Sensitivity analysis on chaotic dynamical systems by Non-Intrusive Least Squares Shadowing (NILSS)

by

Angxiu Ni

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Signature redacted

Author ....
Department of Aeronautics and Astronautics
May 18, 2017

Signature redacted

Certified by..... .....
Qiqi Wang
Associate Professor
Thesis Supervisor

Signature redacted

Accepted by ....
Youssef M. Marzouk
Chairman, Graduate Program Committee

Sensitivity analysis on chaotic dynamical systems by Non-Intrusive Least Squares Shadowing (NILSS)

by

Angxiu Ni

Submitted to the Department of Aeronautics and Astronautics
on May 18, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis develops the Non-Intrusive Least Squares Shadowing (NILSS) method, which computes the sensitivity for long-time averaged objectives in chaotic dynamical systems. In NILSS, we represent a tangent solution by a linear combination of one inhomogeneous tangent solution and several homogeneous tangent solutions. Next, we solve a least squares problem using this representation; thus, the resulting solution can be used for computing sensitivities. NILSS is easy to implement with existing solvers. In addition, for chaotic systems with many degrees of freedom but few unstable modes, NILSS has a low computational cost. NILSS is applied to two chaotic PDE systems: the Lorenz 63 system and a CFD simulation of flow over a backward-facing step. In both cases, the sensitivities computed by NILSS reflect the trends in the long-time averaged objectives of dynamical systems.

Thesis Supervisor: Qiqi Wang
Title: Associate Professor

Acknowledgments

I would like to thank my adviser, Professor Qiqi Wang, for his support and guidance during my three years at MIT. His knowledge and insight helped develop the directions for my research; working with him has been inspiring.

I would like to thank professors in the Department of AeroAstro for their insightful feedback on my research. They had with me many open discussions, which cleared many anxieties. I am especially grateful to Professor Youssef Marzouk, Jaime Peraire, Eytan Modiano, David Darmofal and Cuong Nguyen.

I would like to thank my friends at ACDL, AeroAstro as well as in the whole MIT community for the time I spent at the institute. We had many fruitful discussions and hilarious moments together. I am especially grateful to Zheng Wang, Ben Zhang, and Pablo Fernandez, who patiently helped me improve my communication skills in English.

I would like to thank the teachers who introduced me the beauty and fun of pure math, specifically Professor Richard Melrose and Professor Michael Artin, who wrote reference letters when I apply to grad schools. Moreover, they both taught me how to live lives.

I am grateful to my parents, Xiuming Liu and Guoqiang Ni, for supporting me through this time. They had always been learning new things so that they could give me the best support they can. Being their son encourages me to keep chasing a higher level of life.

Contents

List of Figures

Chapter 1

Introduction

Many important phenomena in engineering, such as turbulent flow [19] and some fluid-structure interactions [13], are chaotic. In these systems, the objectives we are often interested in are long-time averaged rather than instantaneous quantities. Furthermore, we want to perform sensitivity analysis, i.e., we want to know how a change in the parameters of a system can affect its objectives. Such sensitivity analysis is the purpose of this thesis.

To rigorously define the problem, we first consider the governing equation for a chaotic dynamical system:

$$\frac{du}{dt} = f(u, s), \quad u(t = 0) = u_0(\phi), \tag{1.1}$$

where $f(u, s) : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ is a smooth function, u is the state, and s is the parameter. The initial condition u_0 is a smooth function of ϕ. A solution $u(t)$ is called the primal solution.

In this thesis, The objective is a long-time averaged quantity. To define it, we first let $J(u, s) : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ be a continuous function that represents the instantaneous objective function. The objective is obtained by averaging J over a infinitely long trajectory:

$$\langle J \rangle_\infty := \lim_{t \to \infty} \langle J \rangle_T, \text{ where } \langle J \rangle_T := \frac{1}{T} \int_0^T J(u, s) dt. \tag{1.2}$$

$\langle J \rangle_T$ depends on s, ϕ, and T, while $\langle J \rangle_\infty$ is determined only by s and u_0. Here we

make the assumption of ergodicity [32], which means that u_0, hence ϕ does not affect $\langle J \rangle_\infty$. As a result, $\langle J \rangle_\infty$ only depends on s.

The purpose of this thesis is to develop an algorithm that computes the sensitivity $d\langle J \rangle_\infty / ds$. The sensitivity can help scientists and engineers design products [17, 26], control processes and systems [5, 6], solve inverse problems [31], estimate simulation errors [3, 16, 15], assimilate measurement data [29, 12] and quantify uncertainties [34].

When a dynamical system is chaotic, computing a meaningful $d\langle J \rangle_\infty / ds$ is challenging, since in general:

$$\frac{d}{ds}\langle J \rangle_\infty \neq \lim_{T\to\infty} \frac{\partial}{\partial s}\langle J \rangle_T (s, \phi, T). \tag{1.3}$$

That is, if we fix $u_0(\phi)$, the process of $T \to \infty$ does not commute with differentiation with respect to s [9]. As a result, the transient method, which employs the conventional tangent method with a fixed u_0, does not converge to the correct sensitivity for chaotic systems. In fact, the transient method diverges most of the time [9].

Many sensitivity analysis methods have been developed to compute $d\langle J \rangle_\infty / ds$. The conventional methods include the finite difference and transient method. The ensemble method, developed by Lea et al. [20, 14], computes the sensitivity by averaging results from the transient method over an ensemble of trajectories. Another recent approach is based on the fluctuation dissipation theorem (FDT), as seen in [30, 23, 35, 21, 1, 2].

