilde{e} \right\rangle = 0.$$

Here β_0 means $Y = 0$ in definition 4. □

Unstable contribution expressed by p

The most expensive operation in the expression of the unstable contribution is to propagate the derivatives of u-vectors by second-order tangent equations, β. With the definition of the summation, p, now we only need to apply $\tilde{\beta}$ once on p, instead of many times on each summand in the definition of p. Hence the computation of U from p is much more efficient. Moreover, we no longer need to compute oblique projections, since $\tilde{v}$ can be computed by non-intrusive shadowing. In the next subsection, we show that similar reductions happen for the inductive relation of p.

Lemma 13. $U.C.^W = \rho(\tilde{U}(p))$.

Proof. By the Leibniz rule in appendix 6.5 lemma 20, $\eta_* = Id$ at $q = 0$, and the definition of $p_{(k)}$ and $p'_{(k)}$ in equation (6.9), we have

$$\begin{aligned} \frac{\nabla_{f_*^{k+1}\Psi^s_{-k}} f_*^{k+1}\eta_* e_{-k}}{\|f_*^{k+1}e_{-k}\|} &= \frac{1}{\|f_*\tilde{e}\|}\left(f_* p_{(k)} + (\nabla_{f_*^k\Psi^s_{-k}} f_*)\tilde{e}\right) = \beta_{f_*^k\Psi^s_{-k}}(p_{(k)}), \\ \frac{\nabla_{f_*^{1-k}\Psi^u_k} f_*\tilde{e}}{\|f_*\tilde{e}\|} &= \frac{1}{\|f_*\tilde{e}\|}\left(f_* p'_{(k)} + (\nabla_{f_*^{-k}\Psi^u_k} f_*)\tilde{e}\right) = \beta_{f_*^{-k}\Psi^u_k}(p'_{(k)}). \end{aligned} \tag{6.10}$$

The summand in the first summation of equation (6.8) becomes

$$\begin{aligned} &\frac{\left\langle \nabla_{f_*^{k+1}\Psi^s_{-k}} f_*^{k+1}\eta_* e_{-k}, f_*^{k+1}e_{-k}\right\rangle}{\|f_*^{k+1}e_{-k}\|^2} - \frac{\left\langle \nabla_{f_*^k\Psi^s_{-k}} f_*^k\eta_* e_{-k}, f_*^k e_{-k}\right\rangle}{\|f_*^k e_{-k}\|^2} \\ =&\left\langle \beta_{f_*^k\Psi^s_{-k}} p_{(k)}, \tilde{e}_1\right\rangle - \left\langle p_{(k)}, \tilde{e}\right\rangle = U_{f_*^k\Psi^s_{-k}}(p_{(k)}) = U_{f_*^k\Psi^s_{-k}}(p^\perp_{(k)}). \end{aligned}$$

The summand in the second summation becomes

$$\frac{\left\langle \nabla_{f_*^{1-k}\Psi^u_k} f_*\tilde{e}, f_*\tilde{e}\right\rangle}{\|f_*\tilde{e}\|^2} - \left\langle \nabla_{f_*^{-k}\Psi^u_k}\tilde{e}, \tilde{e}\right\rangle = \left\langle \beta_{f_*^{-k}\Psi^u_k} p'_{(k)}, \tilde{e}_1\right\rangle - \left\langle p'_{(k)}, \tilde{e}\right\rangle = U_{f_*^{-k}\Psi^u_k}(p'_{(k)}).$$

The lemma is proved by summing over k and lemma 12 (1). □

Characterizing p by induction

We give a new characterization, called the fast characterization, of the expansion formula, which allows even faster computation than directly using the expansion. Roughly speaking, we will show that p satisfies an inductive relation given by a renormalized second-order tangent equation, whose stability indicates that any non-covariant sequences satisfying this equation will eventually converge to p. This is in fact the non-intrusive formulation for the unstable contribution. In comparison to the original definition of non-intrusive shadowing, here we allow also second-order tangent solutions, and its stability is given by renormalizations, rather than subtracting homogeneous solutions.

Lemma 14 (inductive relation of p)**.** $p_1 := p \circ f = P^\perp\tilde{\beta}p$.

Remark. (1) Should we know the correct $p(x_0)$, we can solve all p_n inductively by $p_{n+1} = P^\perp\tilde{\beta}(p_n)$; this is more efficient than computing from the definition of p, because the most expensive operation, $\tilde{\beta}$, now only needs to operate once on p, instead of many times on each summand in the definition of p. (2) The oblique projection is replaced by the orthogonal projection, whose computation is easier and faster, in particular, it can be done with only u

many first order tangent solutions. (3) Hence, the inductive relation requires only u many first and second-order tangent solutions. (4) The subscript convention is explained in equation (2.1).

Proof. Write down the definition of p_1, relabel subscripts, we get

$$\begin{aligned} p_1 =& \sum_{k=0}^{\infty} \frac{P^{\perp} \nabla_{f_*^k \Psi^s_{1-k}} f_*^k \eta_* e_{1-k}}{\| f_*^k e_{1-k} \|} - \sum_{k=1}^{\infty} \nabla_{f_*^{-k} \Psi^u_{1+k}} \tilde{e}_1 \\ =& \sum_{k=-1}^{\infty} \frac{P^{\perp} \nabla_{f_*^{k+1} \Psi^s_{-k}} f_*^{k+1} \eta_* e_{-k}}{\| f_*^{k+1} e_{-k} \|} - \sum_{k=2}^{\infty} \nabla_{f_*^{1-k} \Psi^u_k} \tilde{e}_1 \end{aligned}$$

The induction of the first summation is achieved by substituting equation (6.10). For the second summation, denote $Y := f_*^{-k} \Psi^u_k$, we have

$$\nabla_{f_* Y} \tilde{e}_1 = \nabla_{f_* Y} \frac{f_* \tilde{e}}{\| f_* \tilde{e} \|} = \beta_Y p'_{(k)} + f_* Y \left(\frac{1}{\| f_* \tilde{e} \|} \right) f_* \tilde{e}.$$

Since $\nabla \tilde{e} \in \mathcal{D}_e^{\perp}$, $\nabla \tilde{e} = P^{\perp} \nabla \tilde{e}$; since $f_* \tilde{e} \in \wedge^u V_1^u$, $P^{\perp} f_* \tilde{e} = 0$. Hence,

$$\nabla_{f_* Y} \tilde{e}_1 = P^{\perp} \nabla_{f_* Y} \tilde{e}_1 = P^{\perp} \beta_Y p'_{(k)}.$$

Substitute into the expression for p_1, we have

$$\begin{aligned} p_1 =& \frac{\psi_1}{\| e_1 \|} P^{\perp} \nabla_{X_1^s} \eta_* e_1 + \sum_{k=0}^{\infty} P^{\perp} \beta_{f_*^k \Psi^s_{-k}} p^{\perp}_{(k)} + \psi_1 \nabla_{X_1^u} \tilde{e}_1 - \sum_{k=1}^{\infty} P^{\perp} \beta_{f_*^{-k} \Psi^u_k} p'_{(k)} \\ =& \frac{\psi_1}{\| e_1 \|} P^{\perp} \nabla_{X_1^s} \eta_* e_1 + \psi_1 \nabla_{X_1^u} \tilde{e}_1 + P^{\perp} \beta_{\tilde{v}} p \end{aligned}$$

To add the first two terms, use lemma 7 on the first term, and that $P^{\perp} e = 0$,

$$\begin{aligned} \frac{\psi}{\| e \|} P^{\perp} \nabla_{X^s} \eta_* e = \frac{\psi}{\| e \|} P^{\perp} \left(\nabla_e X - \nabla_{X^u} e \right) = \psi P^{\perp} \left(\nabla_{\tilde{e}} X - \frac{1}{\| e \|} \nabla_{X^u} e - e X^u (\frac{1}{\| e \|}) \right) \\ = \psi P^{\perp} (\nabla_{\tilde{e}} X - \nabla_{X^u} \tilde{e}) = \psi P^{\perp} \nabla_{\tilde{e}} X - \psi \nabla_{X^u} \tilde{e}. \end{aligned}$$

Hence $p_1 = \psi_1 P^{\perp} \nabla_{\tilde{e}_1} X_1 + P^{\perp} \beta_{\tilde{v}} p$, as claimed. □

Hence, we have found that the sequence $\{p_n\}_{n \geq 0}$ satisfies an inductive relation. This is a covariant sequence, that is, $p_n(x_0) = p(x_n)$. We may as well define a non-covariant sequence, r, only from the inductive relation of p. It turns out that r, the non-covariant version of p, convergences exponentially fast to the true p on any trajectory.

Lemma 15 (stability of renormalized second-order tangent equation)**.** *For any $x_0 \in K$, any $r_0 \in \mathcal{D}_e(x_0)$, define a non-covariant sequence, $\{r_n\}_{n\geq 0}$, by the inductive relation of p,*

$$r_n \in \mathcal{D}_e(x_n), \quad r_{n+1} := P^{\perp}\tilde{\beta} r_n.$$

Then $\{r_n\}_{n\geq 0}$ approximates its covariant counterpart, that is,

$$\lim_{n\to\infty} r_n - p_n(x_0) = \lim_{n\to\infty} P^{\perp}\tilde{\beta}^n r_0 - P^{\perp}\tilde{\beta}^n p(x_0) = 0.$$

Remark. (1) An easy choice of elements in $\mathcal{D}_e$ is zero. (2) The notation β^n means applying the inhomogeneous propagation operator n times. This is done similar to the pushforward operator, with $\tilde{v}$ and $\tilde{e}$ evaluated at suitable steps.

Proof. $P^{\perp}\tilde{\beta}^n r_0 - P^{\perp}\tilde{\beta}^n p = P^{\perp} f_*^n (r_0 - p)/\|f_*^n \tilde{e}\| = P^{\perp} f_*^n P^s (r_0 - p)/\|f_*^n \tilde{e}\| \leq C\lambda^{2n}$. □

Finally, we can prove the main theorem of this chapter, the fast characterization of the unstable contribution, which is theorem 6 stated in introduction section.

Remark. (of theorem 6) (1) In practice, a ρ-typical point x_0 can be found by running almost all trajectories starting from the attractor basin for some time. (2) In practice, the unstable subspace, V^u, can be obtained by pushing-forward u many randomly initiated vectors, because unstable vectors grow faster than stable ones; hence we can obtain correct $\{P^{\perp}(x_0)\}_{n\geq 0}$. (3) A more careful analysis should prove this theorem for almost all x_0 in the attractor basin, and almost all initial guess of $V^u(x_0)$.

Proof. (of theorem 6) By the same tail bound in lemma 6 and 10, we can show that p is continuous on the attractor K. By the continuity of $\tilde{U}$, $\tilde{U}(p)$ is continuous on K. Hence, by the ergodic theorem and lemma 13, we see that almost surely according to the SRB measure,

$$U.C.^W = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \tilde{U}(p_n)$$

By lemma 14, the definition of $\tilde{U}$, the orthogonal condition, we have

$$\begin{aligned} U.C.^W &= \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \tilde{U}((P^{\perp}\tilde{\beta})^n p) \\ &= \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \left\langle \tilde{\beta}(P^{\perp}\tilde{\beta})^n p, \tilde{e}_{n+1} \right\rangle - \left\langle (P^{\perp}\tilde{\beta})^n p, \tilde{e}_n \right\rangle = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \left\langle \tilde{\beta}(P^{\perp}\tilde{\beta})^n p, \tilde{e}_{n+1} \right\rangle \end{aligned}$$

Finally, apply lemma 15 to replace p by its non-covariant approximation, r. □

After our submission to arXiv, there appears a second version of the S3 algorithm [17]. Comparing to the old S3, the new S3 uses several techniques of this chapter, such as the renormalized second-order tangent equations, orthogonal projections, and an inductive formula for the unstable contribution. It also changes to a shadowing/unstable decomposition of the

linear response, similar to the one we use: we pointed out that the seemingly complicated 'v' in their writing is our $P^{\perp}v$. Currently, the new S3 works only for $u = 1$ and $\mathcal{M} = \mathbb{R}^M$; however, if using our tools in section 6.1, section 6.1, and in appendices, it might have a better chance to extend to general u and $\mathcal{M}$ than the old S3. It also lacks our fast characterization, which can make it faster and easier to implement. It also lacks the multi-segment treatment which we will show later.