In this research study, we consider the Least Squares Shadowing (LSS) approach, developed by Wang, Hu and Blonigan [33, 34]. LSS computes a bounded shift of a trajectory under an infinitesimal parameter change, which is called the LSS solution. The LSS solution can then be used to compute the derivative $d\langle J \rangle_\infty / ds$. LSS has been successfully applied to dynamical systems such as the Lorenz 63 system and a modified Kuramoto-Sivashinsky equation [7, 34, 8]. LSS has also been applied, by Blonigan et al., to sensitivity analysis for flow around airfoils [9, 8]. From a theoretical standpoint, Wang has proven that, under ergodicity and hyperbolicity assumptions,

LSS converges to the correct sensitivity at a rate of $T^{-0.5}$, where T is the trajectory time length[33].

However, for large systems which arise in real life problems, LSS is expensive, since it involves solving a large linear system, where the number of variables is the system dimension times the number of time steps. As the system gets larger and the trajectory longer, the linear system becomes very large and possibly stiff. Although solving the system could be accelerated by preconditioners and iterative methods [8], there would still be a large cost in both computational time and memory. Furthermore, LSS requires the Jacobian matrix $\partial_u f(u, s)$ at each time step, which many existing simulation software may not readily provide; and making modifications to existing codes can be difficult.

To reduce the computational cost and ease the implementation of LSS, this thesis presents the Non-Intrusive Least Squares Shadowing (NILSS) method. The computational and memory cost of NILSS are both proportional to the number of positive Lyapunov Exponents (LE). For many real life applications this number is smaller than the dimension of the dynamical system, and the cost of NILSS could be lower than LSS. Another benefit is that NILSS requires less modification to the underlying tangent solver than LSS, since it does not require the Jacobian matrix $\partial_u f(u, s)$.

The rest of this thesis presents the NILSS algorithm as follows: First, we examine the long-time and transient effects due to perturbations in the system parameters. We also examine how transient effects are also generated by perturbations in initial conditions. Next, we describe the NILSS method as a procedure that distills the long-time effect by subtracting the transient effect, where perturbations are expressed by tangent solutions. Then, we present a step-by-step description of the NILSS algorithm. Finally, we apply NILSS to the Lorenz 63 system and a CFD simulation of a flow over a backward-facing step.

Chapter 2

The idea of NILSS

In our definition of the objective, the average is taken over a trajectory. Hence, intuitively, by looking at how the trajectory is perturbed due to parameter perturbations, the sensitivity of the objective could be revealed. In this chapter, we first examine such perturbations, where we shall see the parameter perturbation has two effects: 1) the long-time effect, which is of our interest, and 2) the transient effect, which could be represented by initial condition perturbations. The following sections of the chapter develops the NILSS algorithm, which can be interpreted as 'subtracting' the transient effect from the parameter perturbation.

2.1 Connection between sensitivity to system parameters and initial conditions

Trajectories of chaotic dynamical systems depend sensitively on system parameters. If we change any parameter by a small amount, the new trajectory will be significantly different than the old one, even though they start from the same initial condition. This is similar to another sensitive dependence on initial conditions, better known as the 'butterfly effect', i.e., a small difference in the initial condition can grow larger and larger as the system evolves.

To illustrate the similarity between the two sensitivities, we consider the Lorenz

63 system, which is a simplified ODE model for atmospheric convection [22]. Lorenz 63 has three states x, y, z and a parameter ρ. In Figure 2-1, we show the sensitive dependence of trajectories on both the initial condition and the parameter. In the left column, on the x-z axis, we plot planar snapshots of 1.8×10^7 trajectories with varying ρ but with the same initial condition $u_0 = (12.00, 6.82, 36.47)$. Here ρ is uniformly distributed in $28 \pm \Delta\rho$, where $\Delta\rho = 1$. Note that a smaller ρ is indicated by colors with shorter wavelengths (blue), while a larger ρ by longer wavelengths (red). On the right column, we plot snapshots of the same number of trajectories with the same parameter $\rho = 28$, but with a varying initial initial condition, which is characterized by a vector that is uniformly distributed along $u_0 \pm \Delta u_0$, where $\Delta u_0 = [0.0939, -0.001053, 1.025]$. As we shall see later, Δu_0 is chosen to have similar effects to the transient effect of varying ρ.

There are many similarities and subtle differences between the effects of a varying ρ and a varying u_0. As we can see in the first three rows of Figure 2-1, in the short time, the varying ρ results in diverging trajectories which looks like the effect of only varying u_0: we call this the transient effect.

The picture in the last row of Figure 2-1 is obtained after letting the trajectories evolve over a long time. The picture on the right gives the attractor of the base parameter. The picture on the left, at first glance, has similar shape as the attractor to its right. However, the left figure is the superposition of many attractors with different parameters, and a closer look shows that it has different colors in different parts. The red color on the upper rim, and the blue on the lower, indicates that as ρ increases, the attractor moves upward in the z direction. To conclude, in the long-time, varying ρ results in a shifted attractor: we call this the long-time effect.

The long-time effect generated by a varying ρ is important for computing the long-time sensitivity, however, it is hidden beneath diverging trajectories and is only visible after a long time and an ensemble of millions of trajectories. As we said, the transient effect is reflected by diverging trajectories, hence if we can find two trajectories, one with ρ and another with $\rho + \delta\rho$, which do not diverge, then their difference does not contain the transient effect. Now with the transient effect gone,

their difference contains only the long-time effect. Thus, we can reveal the long-time effect with a shorter trajectory.

Our main goal in this paper is to devise an algorithm that can generate the transient effect and subsequently 'subtract' the transient effect from a varying ρ, so that we can find two trajectories that do not diverge from each other, and whose difference only contains the long-time effect. In fact, in Figure 2-1, $\Delta u_0 = v\Delta\rho$ and v represents the NILSS solution. As we shall see, this change in the initial condition yields the transient effect, and by subtracting it from the two effects of a varying ρ, we can distill the long-time effect using a short trajectory. We will clarify the qualitative description of 'subtraction' in later sections.

2.2 Describing perturbations by tangents

To solidify the intuition built in Section 2.1, in this section, we mathematically characterize the two perturbations as two different tangent solutions, i.e., inhomogeneous and homogeneous tangents. To distill the long-time effect, denoted by some inhomogeneous tangent v, we want to construct a homogeneous tangent w which represents the transient effect brought about by varying initial conditions, and subtract it from the conventional inhomogeneous tangent v^*, which represents the two effects of the varying parameters. In Section 2.3, we see how to mathematically construct such a w as a linear combination of unstable Characteristic Lyapunov Vectors (CLV). In Section 2.4, we give a computationally efficient formula for w from only the conventional tangent v^* and several homogeneous solutions $\{w_j\}$, which approximate unstable CLVs. Finally, Section 2.5 explains how to compute $d\langle J\rangle_\infty/ds$.

Now we mathematically describe the trajectory perturbations generated by parameter and initial condition perturbations. This is done by tangent solutions. Specifically, the perturbation due to parameter change is described by inhomogeneous tangents, while that due to initial condition change is described by homogeneous tangents.

First, we differentiate the dynamical system in Equation (1.1) with respect to s,

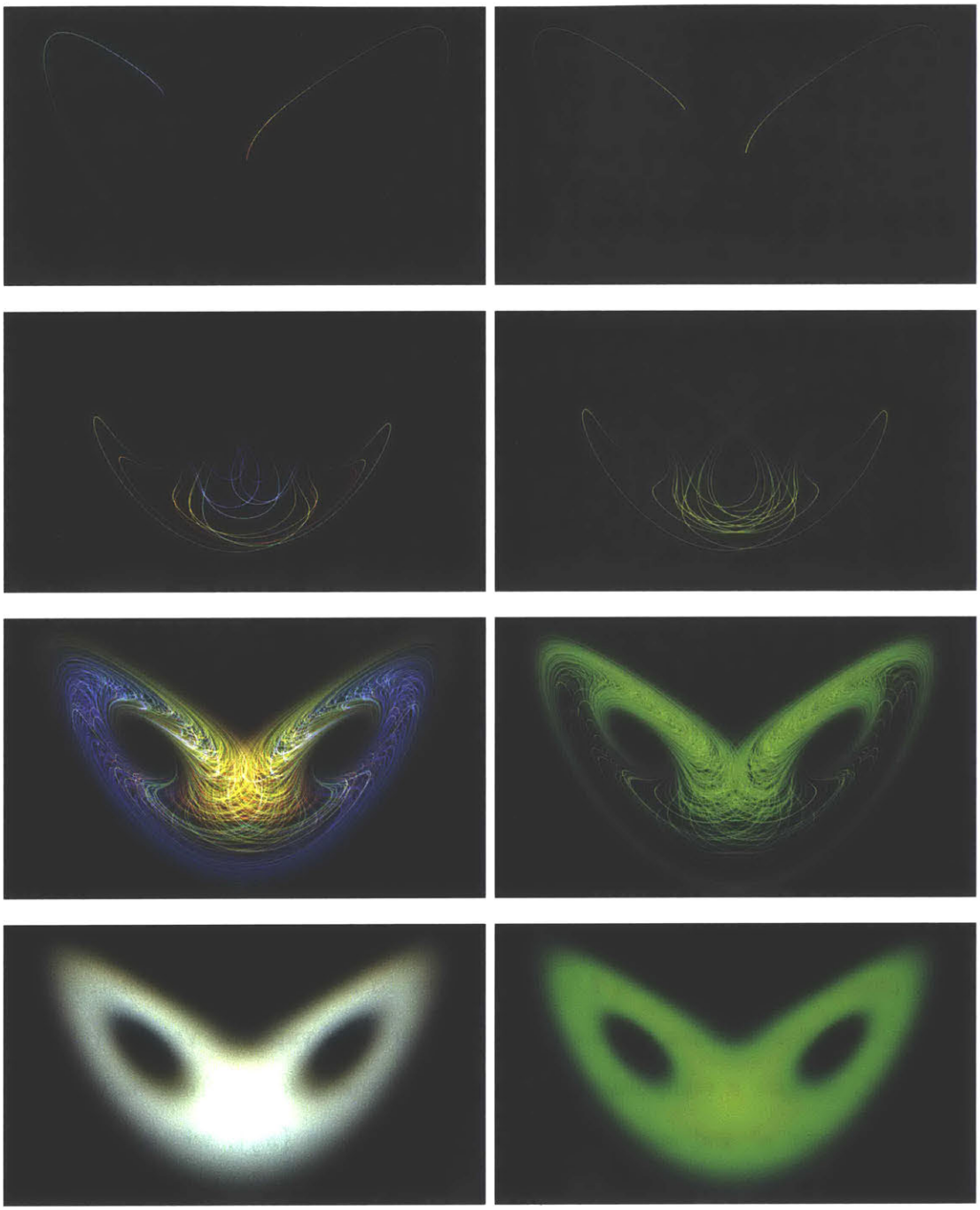

Figure 2-1: Snapshots of an ensemble of 1.8×10^7 trajectories of the Lorenz 63 system. Left column: trajectories with different parameters that are uniformly distributed over the range [27,29], where smaller ρ is indicated by blue, larger ρ by red. Right column: trajectories with fixed ρ but with initial conditions uniformly distributed over $(12.00, 6.82, 36.47) \pm [0.0939, -0.001053, 1.025]$. From top to bottom: snapshots taken at time 1.67, 5.0, 10.0, and 41.67.