6.3 Fast linear response algorithm

This section concerns the practical aspect of the algorithm. First, to further reduce computational cost and round-off error, we show that the renormalization only needs to be done intermittently. We will also write major equations in matrix notations, which are more suitable for computer programming. Then we give a procedure list of the algorithm. Finally, we give several remarks on the implementation of the algorithm.

Intermittent renormalization

This subsection explains how renormalization only needs to be done once after a segment of several steps. Here renormalization refers to

- The orthogonal projection, $P^{\perp}$, used in the fast characterization, defined in appendix 6.5.
- The rescaling in β, that is, dividing by the $\|f_*\tilde{e}\|$ factor in definition 4.
- Othonormalizing the basis vectors of V^u.

The third operation does not explicitly appear in the fast characterization, but a basis is implicitly required by the well-definedness of the hyper-cube e, and a good basis improves numerical performance. We first show that both orthogonal projection and rescaling can be done intermittently. Then, we show that all three renormalizing operations can be done together by a short formula using matrix notation.

The subscript convention for multiple segments, shown in figure 6.2, is similar to that of the non-intrusive shadowing algorithm [59, 58]. We divide a trajectory into small segments, each containing N steps. The α-th segment consists of step αN to $\alpha N + N$, where α runs from 0 to $A - 1$; notice that the last step of segment α is also the first step of segment $\alpha + 1$. We use double subscript, such as $x_{\alpha,n}$, to indicate the n-th step in the α-th segment, which is the $(\alpha N + n)$-th step in total. Note that for some quantities defined on each step, for example, $e_{\alpha,N} \neq e_{\alpha+1,0}$, since renormalization is performed at the interface across segments. Continuity across interfaces is true only for some quantities, such as shadowing directions v, $\tilde{v}$, and unit hyper-cube $\tilde{e}$. Later, we will defind some quantities on the α-th segment, such as C_α, d_α in equation (6.11), their subscripts are the same as the segment they are defined

on. For quantities to be defined at interfaces, such as $Q_\alpha, R_\alpha, b_\alpha$ in equation (6.12), their subscripts are the same as the total step number of the interface, divided by N.

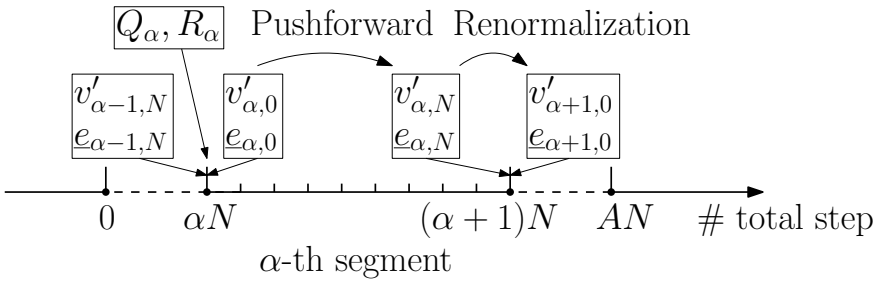


Figure 6.2: Subscript convention on multiple segments.

Lemma 16 (intermittent orthogonal projection)**.** *For any $\tilde{r}_{0,0} \in \mathcal{D}_e$, let*

$$\tilde{r}_{\alpha,n} := \tilde{\beta}\tilde{r}_{\alpha,n-1}, \quad \tilde{r}_{\alpha+1,0} := P^\perp \tilde{r}_{\alpha,N},$$

then almost surely according to Lebesgue measure,

$$U.C.^W = \lim_{A\to\infty} \frac{1}{NA} \sum_{\alpha=0}^{A-1} \langle \tilde{r}_{\alpha,N}, \tilde{e}_{\alpha,N} \rangle .$$

Proof. Denote $(P^\perp\tilde{\beta})^{\alpha N}\tilde{r}_{0,0}$ by r', by lemma 12, lemma 13, and the definition of the SRB measure, the unstable contribution from step αN to $\alpha N + N - 1$ is,

$$\begin{aligned}\sum_{n=0}^{N-1} \tilde{U}((P^\perp\tilde{\beta})^{\alpha N+n} r_{0,0}) &= \sum_{n=0}^{N-1} \tilde{U}((P^\perp\tilde{\beta})^n r') = \sum_{n=0}^{N-1} \tilde{U}(P^\perp\tilde{\beta}^n r') = \sum_{n=0}^{N-1} \tilde{U}(\tilde{\beta}^n r') \\ &= \sum_{n=0}^{N-1} \left\langle \tilde{\beta}^{n+1} r', \tilde{e}_{\alpha N+n+1} \right\rangle - \left\langle \tilde{\beta}^n r', \tilde{e}_{\alpha N+n} \right\rangle = \left\langle \tilde{\beta}^N r', \tilde{e}_{\alpha N+N} \right\rangle .\end{aligned}$$

Average over all steps and adopt the subscript convention to prove the lemma. □

Lemma 17 (intermittent rescaling)**.** *Let e be first-order tangent solutions,*

$$e_{0,0} = \tilde{e}_{0,0},\ e_{\alpha,n} := f_* e_{\alpha,n-1},\ e_{\alpha+1,0} := e_{\alpha,N}/\|e_{\alpha,N}\| = \tilde{e}_{\alpha+1,0};$$

For any $r_{0,0} \in \mathcal{D}_e$, let r be governed by the second-order tangent equation,

$$r_{\alpha,n} := f_* r_{\alpha,n-1} + (\nabla_{\tilde{v}_{\alpha,n-1}} f_*) e_{\alpha,n-1} + \psi_{\alpha,n} \nabla_{e_{\alpha,n}} X_{\alpha,n},\ \ r_{\alpha+1,0} := P^\perp r_{\alpha,N}/\|e_{\alpha,N}\|.$$

Then almost surely according to Lebesgue measure,

$$U.C.^W = \lim_{A\to\infty} \frac{1}{NA} \sum_{\alpha=0}^{A-1} \frac{\langle r_{\alpha,N}, e_{\alpha,N} \rangle}{\langle e_{\alpha,N}, e_{\alpha,N} \rangle}.$$

Proof. We prove by induction that if we choose $r_{0,0} = \tilde{r}_{0,0}$, then $r_{\alpha,n} = \|e_{\alpha,n}\|\tilde{r}_{\alpha,n}$. Within segment α, assuming we have this relation hold for $n-1$, then

$$r_{\alpha,n} = \|e_{\alpha,n-1}\|f_*\tilde{r}_{\alpha,n-1} + \|e_{\alpha,n-1}\|(\nabla_{\tilde{v}_{\alpha,n-1}}f_*)\tilde{e}_{\alpha,n-1} + \|e_{\alpha,n}\|\psi_{\alpha,n}\nabla_{\tilde{e}_{\alpha,n}}X_{\alpha,n} = \|e_{\alpha,n}\|\tilde{r}_{\alpha,n}.$$

Hence the relation also holds for n; it also holds across interfaces, since

$$r_{\alpha+1,0} = P^\perp r_{\alpha,N}/\|e_{\alpha,N}\| = P^\perp\tilde{r}_{\alpha,N} = \tilde{r}_{\alpha+1,0} = \tilde{r}_{\alpha+1,0}\|e_{\alpha+1,0}\|,$$

where $\|e_{\alpha+1,0}\| = 1$ by construction. Finally, substitute into lemma 16. □

Proposition 18 (intermittent renormalization)**.** *Neglecting the first two subscripts, α and n, let $e := \wedge_{i=1}^u e_i$, $r := \sum_i e_1 \wedge \cdots \wedge r_i \wedge \cdots \wedge e_u$. Denote matrices $\underline{e} := [e_1, \cdots, e_u]$, $\underline{r} := [r_1, \cdots, r_u]$. Then between segments, the renormalization in lemma 17 is realized by:*

$$\begin{aligned}
&\underline{e}_{\alpha,N} = Q_{\alpha+1}R_{\alpha+1}, \quad \underline{e}_{\alpha+1,0} = Q_{\alpha+1},\\
&\underline{r}^\perp_{\alpha,N} = \underline{r}_{\alpha,N} - Q_{\alpha+1}Q^T_{\alpha+1}\underline{r}_{\alpha,N}, \quad \underline{r}_{\alpha+1,0} = \underline{r}^\perp_{\alpha,N}R^{-1}_{\alpha+1}.
\end{aligned}$$

Here the first equation means to perform QR factorization, and $Q^T\underline{r} := [\langle Q_i, r_j\rangle]$ is a matrix. Use $\mathrm{Tr}(\cdot)$ *to denote trace of a matrix, the unstable contribution is,*

$$U.C.^W = \lim_{A\to\infty}\frac{1}{NA}\sum_{\alpha=0}^{A-1}\mathrm{Tr}\left(R^{-1}_{\alpha+1}Q^T_{\alpha+1}\underline{r}_{\alpha,N}\right).$$

Remark. (1) Rewriting r on the new basis does not change r as a u-vector, but it makes the numerical properties better. (2) Computing $\underline{e}$ via pushforwards and renormalization is an important part of the non-intrusive showing algorithm. (3) The pushforward relation inside a segment is the same as that in lemma 17.

Proof. The renormalization on e is due to the definition of QR factorization. For r, first substitute the QR factorization into the expression for projection,

$$\underline{r}^\perp_{\alpha,N} = \underline{r}_{\alpha,N} - \underline{e}_{\alpha,N}(\underline{e}^T_{\alpha,N}\underline{e}_{\alpha,N})^{-1}(\underline{e}^T_{\alpha,N}\underline{r}_{\alpha,N}) = \underline{r}_{\alpha,N} - Q_{\alpha+1}Q^T_{\alpha+1}\underline{r}_{\alpha,N}.$$

Use appendix 6.5 lemma 21 to rewrite $r^\perp_{\alpha,N}$ on to the new basis, $Q_{\alpha+1}$, then rescale,

$$\underline{r}_{\alpha+1,0} = \det(R_{\alpha+1})\,\underline{r}^\perp_{\alpha,N}R^{-1}_{\alpha+1}/\|e_{\alpha,N}\| = \underline{r}^\perp_{\alpha,N}R^{-1}_{\alpha+1},$$

since $\det(R_{\alpha+1}) = \|e_{\alpha,N}\|$ by definition of QR factorization. Finally, by lemma 25 in appendix 6.5,

$$U.C.^W \approx \frac{1}{NA}\sum_{\alpha=0}^{A-1}\frac{\langle r_{\alpha,N}, e_{\alpha,N}\rangle}{\langle e_{\alpha,N}, e_{\alpha,N}\rangle} = \frac{1}{NA}\sum_{\alpha=0}^{A-1}\sum_{i=1}^{u}e^i_{u,\alpha,N}r^\parallel_{\alpha,N,i}$$

where e_u^i is the covector of e_i in $\mathcal{V}^u$. Since $(e_u^i r_l^{\parallel})e_i = r_l^{\parallel}$, we have

$$(e_u^i r_l^{\parallel}) \langle e_i, e_j \rangle = \left\langle r_l^{\parallel}, e_j \right\rangle = \langle r_l, e_j \rangle .$$

This is a linear equation system, with solution

$$e_u^i r_l^{\parallel} = i - \text{th entry of the vector } (\underline{e}^T \underline{e})^{-1} (\underline{e}^T r_l).$$

Hence we can further write the expression of $U.C.^W$ in matrix notation,

$$U.C.^W \approx \frac{1}{NA} \sum_{\alpha=0}^{A-1} \mathrm{Tr} \left((\underline{e}_{\alpha,N}^T \underline{e}_{\alpha,N})^{-1} (\underline{e}_{\alpha,N}^T \underline{r}_{\alpha,N}) \right).$$

Substituting the QR factorization of $\underline{e}$, we have

$$(\underline{e}^T \underline{e})^{-1} (\underline{e}^T \underline{r}) = (R^T Q^T Q R)^{-1} R^T Q^T \underline{r} = R^{-1} Q^T \underline{r}.$$

□

Procedure list

This subsection gives a detailed procedure list of the fast linear response algorithm. It does not require differential geometry knowledge to understand this list when $\mathcal{M} = \mathbb{R}^M$, and corresponding simplifications are explained. All sequences in this subsection are non-covariant, which means that they are defined by some inductive relation. In particular, here the unstable vectors $\underline{e}$ and shadowing directions $v, \tilde{v}$ are non-covariant, which are very good approximations of their covariant counterparts. We still use r as the non-covariant version of p, since the approximation has just been established in this chapter. The subscript explanation is in figure 6.2.