while keeping ϕ fixed. Then, we let $v^* = \partial u/\partial s$. Thus, the governing equation for v^* is:

$$\frac{dv^*}{dt} - \partial_u f v^* = \partial_s f, \quad v^*(t=0) = 0 \tag{2.1}$$

where $\partial_u f$ is an $\mathbb{R}^m \times \mathbb{R}^m$ matrix and $\partial_s f$ is an $\mathbb{R}^m$ column vector. The zero initial condition v^* reflects that u_0 remains unchanged. By definition, v^* reflects the trajectory perturbation due to parameter changes under fixed initial conditions, which is shown in the left column of Figure 2-1.

Under the assumption of ergodicity, the long time behavior is not affected by the selection of initial conditions. This suggests that ϕ is not necessarily fixed if we are only interested in the change of the long-time average. We define inhomogeneous tangent solutions as solutions that satisfy the ODE in Equation (2.1), but without the initial condition:

$$\frac{dv}{dt} - \partial_u f v = \partial_s f. \tag{2.2}$$

Notice that this ODE is under-determined. To get a solution we can either provide an initial condition, as we did for v^*, or put the ODE as a constraint for some optimization problems. By its definition, v reflects the trajectory perturbation due to a change in the parameter, while the initial condition change is not specified.

We define $w = \partial u/\partial \phi$, where s is assumed to be fixed and w satisfies the so called homogeneous tangent equation:

$$\frac{dw}{dt} - \partial_u f w = 0. \tag{2.3}$$

By its definition, w characterizes the perturbations due to initial condition changes while s is fixed, as shown in the right column of Figure 2-1.

Hence v^* and w describe the effect of only varying s and u_0, respectively. Also, Equation (2.3) differs from Equation (2.2) by setting the right hand side to zero. For two different inhomogeneous tangent solutions, say v^* and an arbitrary v, their difference is a homogeneous tangent solution w.

We know that if we vary s, we generate two effects: one is equivalent to varying u_0;

while the other shifts the attractor. Since we are interested in the latter, we want to find a w such that $v = v^* + w$ contains only the long-time but not the transient effect. Here we used addition, but we can replace w by $-w$ so that we have subtraction in the formula.

Subtracting such w from v^* is the main idea behind NILSS. As discussed in Section 2.1, we want to find two trajectories, one associated with parameter s and the other with $s + \delta s$, which do not diverge. Given the tangent solution definition, we can mathematically state that a v, if its Euclidean norm[1] of its orthogonal projection onto $V^{\perp}(u)$ remains bounded as the trajectory length goes to infinity, then this v suffices to reveal the long-time effect of the varying parameter. We denote this sufficient v by the shadowing direction, v^{∞}, whose existence is proved by the shadowing lemma [24]. Here $V^{\perp}(u)$ is defined as:

$$V^{\perp}(u) = \{p \in \mathbb{R}^m : p^T f(u) = 0\}, \tag{2.4}$$

where p^T is the transpose of the column vector p. Moreover, the orthogonal projection $p^{\perp}$ of p is defined as:

$$p^{\perp} = p - \frac{f^T p}{f^T f} f. \tag{2.5}$$

$v^{\infty\perp}$ is defined by substituting p by v^{∞}. We define $w^{\perp}$, $\delta u^{\perp}$, $v^{*\perp}$, and $\{\zeta_j^{\perp}\}$ in a similar way. We use the norm of $v^{\infty\perp}$ because it describes the perpendicular distance between two trajectories. A more mathematical explanation of why such v^{∞} can be used to compute the sensitivity is in Appendix C.

2.3 Constructing w from unstable Characteristic Lyapunov Vectors (CLV)

The main goal of NILSS is to find a w such that $v^{\perp} = v^{*\perp} + w^{\perp}$ approximates $v^{\infty\perp}$ on a finite trajectory. Here v^{∞} is the inhomogeneous tangent the norm of whose

[1] In this paper, the norm we use is Euclidean norm.

projection, $v^{\infty\perp}$, remains bounded even on an infinitely long trajectory. Notice that the NILSS solution $v^{\perp}$ may be not bounded if we extend it to an infinitely long trajectory; however, on the finite trajectory where NILSS is solved, $v^{\perp}$ provides a good approximation of $v^{\infty\perp}$. Specifically, this means that if we apply both $v^{\perp}$ and $v^{\infty\perp}$ to the formula that computes sensitivity in Equation (3.26), the results are similar.

In this subsection, we shall see one way to construct such a w by supposing that we know v^{∞} and all CLVs. This method is unrealistic since it requires too much computation. Yet it is informative since it shows that we only need a linear combination of unstable CLVs to construct a desired w. Based on this knowledge, we develop the NILSS method in the next subsection.

To further clarify this method, we should first define Lyapunov Exponent (LE) and the corresponding CLVs. We assume that the dynamical system has a full set of LEs and corresponding CLVs [27]. That is, there are $\{\lambda_j, j = 1, 2, \cdots, m\}$, such that for any trajectory on the attractor and a corresponding homogeneous tangent solution $w(u)$, there is a unique representation of $w(u)$:

$$w(u) = \sum_{j=1}^{m} a_j \zeta_j(u), \tag{2.6}$$

where $a_j \in \mathbb{R}$ is a constant for all $u(t)$ on the trajectory.

Here each $\zeta_j(u)$ is a homogeneous tangent solution, and its norm behaves like an exponential function of time. That is, there exists $C_1, C_2 > 0$, such that for any $u(t)$ on the attractor and any j and t, a CLV satisfies

$$C_1 e^{\lambda_j t} \|\zeta_j(u(0))\| \leq \|\zeta_j(u(t))\| \leq C_2 e^{\lambda_j t} \|\zeta_j(u(0))\|, \tag{2.7}$$

where $\{\lambda_j\}$ and $\{\zeta_j\}$ are LEs and CLVs, respectively. CLVs with $\lambda_j > 0$ are called unstable modes, those with negative $\lambda_j < 0$ are stable modes, and those with $\lambda_j = 0$ are neutral modes. We denote the unstable modes by $\zeta_1, \cdots, \zeta_{m_{us}}$ and the neutral modes by ζ_m. The remaining modes are the stable modes. In fact, unstable modes

are the reason for the 'butterfly effect' since a perturbation in the unstable subspace grows exponentially over time.

We assume that for all s we are interested in, there is no point u on the attractor Λ such that $f(u) = 0$ and Λ is bounded. These two assumptions imply that, per Appendix A, $f(u)$ is a CLV whose LE is 0. We further assume that $f(u)$ is the only neutral mode.

Although CLVs are not necessarily in $V^{\perp}$, we can project them onto $V^{\perp}$. Thus, Equation (2.6) becomes:

$$w^{\perp} = \sum_{j=1}^{m-1} a_j \, \zeta_j^{\perp}(u), \tag{2.8}$$

where $w^{\perp}$ and $\zeta_j^{\perp}$ are orthogonal projections as defined by Equation (2.5). Because $V^{\perp}$ is perpendicular to $f(u)$, the projection of the neutral mode is zero. This implies that the summation in Equation (2.8) only considers the stable and unstable modes, the total number of which is $m - 1$. We also call $\zeta_j^{\perp}$ stable or unstable modes based on their corresponding λ_j.

We assume that all CLVs are uniformly bounded away from each other. Under this assumption, the norm of stable and unstable modes $\{\zeta_j^{\perp}\}$ also behave like exponentials, as defined in Equation 2.7. Appendix B justifies this claim.

Suppose that v^{∞} and its CLVs are known. Since $v^{\infty} - v^{*}$ is a homogeneous tangent solution, we can decompose $v^{\infty\perp} - v^{*\perp}$ via Equation (2.8). By using the first m_{us} coefficients in this decomposition, we let

$$w = \sum_{j=1}^{m_{us}} a_j \zeta_j. \tag{2.9}$$

Thus, $v^{\perp} = v^{*\perp} + w$ approximates $v^{\infty\perp}$ since $v^{\infty\perp} - v^{\perp}$ is composed of only stable modes, which decay exponentially.

The important information in this method is that w is a linear combination of only unstable modes. To find the coefficients of this linear combination using the method given here, we need to know all the CLVs and v^{∞}. This method is infeasible since the computational cost will be high to find all CLVs and v^{∞} is unknown *a priori*. These

difficulties are overcome in the next subsection.

2.4 Computing $v^{\perp}$ by NILSS

In NILSS, we compute $v = v^* + w$ such that $v^{\perp} \approx v^{\infty\perp}$. More specifically, we want the integration of $v^{\perp}$ to approximate $v^{\infty\perp}$ so that later, when computing the sensitivity via Equation (2.12), $v^{\perp}$ yields a result close to the result given by $v^{\infty\perp}$. To achieve this, we solve the NILSS problem on a single time segment, which is to minimize the L^2 norm of $v^{\perp} = v^{*\perp} + W^{\perp}a$:

$$\min_a \frac{1}{2}\int_0^T (v^{*\perp} + W^{\perp}a)^T(v^{*\perp} + W^{\perp}a)\, dt, \tag{2.10}$$

which is simply a least squares problem with arguments $a \in \mathbb{R}^M$, where M is an integer larger than the number of positive LEs m_{us}. Here v^* is the conventional tangent solution, and $W^{\perp}(t)$ is a matrix whose columns are homogeneous tangent solutions $\{w_j^{\perp}(t), j = 1, ..., M\}$. The initial conditions $\{w_j(t = 0)\}$ are randomized unit vectors in $\mathbb{R}^m$.

First we need to see that a desired $v^{\perp}$ exists in the feasible solution space, or that some a can yield a desired $v^{\perp} = v^{*\perp} + W^{\perp}a$. Our discussion in the last subsection confirms the existence if we use unstable CLVs instead of W. Moreover, [4] proves that as time evolves, the span of $\{w_j^{\perp}(t), j = 1, ..., M\}$ converges to the span of the CLVs $\{\zeta_j^{\perp}(t), j = 1, ..., M\}$ with the largest M LEs. As a result, replacing unstable CLVs by $W^{\perp}$ gives a feasible solution space that contains a $v^{\perp}$ such that $v^{\perp} \approx v^{\infty\perp}$.