1. Evolve the dynamical system for a sufficient number of steps before $n = 0$, so that x_0 is on the attractor at the beginning of our algorithm. Then, evolve the system from segment $\alpha = 0$ to $\alpha = A - 1$, each containing N steps, to obtain the trajectory,

$$x_{\alpha,n+1} = f(x_{\alpha,n}), \quad x_{\alpha+1,0} = x_{\alpha,N}.$$

2. Start with initial condition $v' = 0$ and $\tilde{v}' = 0$, and random initial conditions for each column in $\underline{e} := [e_1, \cdots, e_u]$. Then, repeat the following procedures for all α.

   a) From initial conditions, solve first-order tangent equations, α neglected,

$$\underline{e}_{n+1} = f_* \underline{e}_n, \quad v'_{n+1} = f_* v'_n + X_{n+1}, \quad \tilde{v}'_{n+1} = f_* \tilde{v}'_n + \Psi_{n+1}.$$

Here $X := \delta f \circ f^{-1} = (\partial f/\partial\gamma) \circ f^{-1}$, where γ is the parameter of the dynamical system; $\Psi_n := \psi_n X_n$, $\psi := \sum_{m=-W}^{W} \Phi \circ f^m$ for a large W, f_* is the pushforward operator. When $\mathcal{M} = \mathbb{R}^M$, f_* is the Jacobian matrix,

$$f_* = [\partial f^i/\partial z^j]_{ij}$$

where $[\cdot]_{ij}$ is the matrix with (i,j)-th entry given inside the bracket, f^i is the i-th component of f, z^j is the j-th coordinate of $\mathbb{R}^M$.

b) Compute and store the covariant matrix and the inner product,

$$\begin{aligned} C_\alpha &:= \sum_{n=0}^{N}{}' \underline{e}_{\alpha,n}^T \underline{e}_{\alpha,n} := \frac{1}{2}\underline{e}_{\alpha,0}^T \underline{e}_{\alpha,0} + \sum_{n=1}^{N-1} \underline{e}_{\alpha,n}^T \underline{e}_{\alpha,n} + \frac{1}{2}\underline{e}_{\alpha,N}^T \underline{e}_{\alpha,N}, \\ d_\alpha &:= \sum_{n=0}^{N}{}' \underline{e}_{\alpha,n}^T v'_{\alpha,n}, \quad \tilde{d}_\alpha := \sum_{n=0}^{N}{}' \underline{e}_{\alpha,n}^T \tilde{v}'_{\alpha,n} dt, \end{aligned} \tag{6.11}$$

where $\sum'$ is the summation with 1/2 weight at the two end points. Here $\underline{e}^T \underline{e} := [\langle e_i, e_j \rangle]$ is a matrix, same for $\underline{r}^T \underline{e}$ and $Q^T \underline{r}$ later.

c) At step N of segment α, orthonormalize $\underline{e}$ with a QR factorization, and compute

$$\underline{e}_{\alpha,N} = Q_{\alpha+1} R_{\alpha+1}, \quad b_{\alpha+1} = Q_{\alpha+1}^T v'_{\alpha,N}, \quad \tilde{b}_{\alpha+1} = Q_{\alpha+1}^T \tilde{v}'_{\alpha,N}. \tag{6.12}$$

d) Set initial conditions of the next segment,

$$\underline{e}_{\alpha+1,0} = Q_{\alpha+1}, \quad v'_{\alpha+1,0} = v'_{\alpha,N} - Q_{\alpha+1} b_{\alpha+1}, \quad \tilde{v}'_{\alpha+1,0} = \tilde{v}'_{\alpha,N} - Q_{\alpha+1} \tilde{b}_{\alpha+1}.$$

3. Solve the non-intrusive shadowing problem,

$$\begin{aligned} &\min_{\{a_\alpha\}} \sum_{\alpha=0}^{A-1} 2 d_\alpha^T a_\alpha + a_\alpha^T C_\alpha a_\alpha \\ &\text{s.t. } a_\alpha = R_\alpha a_{\alpha-1} + b_\alpha, \quad \alpha = 1, \ldots, A-1. \end{aligned}$$

A good way to solve this is via the Schur complement, as given in section 3.4 of [61]; similar tricks were used earlier by Blonigan in a different setting [7]. Solve the same problem again, with b replaced by $\tilde{b}$, for $\tilde{\alpha}$. Then compute v_α and $\tilde{v}_\alpha$,

$$v_\alpha = v'_\alpha + \underline{e}_\alpha a_\alpha, \quad \tilde{v}_\alpha = \tilde{v}'_\alpha + \underline{e}_\alpha \tilde{a}_\alpha.$$

4. Compute the shadowing contribution,

$$S.C. = \lim_{A\to\infty} \frac{1}{AN} \sum_{\alpha=0}^{A-1} \sum_{n=0}^{N}{}' v_{\alpha,n}(\Phi_{\alpha,n}),$$

where $v(\cdot)$ means to differentiate a function in the direction of v.

5. Denote $\underline{r} := [r_1, \cdots, r_u]$. Set initial condition $r_{0,i} = 0$ for all $1 \leq i \leq u$. Then repeat the following procedures for all α.

   a) From initial conditions, solve second-order tangent equations, α neglected,

   $$r_{n+1,i} = f_* r_{n,i} + (\nabla_{\tilde{v}_n} f_*) e_{n,i} + \psi_{n+1} \nabla_{e_{n+1,i}} X_{n+1} \ .$$

   Here $\nabla_{(\cdot)} f_*$ is the Riemannian connection of the pushforward operator defined in appendix 6.5. In $\mathbb{R}^M$, ∇f_* is the derivative of Jacobian, or the Hessian tensor, and

   $$\nabla_Y f_* = [Y(\partial f^i / \partial z^j)]_{ij}.$$

   To compute $\nabla_{e_{n+1,i}} X_{n+1}$, denote the coordinate at x_n and x_{n+1} by ζ and z, then

   $$\nabla_{e_{n+1,i}} X_{n+1} = \nabla_{f_* e_{n,i}} X_n \circ f = \nabla_{f_* e_{n,i}} \delta f_n = \nabla_{f_* e_{n,i}} \left(\delta f_n^j \frac{\partial}{\partial z^j} \right)$$
   $$= f_* e_{n,i} \left(\delta f_n^j \right) \frac{\partial}{\partial z^j} + \delta f_n^j \nabla_{f_* e_{n,i}} \frac{\partial}{\partial z^j} = e_{n,i}^l \frac{\partial \delta f_n^j}{\partial \zeta^l} \frac{\partial}{\partial z^j} + \delta f_n^j \nabla_{f_* e_{n,i}} \frac{\partial}{\partial z^j}.$$

   Here $\frac{\partial}{\partial z_j}$ is the j-th coordinate vector. In $\mathbb{R}^M$, both coordinates ζ and z are the canonical coordinate, both denoted by z, and $\nabla \frac{\partial}{\partial z_j}$ is zero, hence

   $$\nabla_{e_{n+1,i}} X_{n+1} = e_{n,i}^l \frac{\partial \delta f_n^j}{\partial z^l} \frac{\partial}{\partial z^j} = [e_{n,i}(\delta f_n^j)]_j.$$

   This is a vector at x_{n+1}, where $[\cdot]_j$ is a vector in $\mathbb{R}^M$ whose j-th entry is given in the bracket. The differentiation $e_{n,i}(\delta f_n^j)$ happens at x_n, when δf_n^j is a function in a neighborhood of x_n.

   b) To set initial conditions of the next segment, first orthogonally project,

   $$\underline{r}_{\alpha,N}^{\perp} = \underline{r}_{\alpha,N} - Q_{\alpha+1} Q_{\alpha+1}^T \underline{r}_{\alpha,N}.$$

   Then change basis and rescale,

   $$\underline{r}_{\alpha+1,0} = \underline{r}_{\alpha,N}^{\perp} R_{\alpha+1}^{-1}.$$

6. Let $\mathrm{Tr}(\cdot)$ be the trace of a matrix, compute the unstable contribution,

   $$U.C.^W = \lim_{A \to \infty} \frac{1}{NA} \sum_{\alpha=0}^{A-1} \mathrm{Tr} \left(R_{\alpha+1}^{-1} Q_{\alpha+1}^T \underline{r}_{\alpha,N} \right).$$

7. The linear response is

   $$\delta\rho(\Phi) = \lim_{W \to \infty} S.C. - U.C.^W.$$

Remarks on implementation

When the number of homogeneous tangent solutions computed, u', is strictly larger than u, the unstable contribution part of our algorithm may or may not work, depending on whether the renormalized second order tangent solutions has a meaningful average. This is different from the non-intrusive shadowing algorithm, which works for any $u' \geq u$. It remains to be investigated whether and how much error is incurred for using a large u, especially how the error relates to the spectrum of the Lyapunov exponents.

Fast linear response could converge even when the system slightly fails the uniform hyperbolicity assumption we used in the proof, because it does not compute oblique projections. This extra robustness is an upshot of non-intrusiveness, and was intentionally pursued during our algorithm design. The convergence of the fast linear response actually depends on the integrability of shadowing directions and renormalized second-order tangent solutions, which might be more abundant in applications than uniform hyperbolicity. Failure of fast linear response can be caused by a large region of homoclinic tangencies, where the stable and unstable directions are close to each other [65], such as the Henon map [31]. This situation is difficult for many algorithms and theoretical analysis for linear response, even for SRB measures.

We discuss how to choose the number of steps in each segment, N. Large N gives fewer segments for the same total number of steps, thus saving some computational cost. However, the major computational cost in the algorithm comes from computing first and second order tangent equations. Hence, the benefit for choosing a very large N is limited. Still, if we really want a large N, the upper bound is from the numerical stability of first and second order tangent equations. Notice that second order tangent equations are essentially first order tangent equations with a second-order inhomogeneous term. Hence, the limiting factors for large N are, the u many first-order homogeneous tangent solutions can not grow too large, or too parallel to each other, over one segment.

Depending on the sparsity of the problem, the main computational complexity of the fast linear response may come from different places. For the worst case, the Hessian tensor ∇f_* is dense, and contracting it with $\tilde{v}$ takes $O(M^3)$ operations. If further assuming that each entry in ∇f_* takes $O(1)$ operations, then it also takes $O(M^3)$ operations to get ∇f_*; in f_*, each entry depends on $O(M)$ variables, and it typically also takes $O(M^3)$ operations to get f_*. Hence, for the dense case, the cost for each step of fast linear response is dominated by obtaining $\nabla_{\tilde{v}} f_*$ and f_*, which only need to be done once, and cost is typically $O(M^3)$. On the other hand, for many problems from engineering, for example fluid mechanics and image processing, ∇f_* and f_* typically has only $O(M)$ entries, with each entry taking $O(1)$ operations to compute. Then the cost mainly comes from computing terms such as $f_* \underline{e}$, whose complexity is $O(uM)$ for each step, and can be computed faster via vectorized programming, as explained below. The sparse case is perhaps the more common case, especially because the notation $O(M)$ hints that we can freely change the dimension M, and this typically comes

from refining meshes of some fields, which typically gives the sparse case.

There are several places in the algorithm where we contract a high-order tensor with several one-dimensional vectors, which can be done more efficiently via the so-called 'vectorized' programming. More specifically, when contracting one tensor with several vectors, we should load the tensor into computer once, and inner-product with all vectors, instead of loading the tensor once for each vector: this saves computer time [58, 57]. Such vectorization can be used in the non-intrusive shadowing algorithm, where several first-order tangent solutions, v' and $\{e_i\}_{i=1}^u$, are multiplying with the same Jacobian matrix f_*, which is a two-dimensional tensor. Vectorization can also be used when solving second-order tangent equations, for the contraction between f_* and $\{r_i\}_{i=1}^u$, $\nabla_{\tilde{v}} f_*$ and $\{e_i\}_{i=1}^u$, and ∇X and $\{e_i\}_{i=1}^u$.

In non-intrusive shadowing, we can compute parts of the shadowing contribution from v' and each e_i, then add them up according to the coefficients a_i. The detailed formulas can be found in the finite-difference version of non-intrusive shadowing [61]. It allows us to gradually add or remove homogeneous solutions in batches, until all u unstable directions show up. It also reduces the cost for one more X to computing a new inhomogeneous tangent solution, which is typically marginal, since we have to load the Jacobian matrix anyway. Similarly, for the unstable contribution, we can solve the second-order tangent equation with different inhomogeneous terms, $\psi \nabla_{e_i} X$, $(\nabla_{v'} f_*) e_i$, and $(\nabla_{e_j} f_*) e_i$; then sum according to $\tilde{a}_j$ to get the entire unstable contribution. This requires tracking about u^2 many second-order tangent solutions, whereas the current procedure tracks only u many. However, for cases with many different Φ and X, or no *a priori* estimation of u, the decomposition allows recycling much of the computed data, thus is more efficient overall.

Finally, we give a very crude and formal estimation on the error and total cost of fast linear response, using the decorrelation step number W, the total number of steps $T := AN$, unstable dimension u, and system dimension M. First, the error for using finite W, $U.C. - U.C.^W$, is $O(\theta^W)$ for some $0 < \theta < 1$, which is the rate of decay of correlation. We make the simplifying assumption that, on any trajectory, for any function φ we care about, and for large enough N,

$$\sum_{n=1}^{N} \varphi(x_n) \sim O(\sqrt{N}),$$

where $\sim$ means approximately equal with large probability. This assumption is verified later in our numerical example, and it can be achieved by more basic assumptions, such as all summands are i.i.d.

Passing Φ to $\Phi - \rho(\Phi)$ makes $\rho(\Phi) = 0$; further notice that ψX is the inhomogeneous term for $\tilde{v}$, and that $\tilde{v}$ and ψ are the inhomogeneous terms for p and $\tilde{U}$; hence, $\tilde{U}(p) \sim \psi \sim O(\sqrt{W})$. Hence the error caused by averaging $\tilde{U}(p)$ on a finite trajectory of length T is $\sim O(\sqrt{W/T})$. Hence the total error for the unstable contribution, which is also the major error for the fast linear response, denoted by h, is

$$h \sim O(\theta^W) + O(\sqrt{W/T})$$

In practice we want the two errors to be roughly equal to each other, hence $T \sim \theta^{-2W} W$. By our discussion of the cost at each step, for typical problems with sparsity, the numerical complexity of fast linear response is

$$\text{Complexity of FLR} \sim O(uM\theta^{-2W} W).$$

In comparison, for the original linear response formula, the size of the integrand is $\sim \lambda_{max}^W$, where $\lambda_{max} > 1$ is the largest Lyapunov exponent. By similar arguments,

$$h \sim O(\theta^W) + O(\lambda_{max}^W/\sqrt{N'}), \quad T = N'W \sim \theta^{-2W}\lambda_{max}^{2W} W.$$

Here N' is the number of sample trajectories, each with W steps, and T is the total number of steps in all sample trajectories. Recall that only one first-order tangent equation and one f is computed at each step, hence, for the sparse case, the cost per step is $O(M)$, and in total,

$$\text{Complexity of ensemble} \sim O(M\theta^{-2W}\lambda_{max}^{2W} W),$$

Hence, for typical problems with sparsity, the numerical complexity of the original linear response formula can be about $O(\lambda_{max}^{2W}/u)$ times higher than fast linear response. Should we be interested in estimating the dense case, note that for a fair comparison with fast linear response, we should assume ∇f_* is dense, in which case computing f_* is $O(M^3)$, as we discussed. Moreover, as we discussed, several techniques can help further reducing the actual computing time of fast linear response.

6.4 A numerical example

This section illustrates the linear response algorithm on a numerical example, selected based on the following considerations. (1) u/M should be large, for the linear response algorithm to have an significant accuracy advantage over the more efficient non-intrusive shadowing algorithm. (2) $u \geq 2, s \geq 1$, and the unstable direction is unknown beforehand. This makes our manipulation of the unstable manifold necessary. It also helps to show that our algorithm works even when an unstable Lyapunov vector, the second one for example, does not foliate with smooth leaves [86]. (3) The region of homoclinic tangencies is small, to avoid explosion of shadowing directions. (4) M should be small, to save computer time.

Our dynamical system is a more unstable and more nonlinear version of the solenoid map. It is obtained by removing from the solenoid map one stable variable, adding one more expanding circle, and adding some nonlinear interactions between the stable and unstable variables:

$$\begin{aligned}
x_{n+1}^1 &= 0.05x_n^1 + 0.1\cos(8x_n^2) - 0.1\sin(5x_n^3) \\
x_{n+1}^2 &= 2x_n^2 + \gamma(1 + x_n^1)\sin(8x_n^2) \mod 2\pi \\
x_{n+1}^3 &= 3x_n^3 + \gamma(1 + x_n^1)\cos(2x_n^3) \mod 2\pi,
\end{aligned}$$

where the superscript labels the coordinates. The perturbation is caused by changing γ, that is, $\delta(\cdot) := \partial(\cdot)/\partial\gamma$. The instantaneous objective function is $\Phi(x) := x^1$.

This problem has $u = 2$ unstable directions. We set each segment to have $N = 20$ steps. The default setting, $A = 1000$ segments, $\gamma = 0.1$, and $W = 10$, is used unless otherwise noted. We implement the fast linear response algorithm, code available at `https://github.com/niangxiu/lra.git`. Figure 6.3 shows a typical trajectory. Figure 6.4 shows that the variance of the computed derivative is proportional to $A^{-0.5}$. Figure 6.5 shows that the bias in the averaged derivative decreases as W increases, but the variance increases like $W^{0.5}$, indicating that we should increase A together with W. This square-root trend verifies the assumption we made for the error estimation in section 6.3.

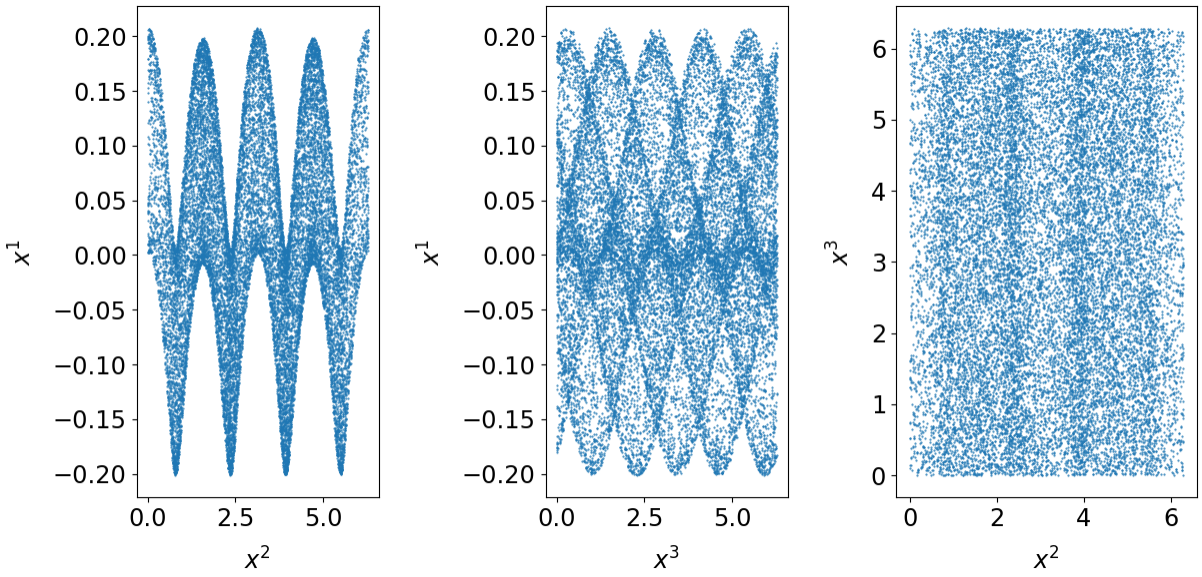


Figure 6.3: The empirical measure of a trajectory with default setting.

Finally, figure 6.6 shows that the derivative computed by the linear response algorithm correctly reflects the trend of the objective as γ changes, with only one short sample trajectory.

Our example is difficult for previous algorithms. Since u/M is large, the unstable contribution is significant, algorithms using approximations on the unstable contribution, such as non-intrusive shadowing, blended response, and S3, have large error. Since the attractor has zero Lebesgue measure, $1 < u < M$, and the unstable and stable directions are unknown beforehand, there is no easy trick to extend algorithms intended for $u = 1$ systems, such as the second version of S3, to our current problem.

Should we want to use algorithms based on the original linear response formula, then with $W = 10$, the magnitude of the integrand is

$$\|f_*^W X(\Phi)\| \sim \lambda_{max}^W \|X\| \|d\Phi\| \sim 3^{10} \times 5 \times 1 \sim 3 \times 10^5$$

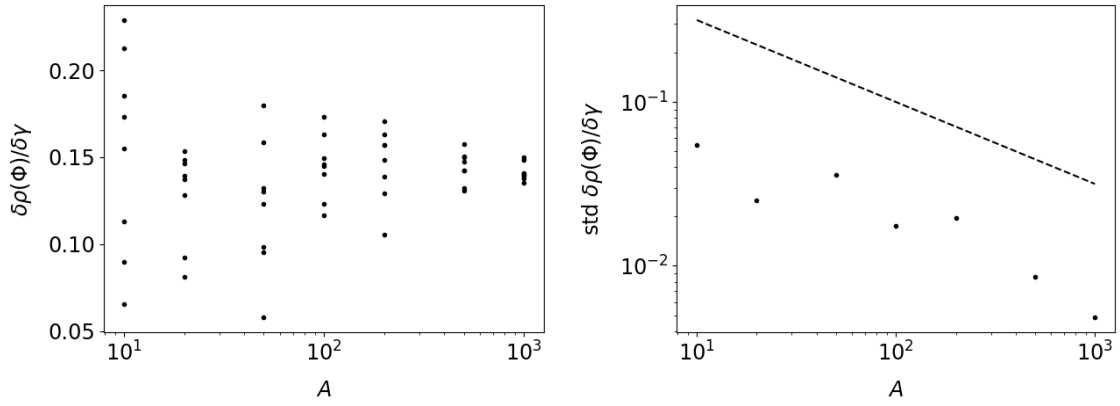


Figure 6.4: Effects of A. Left: derivatives from 8 independent computations for each A. Right: the sample standard deviation of the computed derivatives, where the dashed line is $A^{-0.5}$.