Next, we need to rationalize that minimizing the L^2 norm of $v^{\perp} = v^{*\perp} + W^{\perp}a$ yields $v^{\perp} \approx v^{\infty\perp}$. First, we notice that $v^{\perp}$ can be written as the summation of $v^{\infty\perp}$ and some homogeneous tangents. Because $v^{\infty\perp}$ is bounded and the unstable modes (now approximated by the span of W) grow exponentially, then minimizing $v^{\perp}$ over a long trajectory implies the difference $v^{\perp} - v^{\infty\perp}$ cannot contain significant unstable components. Although stable modes may be left in this difference, they decay exponentially. The effect of the minimization is illustrated by Figure 2-2.

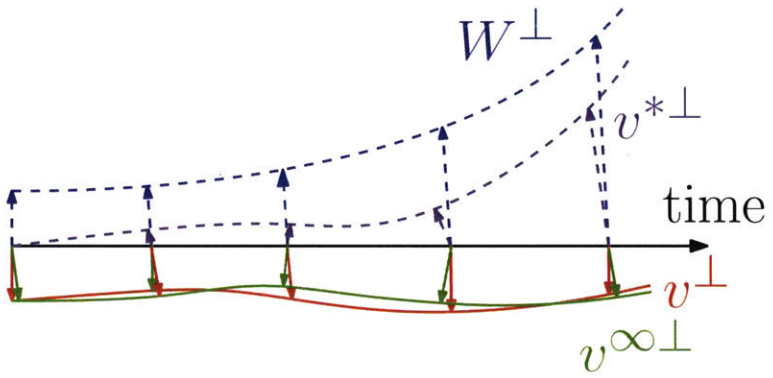

Figure 2-2: Intuition of NILSS: through minimization over $\|v^{\perp}\|$, we find a column vector a, such that $v^{\perp} = v^{*\perp} + W^{\perp}a \approx v^{\infty\perp}$. This is because most unstable components in $v^{*\perp} - v^{\infty\perp}$ are subtracted by $W^{\perp}a$ during the minimization.

2.5 Computing $d\langle J\rangle_{\infty}/ds$ from the tangent solution

Since $v - v^{\perp}$ is parallel to f, we can define ξ as the scalar which satisfies:

$$\xi f = v - v^{\perp}. \tag{2.11}$$

To find a pair $(v^{\perp}, \xi)$, we first find a v which solves Equation (2.2), project v onto the subspace $V^{\perp}$ to find $v^{\perp}$, then use Equation (2.11) to find ξ.

Once we obtain the solution vector a of the NILSS problem, we can construct $v = v^{*} + Wa$ and compute the corresponding ξ. Then we have the following approximation for $d\langle J\rangle_{\infty}/ds$:

$$\frac{d\langle J\rangle_{\infty}}{ds} \approx \frac{1}{T}\left[\int_0^T (\partial_u J\, v + \partial_s J)\, dt + \xi|_0^T \langle J\rangle_T - (\xi J)|_0^T\right], \tag{2.12}$$

where $\langle J\rangle_T$ is defined in Equation (1.2). Notice that in Equation (2.12), we use the tangent solution v instead of its projection $v^{\perp}$. The derivation of Equation (2.12) is in Appendix C.

2.6 Benefits of NILSS

The first benefit of NILSS is that it is easily implemented with existing tangent solvers. The data used in the NILSS problem are $v^{*\perp}$ and $\{w_j^{\perp}\}$. Here v^* is the result of a conventional tangent solver. $\{w_j\}$ are given by homogeneous tangent solvers, which can be obtained by setting the right hand side in Equation (2.2) to zero in conventional tangent solvers. Once we have v^* and $\{w_j\}$, $v^{*\perp}$ and $\{w_j^{\perp}\}$ can be computed by orthogonal projection onto $V^{\perp}(u)$, as shown in Equation (2.5).

Another way to compute those tangent solutions is to approximate them by finite difference solutions. This leads to the finite difference NILSS.[2] In this way, NILSS requires only primal simulation and no longer the tangent solvers. An explanation of finite difference NILSS is in Appendix D.

In NILSS, the optimization problem is comprised of only a small part of the computational cost, since there are only M arguments in Equation (2.10). The main cost comes from setting-up the optimization problem by computing $v^{*\perp}$ and $w_1^{\perp}(t), ..., w_M^{\perp}(t)$. Hence the cost of NILSS is proportional to the number of unstable modes m_{us}. For engineering problems, m_{us} is usually much smaller than m; thus, the cost of NILSS is low.

A beneficial side-effect is that NILSS uses less computer memory than LSS. Furthermore, the tangent solutions used in NILSS do not need to be saved in the computer memory concurrently. NILSS can use tangent solutions saved on an external hard drive, which can then be read in pairs to compute their inner product; this may reduce the computational speed, but further saves computer memory.

[2]The python package 'fds', which implements the algorithm introduced in this paper, for both tangent NILSS and the finite difference variant, are available at github.com/qiqi/fds.

Chapter 3

Tangent NILSS Algorithm

In this chapter, we first address the numerical stability of the algorithm by rescaling $v^{*\perp}$ and $W^{\perp}$ after every short segment of time ΔT. Then, we discuss the criterion for determining the number of homogeneous solutions M and segment length ΔT. Finally, we provide a walk-through of the NILSS algorithm.