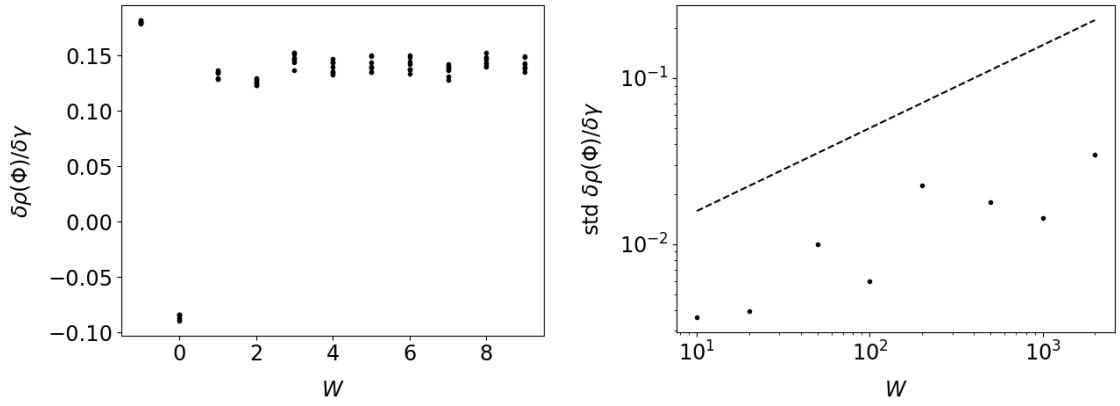


Figure 6.5: Effects of W. Left: derivatives computed by different W's. Right: standard deviation of derivatives, where the dashed line is $0.005W^{0.5}$.

In order to get the standard deviation within 5×10^{-3}, the number of sample trajectories required is about 3×10^{15}. Hence, the total number of steps computed is about 3×10^{16}, where each step contains one application of f and one step of first order tangent equation. In comparison, as shown in figure 6.4, the same setting requires running the fast linear response for 10^4 steps, where each step contains one application of f, 4 first order tangent, and 2 second order tangent equations. Say the cost per step is 3 times of that of original linear response, then, overall, fast linear response is about 10^{12} times faster than algorithms using the original linear response formula.

Finally, we compare with finite difference, or some functional regression method, which are

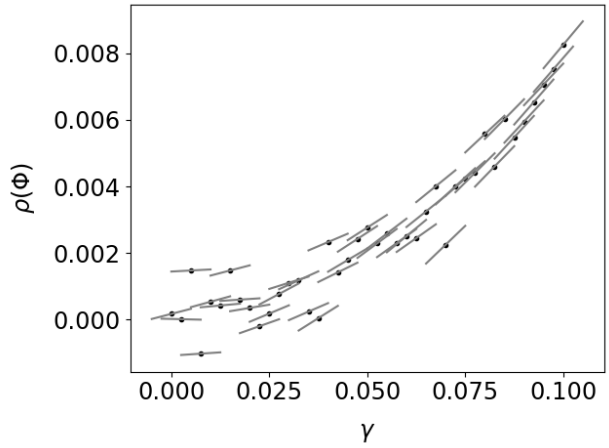


Figure 6.6: Averaged objectives and derivatives for different parameter γ. The grey lines are the derivatives computed by fast linear response.

typically regarded as approximate methods. In figure 6.6, because $\rho(\Phi)$ computed from finite trajectories has oscillations, roughly speaking, in order to reveal the correct trend between $\rho(\Phi)$ and γ, it takes about 40 data points with different γ. Although it is hard to quantify the error in such regression method, which would require imposing a probability distribution on regression models; we can still roughly say that, overall, the fast linear response is a few times faster than finite difference. This is very encouraging, because in the most simple non-chaotic situations, computing the derivative function is typically a few times faster than finite difference: fast linear response recovers such cost in chaotic systems. This hints that the fast linear response could perhaps be close to the best possible efficiency, in terms of exact algorithms for differentiating SRB measures on fractal attractors.

6.5 Appendix

Pushforward operators as tensors

In $\mathbb{R}^M$, the pushforward operator f_* is matrix. In particular, when differentiating a composition of several pushforwards, the Leibniz rule applies. In this section, we establish the Leibniz rule for general pushforward operators on Riemannian manifolds. To achieve this, we will define the pushforward operator on vectors as $(1,1)$-tensor, and define its Riemannian connection. Our definitions will comply with the Leibniz rule. Finally, we extend this Leibniz rule to u-vectors.

Definition 5 (f_*)**.** Let $f : \mathcal{M}_1 \to \mathcal{M}_2$ be a C^∞ diffeomorphism. Let $e \in \mathfrak{X}(\mathcal{M}_1)$ be a C^∞

vector field over $\mathcal{M}_1$; $\alpha \in \mathfrak{X}(\mathcal{M}_2)^*$ be a C^∞ 1-form over $\mathcal{M}_2$. Define

$$f_*(\alpha, e) := \alpha(f_* e) = e(f^* \alpha).$$

Let z_1, z_2 be coordinates on $\mathcal{M}_1$, $\mathcal{M}_2$ respectively. Written in coordinates, we have

$$f_* = \frac{\partial}{\partial z_2^j} f_i^j dz_1^i, \quad f_*(\alpha, e) = (\alpha \frac{\partial}{\partial z_2^j}) f_i^j (e dz_1^i),$$

where f_i^j is the Jacobian matrix under z_1 and z_2. Under our definition, f_* is a tensor field, in the sense that it is a C^∞-multilinear function $f_* : \mathfrak{X}(\mathcal{M}_1) \times \mathfrak{X}^*(\mathcal{M}_2) \to C^\infty(\mathcal{M}_2)$. We then define the Riemannian connection of this pushforward operator.

Let $\frac{\partial}{\partial q} \in T\mathcal{M}_1$, define

$$\nabla f_*(\frac{\partial}{\partial q}, \alpha, e) :=(f_* \frac{\partial}{\partial q})(\alpha f_* e) - (\nabla_{f_* \frac{\partial}{\partial q}} \alpha) f_* e - \alpha f_* \nabla_{\frac{\partial}{\partial q}} e = \alpha \nabla_{f_* \frac{\partial}{\partial q}} (f_* e) - \alpha f_* \nabla_{\frac{\partial}{\partial q}} e.$$

Under this definition, ∇f_* is also a tensor field, in the sense that it is a C^∞-multilinear function : $\mathfrak{X}(\mathcal{M}_1) \times \mathfrak{X}(\mathcal{M}_1) \times \mathfrak{X}^*(\mathcal{M}_2) \to C^\infty(\mathcal{M}_2)$. Notice that, similar to typical Riemannian connections, ∇f_* can be defined by the pointwise value of $\frac{\partial}{\partial q}$. Since α is a common factor, we may neglect it on both sides of the equation.

Definition 6 (∇f_*)**.** Define the Riemannian connection of pushforward operator, which is a tensor, $\nabla f_*(\frac{\partial}{\partial q}, e) : \mathfrak{X}(\mathcal{M}_1) \times \mathfrak{X}(\mathcal{M}_1) \to \mathfrak{X}^*(\mathcal{M}_2)$, as

$$\nabla f_*(\frac{\partial}{\partial q}, e) := (\nabla_{\frac{\partial}{\partial q}} f_*) e := \nabla_{f_* \frac{\partial}{\partial q}} (f_* e) - f_* \nabla_{\frac{\partial}{\partial q}} e.$$

To write ∇f_* in coordinates, by the coordinate form of f_*, we have

$$\nabla f_*(\frac{\partial}{\partial q}, e) = (\nabla_{f_* \frac{\partial}{\partial q}} \frac{\partial}{\partial z_2^j}) f_i^j (dz_1^i e) + (\frac{\partial}{\partial z_2^j})(df_i^j \frac{\partial}{\partial q})(dz_1^i e) + (\frac{\partial}{\partial z_2^j}) f_i^j (e \nabla_{\frac{\partial}{\partial q}} dz_1^i).$$

Lemma 19 (Leibniz rule for composition of pushforward operators)**.** *Let $g : \mathcal{M}_2 \to \mathcal{M}_3$ be a diffeomorphism. Then*

$$\nabla_{g_* f_* \frac{\partial}{\partial q}} (g_* f_* e) = (\nabla_{f_* \frac{\partial}{\partial q}} g_*) f_* e + g_* (\nabla_{\frac{\partial}{\partial q}} f_*) e + g_* f_* \nabla_{\frac{\partial}{\partial q}} e.$$

Remark. Besides the proof below, readers may find it consolidating to prove by writing everything in coordinates, which also helps us to check that the coordinate form is correct. When doing that, use the following relations to cancel or combine terms:

$$\frac{\partial}{\partial z_2^l} \nabla_{\frac{\partial}{\partial q}} dz_2^j + dz_2^j \nabla_{\frac{\partial}{\partial q}} \frac{\partial}{\partial z_2^l} = \frac{\partial}{\partial q} \delta_l^j = 0, \quad e \nabla_{\frac{\partial}{\partial q}} dz_1^i + dz_1^i \nabla_{\frac{\partial}{\partial q}} e = \frac{\partial}{\partial q} (e dz_1^i).$$

Proof. By definition 6,

$$\begin{aligned}\nabla_{g_* f_* \frac{\partial}{\partial q}}(g_* f_* e) = \nabla_{g_*(f_* \frac{\partial}{\partial q})}(g_*(f_* e)) &= (\nabla_{f_* \frac{\partial}{\partial q}} g_*)(f_* e) + g_* \nabla_{f_* \frac{\partial}{\partial q}}(f_* e) \\ &= (\nabla_{f_* \frac{\partial}{\partial q}} g_*)(f_* e) + g_*(\nabla_{\frac{\partial}{\partial q}} f_*)e + g_* f_* \nabla_{\frac{\partial}{\partial q}} e.\end{aligned}$$

□

Finally, we extend the Leibniz rule to the case where $e = e_1 \wedge \cdots \wedge e_u$ is a u-vector. Now $f_* e = f_* e_1 \wedge \cdots \wedge f_* e_u$, and

$$\nabla e = \sum_{i=1}^{u} e_1 \wedge \cdots \nabla e_i \wedge \cdots \wedge e_u, \quad \nabla(f_* e) = \sum_{i=1}^{u} f_* e_1 \wedge \cdots \nabla(f_* e_i) \wedge \cdots \wedge f_* e_u.$$

Define the Riemannian connection of pushforward operators on u-vectors,

$$(\nabla f_*)e := \nabla(f_* e) - f_* \nabla e = \sum_{i=1}^{u} f_* e_1 \wedge \cdots (\nabla f_*) e_i \wedge \cdots \wedge f_* e_u.$$

Lemma 20 (Leibniz rule for differentiating u-vectors)**.** *Let e be a C^∞ u-vector, then*

$$\begin{aligned}\nabla_{g_* f_* \frac{\partial}{\partial q}}(g_* f_* e) &= (\nabla_{f_* \frac{\partial}{\partial q}} g_*) f_* e + g_*(\nabla_{\frac{\partial}{\partial q}} f_*)e + g_* f_* \nabla_{\frac{\partial}{\partial q}} e, \\ \nabla_{f_*^k \frac{\partial}{\partial q}}(f_*^k e) &= \sum_{n=0}^{k-1} f_*^{k-n-1}(\nabla_{f_*^n \frac{\partial}{\partial q}} f_*) f_*^n e + f_*^k \nabla_{\frac{\partial}{\partial q}} e.\end{aligned}$$

Remark. (1) There is no need to add a subscript to ∇f_* to indicate steps, since the two vectors it applies to, $f_*^n e$ and $f_*^n \frac{\partial}{\partial q}$, already well-locate this tensor. (2) We use C^∞ in this section only because 'C^∞-multilinear' is a conventional terminology in differential geometry text. In fact, we only require the differentiation to be meaningful at the particular point and the particular direction. For example, the second equation of lemma applies to the rough u-vector field, e, defined in equation (6.2), differentiated in an unstable direction.