3.1 Solving NILSS on multiple time segments

Since both $v^{*\perp}$ and $W^{\perp}$ grow exponentially, the round-off error when storing them in the computer become non-negligible over time. The growth in $v^{*\perp}$ and $W^{\perp}$ will also generate an ill-conditioned covariance matrix $(W^{\perp})^T W^{\perp}$, since all $\{w_j^{\perp}\}$ will eventually be dominated by the fastest growing unstable CLV. This section shows how to prevent this by partitioning a long trajectory into a series of shorter segments.

We partition the time domain into K time segments $[t_0, t_1], [t_1, t_2], \ldots, [t_{K-1}, t_K]$, with $t_0 = 0, t_K = T$. Next, we define time segment i as $[t_i, t_{i+1}], i = 0, \ldots, K-1$. For each time segment i, we define an inhomogeneous solution $\{v_i^*\}$ and homogeneous solutions $\{W_i\}$, such that each $W_i = [w_{i,1}, \cdots, w_{i,M}]$. This notation is depicted in Figure 3-1.

We want to rescale and orthogonalize $v_i^{*\perp}$ and $W_i^{\perp}$ at the end of each segment so that they do not grow too large or become dominated by the fastest growing CLV. We also want to keep the affine vector space $v^{*\perp} + span(W^{\perp})$ the same across interfaces

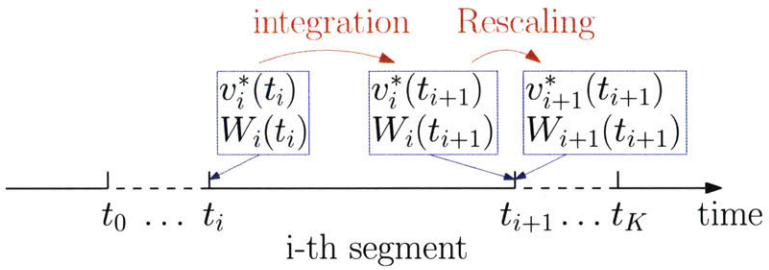

Figure 3-1: Notations used for NILSS, $t_0 = 0, t_K = T$

between contingent segments, so that we can recover a continuous $v^{\perp}$:

$$v_i^{*\perp}(t_i) + span\left(W_i^{\perp}(t_i)\right) = v_{i-1}^{*\perp}(t_i) + span\left(W_{i-1}^{\perp}(t_i)\right), \tag{3.1}$$

where $span(W^{\perp})$ is the vector space spanned by the column vectors of $W^{\perp}$.

To achieve this, we first orthonormalize $W^{\perp}$ via a QR decomposition:

$$W_i^{\perp}(t_{i+1}) = Q_i R_i. \tag{3.2}$$

We set the initial conditions of the next tangent segment to

$$W_{i+1}(t_{i+1}) = Q_i. \tag{3.3}$$

In QR factorization, the column vectors in Q_i and $W_i^{\perp}$ could represent each other if the column vectors in $W_i^{\perp}$ are linearly independent. Indeed, the linear independence of the initial condition of $W_i^{\perp}$ can be preserved after ΔT, if f is Lipschitz continuous. Hence, the $span\left(W_i^{\perp}(t_i)\right) = span\left(W_{i-1}^{\perp}(t_i)\right)$.

We subtract from $v^{*\perp}$ its orthogonal projection on $W^{\perp}$ to obtain the initial condition of the next time segment:

$$v_{i+1}^*(t_{i+1}) = v_i^{*\perp}(t_{i+1}) - W_{i+1}^{\perp}(t_{i+1})b_i, \tag{3.4}$$

where $b_i = W_{i+1}^{\perp}(t_{i+1})^T v_i^{*\perp}(t_{i+1})$. $v_{i+1}^{*\perp}(t_{i+1})$ is still in the affine space $v_i^{*\perp}(t_i) + span\left(W_i^{\perp}(t_i)\right)$. The norm of $v_i^{*\perp}(t_i)$ is reduced, since the unstable modes in it are subtracted through the projection.

We can recover a continuous $v^{\perp}$ over the whole trajectory. Now Equation (3.1) is satisfied, for any a_{i-1}, there exists a_i such that:

$$v_i^{*\perp}(t_i) + W_i^{\perp}(t_i)a_i = v_{i-1}^{*\perp}(t_i) + W_{i-1}^{\perp}(t_i)a_{i-1}. \tag{3.5}$$

Hence we have the necessary conditions to enforce the continuity requirement:

$$v_i^{\perp}(t_i) \;=\; v_{i-1}^{\perp}(t_i). \tag{3.6}$$

The solution $v^{\perp}$ over multiple time segments is equivalent to that over a longer segment. However, rescaling $v^{*\perp}$ and $W^{\perp}$ at the end of each time segment prevents them from growing too large.

Using QR factorization to rescale homogeneous solutions, while keeping a continuous affine subspace, is not a new idea. In a widely used technique for computing LE [4], the same idea was used.

3.2 Determining parameters for NILSS

There are two parameters in NILSS that users should choose: the number of homogeneous solutions M and the length of each time segment ΔT. This section discusses the criteria for determining these parameters. Once the parameters are determined, we can proceed to following sections about the detailed algorithm of NILSS.