Proof. Inductively apply that $\nabla(f_* e) = (\nabla f_*)e + f_* \nabla e$. □

Derivative-like u-vectors

Let there be a u-vector field $e := e_1 \wedge \cdots \wedge e_u$, smoothly defined along a direction $\frac{\partial}{\partial q}$. The derivative is $\nabla_{\frac{\partial}{\partial q}} e = \sum_i e_1 \wedge \cdots \nabla_{\frac{\partial}{\partial q}} e_i \cdots \wedge e_u$, that is, one entry in each summand is typically not in the span of $\{e_i\}_{i=1}^u$. However, it is not trivial that the summation of derivatives still has this form, especially when the basis of e and directions $\frac{\partial}{\partial q}$ in each summand are different. Motivated by this, we define a subspace of u-vectors which looks like these derivatives, and show some related properties.

Definition 7. At point x, the collection of derivative-like u-vectors of $e = \wedge_{i=1}^{u} e_i$, written on basis $\{e_i\}_{i=1}^{u}$, is defined as

$$\mathcal{D}_e := \{p \in \wedge^u T_x\mathcal{M} : p = \sum_i e_1 \wedge \cdots \wedge p_i \wedge \cdots \wedge e_u,\ p_i \in T_x\mathcal{M}\}.$$

In this chapter, the basis $\{e_i\}_{i=1}^{u}$ is typically the basis of the unstable subspace. Letting $\{e_i\}_{i=1}^{M}$ be a full basis of $T_x\mathcal{M}$ whose first u vectors spans V^u, then by decomposing p onto the basis of $\wedge^u T_x\mathcal{M}$, we can see that $\mathcal{D}_e$ is the direct sum

$$\mathcal{D}_e = \text{span}\{e\} \oplus \sum_{i=1}^{u} \sum_{j>u} \text{span}\{e_1 \wedge \cdots \wedge e_{i-1} \wedge e_{i+1} \wedge \cdots \wedge e_u \wedge e_j\}. \tag{6.13}$$

Hence, under a given basis, p_i is unique modulo $\text{span}\{e_i\}_{i=1}^{u}$.

So long as $\text{span}\{e_1, \cdots, e_u\}$ is the same, $\mathcal{D}_e$ as a subspace is independent of the selection of basis. To see this, let $\{e_i'\}_{i=1}^{M}$ be another full basis such that $\text{span}\{e_1, \cdots, e_u\} = \text{span}\{e_1', \cdots, e_u'\} = V^u$, and $\mathcal{D}_{e'}$ be the corresponding subspace. Write each term in equation (6.13) by the other basis. we can see that

$$\begin{aligned} &\text{span}\{e_1 \wedge \cdots \wedge e_{i-1} \wedge e_{i+1} \wedge \cdots \wedge e_u \wedge e_j\} \\ \subset &\text{span}\{e'\} \oplus \sum_{i=1}^{u} \sum_{j>u} \text{span}\{e_1' \wedge \cdots \wedge e_{i-1}' \wedge e_{i+1}' \wedge \cdots \wedge e_u' \wedge e_j'\}. \end{aligned}$$

Hence, $\mathcal{D}_e \subset \mathcal{D}_{e'}$. By symmetry, $\mathcal{D}_e = \mathcal{D}_{e'}$.

The next step is to consider how to express derivative-like u-vectors under a new basis of the same span. Should p indeed be the derivative of a u-vector field, there is a formula for changing to a new basis which is a constant linear combination of the old basis. We will show that the same formula is true when p is only derivative-like.

Lemma 21 (change of basis formula)**.** *If $\{e_j'\}_{j=1}^{u}$ is another basis of V^u such that $e_j' = a_j^i e_i$, and let $p_k' := a_k^i p_i$, then*

$$\sum_k e_1' \wedge \cdots \wedge p_k' \wedge \cdots \wedge e_u' = \det(a_i^j) \sum_k e_1 \wedge \cdots \wedge p_k \wedge \cdots \wedge e_u.$$

Notice that typically $p_k \notin V^u$.

Remark. (1) If $p_k = \nabla_Y e_k$ for some vector Y, then this lemma is just applying ∇_Y on $e' = \det(a_i^j)e$. (2) We use this lemma for proving lemma 18, where p is a summation of derivatives. For this particular case, we may as well first prove the lemma for each summand, then apply linearity to obtain the lemma for the summation. However, here we give a more general and algebraic proof, which does not rely on the fact that p is a derivative or a summation of derivatives.

Proof. We can reorder e'_i so that for all i, $S_i := \{e'_1, \cdots, e'_i, e_{i+1}, \cdots, e_u\}$ is an independent set of vectors . Changing basis from S_0 to S_u can be achieved by sequentially changing from S_i to S_{i+1}, where only one vector is changed in each step. Hence, by induction, it suffices to show that the lemma is true for any one step, for example, the first step. Hence, it suffices to prove for the case where $e'_1 = a^i e_i$, and $e'_j = e_j$ for all $j \geq 2$. Now $p'_1 = a^i p_i$, $p'_j = p_j$ for all $j \geq 2$. the left hand side is

$$\begin{aligned} LHS = \sum_{k=1}^{u} e'_1 \wedge \cdots \wedge p'_k \wedge \cdots \wedge e'_u = p'_1 \wedge e'_2 \wedge \cdots \wedge e'_u + \sum_{k\geq 2}^{u} e'_1 \wedge \cdots \wedge p'_k \wedge \cdots \wedge e'_u \\ = \sum_i a^i p_i \wedge e_2 \wedge \cdots \wedge e_u + \sum_{k\geq 2} \sum_i a^i e_i \wedge e_2 \wedge \cdots \wedge p_k \wedge \cdots \wedge e_u \\ = \sum_i a^i p_i \wedge e_2 \wedge \cdots \wedge e_u + a^1 \sum_{k\geq 2} e_1 \wedge \cdots p_k \cdots \wedge e_u + \sum_{k\geq 2} \sum_{i\geq 2} a^i e_i \wedge e_2 \cdots \wedge p_k \wedge \cdots \wedge e_u \end{aligned}$$

In the last summation, notice that an exterior product vanishes if e_i appears twice. Hence,

$$LHS = a^1 \sum_{k\geq 2} e_1 \cdots p_k \cdots e_u + \left(\sum_i a^i p_i \wedge e_2 \cdots e_u + \sum_{k\geq 2} a^k e_k \wedge e_2 \cdots p_k \cdots e_u \right)$$

Comparing the two summations in the parenthesis, notice that interchanging the position of p_k and e_k in an exterior product changes the sign, hence all terms cancel, except the term with $i = 1$, and

$$\begin{aligned} LHS = a^1 \sum_{k\geq 2} e_1 \wedge \cdots p_k \wedge \cdots \wedge e_u + \sum_{i=1} a^i p_i \wedge e_2 \wedge \cdots \wedge e_u \\ = a^1 \sum_k e_1 \wedge \cdots \wedge p_k \cdots \wedge e_u = a^1 p. \end{aligned}$$

Since a^1 is the determinant of our current transformation matrix, we have proved the lemma for one step. The lemma is proved by induction. □

Projection operators of derivative-like u-vectors

For derivative-like u-vectors defined in appendix 6.5, only one entry is not in V^u in each summand, hence, we can extend the definition of projection operators for one-vectors to $\mathcal{D}_e$, by applying the projection on the one exceptional entry in each summand. In this section, $e \in \wedge^u V^u$ by default.

Definition 8. The projection operator P on $p \in \mathcal{D}_e$ is

$$Pp = \sum_i e_1 \wedge \cdots \wedge Pp_i \wedge \cdots \wedge e_u,$$

where P on the right side is the projection operator on one-vectors.

The projection operators used in this chapter are P^u, P^s, $P^{\|}$, and $P^{\perp}$. The first two operators are oblique projections along stable or unstable subspace to the unstable or stable subspace, respectively. The last two operators are orthogonal projection projections onto the unstable subspace and its orthogonal complement. These operators, applied on one-vectors, are illustrated in figure 6.7. Notice that computing P^u and P^s both require both V^u and V^s, whereas computing $P^{\|}$ and $P^{\perp}$ only require V^u, thus are faster. For both single and u-vectors, denote

$$p^u := P^u p,\ p^s := P^s p;\ p^{\|} := P^{\|} p,\ p^{\perp} := P^{\perp} p.$$

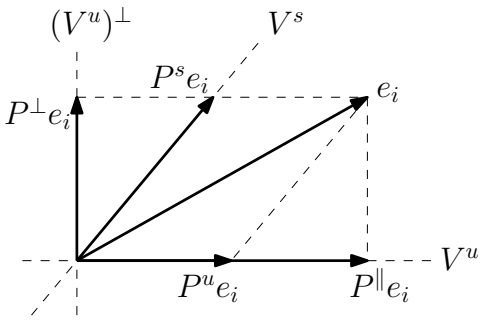


Figure 6.7: P^u, P^s, and $P^{\|}, P^{\perp}$ applied on e_i.

For fixed V^u, $P^{\|}$ and $P^{\perp}$ do not depend on the choice of basis. To see this, notice that

$$\mathcal{D}_e^{\perp} := \sum_{i=1}^{u} \sum_{j>u} \operatorname{span}\{e_1 \wedge \cdots \wedge e_{i-1} \wedge e_{i+1} \wedge \cdots \wedge e_u \wedge e_j\}.$$

is the same so long as $e_{u+1}, \cdots, e_M$ is a basis of $(V^u)^{\perp}$; also notice that $P^{\|}$ and $P^{\perp}$ are operators taking components in the decomposition $\mathcal{D}_e = \operatorname{span}\{e\} \oplus \mathcal{D}_e^{\perp}$. Similarly, if both V^u and V^s are fixed, P^u and P^s do not depend on the choice of basis.

Lemma 22 (composing projections with addition)**.** *For $p, p' \in \mathcal{D}_e$,*

$$P(p + p') = Pp + Pp'.$$

Lemma 23 (composing projection operators)**.**

$$P^{\|}P^u = P^u, P^u P^{\|} = P^{\|};\ P^{\perp}P^s = P^{\perp}, P^s P^{\perp} = P^s;\ P^{\perp}P^u = P^s P^{\|} = 0.$$

Remark. Notice that typically $P^u P^{\perp} \neq 0$, $P^{\|}P^s \neq 0$.

Lemma 24 (composing projections with pushforwards)**.**

$$f_* P^u = P^u f_* = P^u f_* P^u,\ f_* P^s = P^s f_* = P^s f_* P^s;$$
$$f_* P^{\|} = P^{\|} f_* P^{\|},\ P^{\perp} f_* = P^{\perp} f_* P^{\perp}.$$

Here the projection P's are evaluated at suitable locations.

Proof. The first three equalities are because of the invariance of stable and unstable subspace. The last equality is because

$$P^{\perp} f_* = P^{\perp} P^s f_* = P^{\perp} P^s f_* P^s = P^{\perp} P^s f_* P^s P^{\perp} = P^{\perp} P^s f_* P^{\perp} = P^{\perp} f_* P^{\perp}.$$

□

Lemma 25 (expressing $P^{\parallel}$ by inner products)**.** *For any $p \in \mathcal{D}_e$,*

$$P^{\parallel} p = \langle p, e \rangle \frac{e}{\|e\|^2}; \quad P^{\perp} p = p - \langle p, e \rangle \frac{e}{\|e\|^2}.$$

Proof. For the first equation, since both sides are in $\wedge^u V^u$, which is a one-dimensional subspace, it suffices to prove the equation taken inner product with e, that is,

$$\left\langle P^{\parallel} p, e \right\rangle = \langle p, e \rangle \frac{\langle e, e \rangle}{\|e\|^2} = \langle p, e \rangle .$$

Further adding $\left\langle P^{\perp} p, e \right\rangle = 0$ to the left proves this equality. The second equality is because $P^{\perp} + P^{\parallel} = Id$, hence $P^{\perp} p = p - P^{\parallel} p$. □

Bibliography

[1] Rafail V Abramov and Andrew J Majda. "Blended response algorithms for linear fluctuation-dissipation for complex nonlinear dynamical systems". In: *Nonlinearity* 20.12 (2007), pp. 2793–2821.

[2] D. V. Anosov. "Geodesic flows on closed Riemannian manifolds of negative curvature". In: *Trudy Mat. Inst. Steklov.* 90 (1967), pp. 1–235. URL: http://mi.mathnet.ru/tm2795.

[3] Wael Bahsoun et al. "A rigorous computational approach to linear response". In: *Nonlinearity* 31.3 (2018), pp. 1073–1109. ISSN: 13616544. DOI: 10.1088/1361-6544/aa9a88. arXiv: 1506.08661.

[4] Viviane Baladi. "Linear response, or else". In: *ICM Seoul 2014 Proceedings* 3 (2014), pp. 525–545. arXiv: 1408.2937. URL: http://arxiv.org/abs/1408.2937.

[5] Giancarlo Benettin et al. "Lyapunov Characteristic Exponents for smooth dynamical systems and for hamiltonian systems; A method for computing all of them. Part 2: Numerical application". In: *Meccanica* 15.1 (1980), pp. 21–30. ISSN: 1572-9648. DOI: 10.1007/BF02128237. URL: http://dx.doi.org/10.1007/BF02128237.

[6] Patrick Blonigan et al. "Toward a chaotic adjoint for LES". In: *Center for Turbulence Research, Proceedings of the Summer Program*. Stanford, 2016, pp. 385–394. arXiv: arXiv:1702.06809v1.

[7] Patrick J Blonigan. "Adjoint sensitivity analysis of chaotic dynamical systems with non-intrusive least squares shadowing". In: *Journal of Computational Physics* 348 (2017), pp. 803–826. ISSN: 10902716. DOI: 10.1016/j.jcp.2017.08.002.

[8] Patrick J. Blonigan. "Least Squares Shadowing for Sensitivity Analysis of Large Chaotic Systems and Fluid Flows". Ph.D thesis. MIT, 2016.

[9] Patrick J. Blonigan and Qiqi Wang. "Multiple shooting shadowing for sensitivity analysis of chaotic dynamical systems". In: *Journal of Computational Physics* 354 (2018), pp. 447–475. DOI: 10.2514/6.2015-1534. arXiv: 1704.02047. URL: http://arc.aiaa.org/doi/10.2514/6.2015-1534.

[10] Jo Bovy. *Lyapunov exponents and strange attractors in discrete and continuous dynamical systems*. Tech. rep. KU Leuven University, Theoretical Physics Project, 2004.

[11] Rufus Bowen. *Equilibrium States and the Ergodic Theory of Anosov Diffeomorphisms Second revised edition*. Springer, 2008, p. 83. ISBN: 9783540776055.

[12] Rufus Bowen. "Markov Partitions for Axiom A Diffeomorphisms". In: *American Journal of Mathematics* 92.3 (1970), pp. 725–747. ISSN: 00029327. DOI: 10.2307/2373370.

[13] Rufus Bowen and David Ruelle. "The ergodic theory of Axiom A flows". In: *Inventiones Mathematicae* 29.3 (1975), pp. 181–202. ISSN: 00209910. DOI: 10.1007/BF01389848.

[14] Guillaume A Brès et al. "Unstructured Large-Eddy Simulations of Supersonic Jets". In: *AIAA Journal* 55.4 (Jan. 2017), pp. 1164–1184. ISSN: 0001-1452. DOI: 10.2514/1.J055084. URL: https://doi.org/10.2514/1.J055084.

[15] Giulio Casati, Giorgio Comparin, and Italo Guarneri. "Decay of correlations in certain hyperbolic systems". In: *Physical Review A* 26.1 (1982), pp. 717–719. ISSN: 10502947. DOI: 10.1103/PhysRevA.26.717.

[16] Nisha Chandramoorthy and Qiqi Wang. "A computable realization of Ruelle's formula for linear response of statistics in chaotic systems". In: *arXiv:2002.04117* (2020). arXiv: 2002.04117. URL: http://arxiv.org/abs/2002.04117.

[17] Nisha Chandramoorthy and Qiqi Wang. "Efficient computation of linear response of chaotic attractors with one-dimensional unstable manifolds". In: *arXiv:2103.08816* (2021), pp. 1–40. arXiv: 2103.08816. URL: http://arxiv.org/abs/2103.08816.

[18] Nisha Chandramoorthy et al. "Algorithmic differentiation of shadowing sensitivities in chaotic systems". In: *SIAM Workshop on Combinatorial Scientific Computing*. Bergen, Norway, 2018, pp. 1–18.

[19] Nisha Chandramoorthy et al. "An Analysis of the Ensemble Adjoint Approach to Sensitivity Analysis in Chaotic Systems". In: *23rd AIAA Computational Fluid Dynamics Conference, AIAA AVIATION Forum (AIAA 2017-3799)*. AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, 2017, pp. 1–11. DOI: doi:10.2514/6.2017-3799. URL: https://doi.org/10.2514/6.2017-3799.

[20] Mario Chater. "Least Squares Shadowing for Sensitivity Analysis of Chaotic Dynamical Systems". Master thesis. MIT, 2016.

[21] P. Collet and J.-P. Eckmann. "Liapunov Multipliers and Decay of Correlations in Dynamical Systems". In: *Journal of Statistical Physics* 115 (2004), pp. 217–254.

[22] Tim Colonius, Sanjiva K Lele, and Parviz Moin. "Boundary conditions for direct computation of aerodynamic Sound Generation". In: *AIAA Journal* 31.9 (1993), pp. 1574–1582.

[23] P. Constantin et al. "Determining modes and fractal dimension of turbulent flows". In: *Journal of Fluid Mechanics* 150 (1985), pp. 427–440. ISSN: 14697645. DOI: 10.1017/S0022112085000209.

[24] James W. Cooley and John W. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series". In: *Mathematics of Computation* 19.90 (1965), p. 297. ISSN: 00255718. DOI: 10.2307/2003354.

[25] Richard Courant, Kurt Friedrichs, and H Lewy. "On the partial difference equations of mathematical physics". In: *IBM Journal of Research and Development* 11.2 (1967), pp. 215–234. URL: http://www.archive.org/details/onpartialdiffere00cour.

[26] Lesley De Cruz et al. "Exploring the Lyapunov instability properties of high-dimensional atmospheric and climate models". In: *Nonlinear Processes in Geophysics* 25.2 (2018), pp. 387–412. ISSN: 16077946. DOI: 10.5194/npg-25-387-2018. arXiv: arXiv:1712.08242v2.

[27] Dmitry Dolgopyat. "On differentiability of SRB states for partially hyperbolic systems". In: *Inventiones Mathematicae* 155.2 (2004), pp. 389–449. ISSN: 00209910. DOI: 10.1007/s00222-003-0324-5.

[28] G L Eyink, T W N Haine, and D J Lea. "Ruelle's linear response formula, ensemble adjoint schemes and Lévy flights". In: *Nonlinearity* 17.5 (2004), pp. 1867–1889.

[29] P. Fernandez and Q. Wang. "Lyapunov spectrum of the separated flow around the NACA 0012 airfoil and its dependence on numerical discretization". In: *Journal of Computational Physics* 350 (2017), pp. 453–469. ISSN: 10902716. DOI: 10.1016/j.jcp.2017.08.056. arXiv: 1612.07409.

[30] Pablo Fernandez, Ngoc-Cuong Nguyen, and Jaime Peraire. "Subgrid-scale modeling and implicit numerical dissipation in DG-based Large-Eddy Simulation". In: *23rd AIAA Computational Fluid Dynamics Conference, AIAA AVIATION Forum, (AIAA 2017-3951)*. Denver, Colorado, 2017, pp. 1–23. ISBN: 978-1-62410-506-7. DOI: 10.2514/6.2017-3951. URL: https://arc.aiaa.org/doi/10.2514/6.2017-3951.

[31] John Erik Fornœss and Estela A. Gavosto. "Existence of generic homoclinic tangencies for henon mappings". In: *The Journal of Geometric Analysis* 2.5 (1992), pp. 429–444. ISSN: 10506926. DOI: 10.1007/BF02921300.

[32] G Gallavotti. "Chaotic Hypothesis". In: *Scholarpedia 3(1):5906* 3.1 (2008), pp. 8–11. ISSN: 1941-6016. DOI: 10.4249/scholarpedia.5906.

[33] G. Gallavotti and E. G D Cohen. "Dynamical ensembles in stationary states". In: *Journal of Statistical Physics* 80.5-6 (1995), pp. 931–970. ISSN: 00224715. DOI: 10.1007/BF02179860. arXiv: 9501015 [chao-dyn].

[34] E. Garnier, N. Adams, and P. Sagaut. *Large Eddy Simulation for Compressible Flows*. Scientific Computation. Springer, 2009. ISBN: 978-90-481-2818-1. DOI: 10.1007/978-90-481-2819-8. URL: http://link.springer.com/10.1007/978-90-481-2819-8.

[35] Francesco Ginelli et al. "Covariant Lyapunov vectors". In: *Journal of Physics A: Mathematical and Theoretical* 46.25 (2013), p. 254005. ISSN: 1751-8121. DOI: 10.1088/1751-8113/46/25/254005. arXiv: arXiv:1212.3961v1. URL: http://iopscience.iop.org/1751-8121/46/25/254005%5Cnhttp://iopscience.iop.org/1751-8121/46/25/254005/pdf/1751-8121_46_25_254005.pdf.

[36] L. Greengard and V. Rokhlin. "A fast algorithm for particle simulations". In: *Journal of Computational Physics* 135.2 (1987), pp. 280–292. ISSN: 00219991. DOI: 10.1006/jcph.1997.5706.

[37] Leslie Greengard and John Strain. "The Fast Gauss Transform". In: *SIAM Journal on Scientific and Statistical Computing* 12.1 (1991), pp. 79–94. ISSN: 0196-5204. DOI: 10.1137/0912004.

[38] Andrey Gritsun and Valerio Lucarini. "Fluctuations, response, and resonances in a simple atmospheric model". In: *Physica D: Nonlinear Phenomena* 349 (2017), pp. 62–76. ISSN: 01672789. DOI: 10.1016/j.physd.2017.02.015. arXiv: 1604.04386. URL: http://dx.doi.org/10.1016/j.physd.2017.02.015.

[39] Stefanie Günther, Nicolas R. Gauger, and Qiqi Wang. "A framework for simultaneous aerodynamic design optimization in the presence of chaos". In: *Journal of Computational Physics* 328 (2017), pp. 387–398. ISSN: 10902716. DOI: 10.1016/j.jcp.2016.10.043.

[40] Morris W. Hirsch, Charles C. Pugh, and Michael Shub. *Invariant Manifolds*. Lecture Notes in Mathematics. Berlin: Springer, 1977, p. 150. DOI: 10.1142/9789812798749_0005.

[41] Francisco Huhn and Luca Magri. "Stability, sensitivity and optimisation of chaotic acoustic oscillations". In: *Journal of Fluid Mechanics* 882 (2020). ISSN: 0022-1120. DOI: 10.1017/jfm.2019.828. arXiv: 1909.12979.

[42] Miaohua Jiang. "Differentiating potential functions of SRB measures on hyperbolic attractors". In: *Ergodic Theory and Dynamical Systems* 32.4 (2012), pp. 1350–1369. ISSN: 01433857. DOI: 10.1017/S0143385711000241.

[43] A B Katok and B A Hasselblatt. *Introduction to the Modern Theory of Dynamical Systems*. Vol. 54. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1997, pp. 1–824. ISBN: 9780521575577. URL: http://books.google.ie/books?id=9nL7ZX8Djp4C.

[44] Pavel V. Kuptsov and Ulrich Parlitz. "Theory and computation of covariant lyapunov vectors". In: *Journal of Nonlinear Science* 22.5 (2012), pp. 727–762. ISSN: 09388974. DOI: 10.1007/s00332-012-9126-5. arXiv: 1105.5228.

[45] L. D. LANDAU and E. M. LIFSHITZ. *Fluid mechanics*. Addison-Wesley, 1959. ISBN: 0891166718.

[46] Davide Lasagna, Ati Sharma, and Johan Meyers. "Periodic shadowing sensitivity analysis of chaotic systems". In: *Journal of Computational Physics* 391 (2019), pp. 119–141. ISSN: 10902716. DOI: 10.1016/j.jcp.2019.04.021. arXiv: 1806.02077.

[47] D J Lea, M R Allen, and T W N Haine. "Sensitivity analysis of the climate of a chaotic system". In: *Tellus Series a-Dynamic Meteorology and Oceanography* 52.5 (2000), pp. 523–532. ISSN: 1477870X. DOI: 10.1256/qj.01.180.

[48] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002. ISBN: 9780511791253. DOI: 10.1017/CBO9780511791253. arXiv: 9809069v1 [arXiv:gr-qc]. URL: http://ebooks.cambridge.org/ref/id/CBO9780511791253.

[49] Valerio Lucarini, Francesco Ragone, and Frank Lunkeit. "Predicting Climate Change Using Response Theory: Global Averages and Spatial Patterns". In: *Journal of Statistical Physics* 166.3-4 (2017), pp. 1036–1064. ISSN: 00224715. DOI: 10.1007/s10955-016-1506-z. arXiv: 1512.06542.

[50] Colin Barr Macdonald. "Constructing High-Order Runge-Kutta Methods with Embedded Strong-Stability-Preserving Pairs". PhD thesis. Simon Fraser University, 2003.

[51] Rajat Mittal and S. Balachandar. "Direct Numerical Simulation of Flow Past Elliptic Cylinders". In: *Journal of Computational Physics* 367.124 (1996), pp. 351–367. URL: http://linkinghub.elsevier.com/retrieve/pii/S0021999196900650.

[52] Chin-Hoh Moeng and John C. Wyngaard. "Evaluation of Turbulent Transport and Dissipation Closures in Second-Order Modeling". In: *Journal of the Atmospheric Sciences* 46.14 (1989), pp. 2311–2330. ISSN: 0022-4928. DOI: 10.1175/1520-0469(1989)046<2311:EOTTAD>2.0.CO;2. URL: http://journals.ametsoc.org/doi/abs/10.1175/1520-0469%281989%29046%3C2311%3AEOTTAD%3E2.0.CO%3B2.

[53] David D. Morrison, James D. Riley, and John F. Zancanaro. "Multiple shooting method for two-point boundary value problems". In: *Communications of the ACM* 5.12 (1962), pp. 613–614. ISSN: 15577317. DOI: 10.1145/355580.369128.

[54] Angxiu Ni. "Adjoint shadowing directions in hyperbolic systems for sensitivity analysis". In: *arXiv:1807.05568* (2018), pp. 1–23.

[55] Angxiu Ni. "Approximating linear response by non-intrusive shadowing algorithms". In: *https://arxiv.org/abs/2003.09801* (2020), pp. 1–12.

[56] Angxiu Ni. "Fast linear response algorithm for differentiating stationary measures of chaos". In: *arXiv:2009.00595* (2020), pp. 1–28.

[57] Angxiu Ni. "Hyperbolicity, shadowing directions and sensitivity analysis of a turbulent three-dimensional flow". In: *Journal of Fluid Mechanics* 863 (2019), pp. 644–669.

[58] Angxiu Ni and Chaitanya Talnikar. "Adjoint sensitivity analysis on chaotic dynamical systems by Non-Intrusive Least Squares Adjoint Shadowing (NILSAS)". In: *Journal of Computational Physics* 395 (2019), pp. 690–709. DOI: https://doi.org/10.1016/j.jcp.2019.06.035.

[59] Angxiu Ni and Qiqi Wang. "Sensitivity analysis on chaotic dynamical systems by Non-Intrusive Least Squares Shadowing (NILSS)". In: *Journal of Computational Physics* 347 (2017), pp. 56–77. ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2017.06.033. URL: http://www.sciencedirect.com/science/article/pii/S0021999117304783.

[60] Angxiu Ni et al. "Sensitivity analysis on chaotic dynamical system by Non-Intrusive Least Square Shadowing (NI-LSS)". In: *46th AIAA Fluid Dynamics Conference, AIAA AVIATION Forum (AIAA 2016-4399)*. American Institute of Aeronautics and Astronautics, June 2016, pp. 1–16. DOI: doi:10.2514/6.2016-4399. URL: https://doi.org/10.2514/6.2016-4399.

[61] Angxiu Ni et al. "Sensitivity analysis on chaotic dynamical systems by Finite Difference Non-Intrusive Least Squares Shadowing (FD-NILSS)". In: *Journal of Computational Physics* 394 (2019), pp. 615–631. DOI: https://doi.org/10.1016/j.jcp.2019.06.004.

[62] Todd A. Oliver et al. "Estimating uncertainties in statistics computed from direct numerical simulation". In: *Physics of Fluids* 26.3 (2014), p. 035101. ISSN: 10897666. DOI: 10.1063/1.4866813. arXiv: 1311.0828.

[63] S. Yu Pilyugin. "Shadowing in Structurally Stable Flows". In: *Journal of Differential Equations* 140.2 (1997), pp. 238–265. ISSN: 00220396. DOI: 10.1006/jdeq.1997.3295.

[64] T J Poinsot and S K Lelef. "Boundary conditions for direct simulations of compressible viscous flows". In: *Journal of Computational Physics* 101.1 (1992), pp. 104–129. ISSN: 0021-9991. DOI: http://dx.doi.org/10.1016/0021-9991(92)90046-2. URL: http://www.sciencedirect.com/science/article/pii/0021999192900462.

[65] Enrique R. Pujals and Martín Sambarino. "Homoclinic tangencies and hyperbolicity for surface diffeomorphisms". In: *Annals of Mathematics* 151.3 (2000), pp. 961–1023. ISSN: 0003486X. DOI: 10.2307/121127. arXiv: 0005303 [math].

[66] P. Roe. "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes". In: *Journal of Computational Physics* 43 (1981), pp. 357–372.

[67] Anatol Roshko. "Experiments on the flow past a circular cylinder at very high Reynolds number". In: *Journal of Fluid Mechanics* 10.3 (1961), pp. 345–356. ISSN: 14697645. DOI: 10.1017/S0022112061000950. URL: http://www.journals.cambridge.org/abstract_S0022112061000950.

[68] David Ruelle. "Differentiation of SRB States". In: *Commun. Math. Phys* 187 (1997), pp. 227–241.

[69] David Ruelle. "Differentiation of SRB states for hyperbolic flows". In: *Ergodic Theory and Dynamical Systems* 28.02 (2008), pp. 613–631.

[70] David Ruelle. "Differentiation of SRB States: Correction and Complements". In: *Communications in Mathematical Physics* (2003), pp. 185–190.

[71] David Ruelle. *Elements of Differentiable Dynamics and Bifurcation Theory*. San Diego: Academic Press, 1989. DOI: 10.1016/c2013-0-11426-2.

[72] Manuel Santos Gutiérrez and Valerio Lucarini. "Response and Sensitivity Using Markov Chains". In: *Journal of Statistical Physics* 179.5-6 (2020), pp. 1572–1593. ISSN: 15729613. DOI: 10.1007/s10955-020-02504-4. arXiv: 1907.12881.

[73] Karim Shawki and George Papadakis. "A preconditioned multiple shooting shadowing algorithm for the sensitivity analysis of chaotic systems". In: *Journal of Computational Physics* 398 (2019), p. 108861. ISSN: 23318422.

[74] Yukiko S. Shimizu and Krzysztof J. Fidkowski. "Output-based error estimation for chaotic flows using reduced-order modeling". In: *AIAA Aerospace Sciences Meeting, 2018* 210059 (2018). DOI: 10.2514/6.2018-0826.

[75] Michael Shub. *Global Stability of Dynamical Systems*. Berlin: Springer, 1987. ISBN: 9781441930798. DOI: 10.1007/978-1-4757-1947-5.

[76] Adam A. Sliwiak, Nisha Chandramoorthy, and Qiqi Wang. "Ergodic sensitivity analysis of one-dimensional chaotic maps". In: *Theoretical and Applied Mechanics Letters* 10 (2020), pp. 438–447. ISSN: 23318422.

[77] S. Smale. "Differentiable Dynamical Systems". In: *Bull. Amer. Math. Soc.* 73 (1967), pp. 747–817. DOI: 10.1016/b978-0-12-601710-6.50004-3.

[78] Chaitanya Talnikar and Qiqi Wang. "A two-level computational graph method for the adjoint of a finite volume based compressible unsteady flow solver". In: *Parallel Computing* 81 (2019), pp. 68–84. ISSN: 0167-8191. DOI: 10.1016/j.parco.2018.12.001.

[79] Chaitanya Talnikar et al. "Optimization with LES – algorithms for dealing with sampling error of turbulence statistics". In: *53rd AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2015-1954)*. Kissimmee, Florida, 2015, pp. 1–11.

[80] D. J. Tritton. "Experiments on the flow past a circular cylinder at low Reynolds numbers". In: *Journal of Fluid Mechanics* 6.4 (1959), pp. 547–567. ISSN: 1469-7645. DOI: 10.1017/S0022112059000829. URL: http://journals.cambridge.org/action/displayAbstract?fromPage=online&aid=367814.

[81] H. K. Versteeg and W. Malaskekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson, 1995, p. 517. ISBN: 0-470-23515-2. DOI: 10.2514/1.22547. arXiv: arXiv:1011.1669v3.

[82] Qiqi Wang. "Convergence of the Least Squares Shadowing Method for Computing Derivative of Ergodic Averages". In: *SIAM Journal on Numerical Analysis* 52.1 (2014), pp. 156–170. ISSN: 0036-1429. DOI: 10.1137/130917065. arXiv: arXiv:1304.3635v7. URL: http://epubs.siam.org/doi/abs/10.1137/130917065%5Cnpapers3://publication/doi/10.1137/130917065.

[83] C. Wieselsberger. "New data on the laws of fluid resistance". In: *Physikalische Zeitscrift* 22 (1921), pp. 321–328. ISSN: 1356-9783. DOI: 10.1080/1356978042000185885.

[84] C. H. K. Williamson and A. Roshko. "Measurements of base pressure in the wake of a cylinder at low Reynolds numbers". In: *Zeitschrift fur Flugwissenschaften und Weltraumforschung* 14 (1990), pp. 38–46.

[85] Caroline L. Wormell and Georg A. Gottwald. "Linear response for macroscopic observables in high-dimensional systems". In: *Chaos* 29.11 (2019). ISSN: 10541500. DOI: 10.1063/1.5122740.

[86] Lai-Sang Young. "Ergodic Theory of Differentiable Dynamical Systems". In: *Real and Complex Dynamical Systems* (1995), pp. 293–336. DOI: 10.1007/978-94-015-8439-5_12.

[87] Lai-Sang Young. "What are SRB measures, and which dynamical systems have them?" In: *Journal of Statistical Physics* 108.5 (2002), pp. 733–754.