M is determined based on the Lyapunov Exponents (LE), which are byproducts of NILSS. According to [4], λ_j, the j-th largest LE, is computed by:

$$\lambda_j \approx \frac{1}{K\Delta T}\sum_{i=1}^{K} |d_{i,j}|\,, \tag{3.7}$$

where $d_{i,j}$ is the j-th diagonal element in R_i.[1] Notice that the computation of $\{R_i\}$ only require W but not v^*. As we shall see in the detailed algorithm later, NILSS can compute homogeneous solutions W before v^*. At the stage of computing W, we

[1]The python package 'fds' has a function that computes LEs.

can gradually increase M and compute more LEs, which appear in decreasing order. Once we have a negative LE, we know that we have found all positive LEs.

ΔT is determined by the constraint that the CLV with the largest LE does not dominate the M-th CLV. If we assume the largest LE is λ_1 and the M-th LE is λ_M, then the ratio between the norm of these two CLVs satisfies:

$$\frac{\|\zeta_1^{\perp}(u(t))\|/\|\zeta_1^{\perp}(u(0))\|}{\|\zeta_M^{\perp}(u(t))\|/\|\zeta_M^{\perp}(u(0))\|} \approx \exp((\lambda_1 - \lambda_M)t). \tag{3.8}$$

This suggests that the ratio between the fastest growing and the M-th CLV grow about three times larger after a time span $(\lambda_1 - \lambda_M)^{-1}$. If ΔT is large, the covariance matrix C_i in Equation (3.11) will be ill-conditioned, which could pose a numerical problem. To prevent this from happening, we rescale $W^{\perp}$ and $v^{*\perp}$ after $\Delta T \leq (\lambda_1 - \lambda_M)^{-1}$. On the other hand, when ΔT get smaller, there are more segments, which leads to a larger optimization problem in Equation (3.22). This concern on cost gives the lower bound of ΔT.

The two criteria in this section are *a posteriori*, which means that we need to actually run NILSS for a few segments to check if they are satisfied. In most cases, the optimization problem does not significantly contribute to the computational cost of NILSS; thus, we recommend readers choose small ΔT to begin. After the ΔT is chosen, we can determine the M accordingly.

3.3 Pre-processing

First, we integrate Equation (1.1) over a sufficient period before $t = 0$ so that u is on the attractor at the beginning of our algorithm. Then, we integrate Equation (1.1) from $t = 0$ to $t = T$ to obtain the primal solution $u(t)$.

3.4 Computing the homogeneous solution $\{W_i\}$

We compute one inhomogeneous and M homogeneous tangent equations for each of the K time segments $[t_0, t_1]$,...,$[t_{K-1}, t_K]$, where $t_0 = 0, t_K = T$. Time segment i is

with the range $[t_i, t_{i+1}]$. This notation is the same as those found in Figure 3-1.

We start at the first segment with random initial conditions for each column vector in W:

$$W_0(0) = [w_{0,1}(0), \ldots, w_{0,M}(0)], \quad \text{with } w_{0,j}(0) \in V^{\perp}(u(0)). \tag{3.9}$$

Then, we proceed with the following algorithm, which starts at $i = 0$.

1. For each $j = 1, \cdots, M$, we start from the initial conditions $\{w_{i,j}(t_i)\}$. We then integrate Equation (2.3) to obtain $w_{i,j}(t)$,$t \in [t_i, t_{i+1}]$. We compute the orthogonal projection onto $V^{\perp}$ using Equation (2.5):

$$W_i^{\perp}(t) = [w_{i,1}^{\perp}(t), \ldots, w_{i,M}^{\perp}(t)], \quad t \in [t_i, t_{i+1}]. \tag{3.10}$$

2. Then, we compute and store the

$$C_i = \int_{t_i}^{t_{i+1}} (W_i^{\perp})^T W_i^{\perp} dt. \tag{3.11}$$

3. We orthonormalize $W_i^{\perp}(t_{i+1})$ with a QR decomposition under the Euclidean norm:

$$W_i^{\perp}(t_{i+1}) = Q_i R_i. \tag{3.12}$$

   Then, we store R_i and set the initial conditions of the next segment to

$$W_{i+1}(t_{i+1}) = Q_i. \tag{3.13}$$

4. Finally, we let $i = i + 1$, after which we go to Step 1 unless $i = K$, in which case we proceed to Section 3.5.

Here we compute $\{w_{i,j}\}$ from Equation (2.3). They may also be computed as the difference between two inhomogeneous tangent solutions:

$$\{w_{i,j}\} = v_{i,j}^{w} - v_i^{0}, \tag{3.14}$$

where $v_{i,j}^w$ has same initial condition as $w_{i,j}$ and v_i^0 has a zero initial condition at t_i. This way of computing homogeneous tangents no longer requires a separate homogeneous tangent solver.

3.5 Computing the inhomogeneous solution $\{v_i^*\}$

We start at the first time segment with initial condition: $v_0^*(0) = 0$, then proceed with the following algorithm starting at $i = 0$.

1. Starting from the initial condition $v_i^*(t_i)$, integrate the inhomogeneous Equation (2.2) to obtain $v_i^*(t)$, $t \in [t_i, t_{i+1}]$. Through Equation (2.5), we compute the orthogonal projection $v_i^{*\perp}(t)$, $t \in [t_i, t_{i+1}]$.

2. Compute and store
$$d_i = \int_{t_i}^{t_{i+1}} W_i^{\perp T} v_i^{*\perp} dt. \tag{3.15}$$

3. Orthogonalize $v_i^{*\perp}(t_{i+1})$ with respect to $W_{i+1}^{\perp}(t_{i+1})$ to obtain the initial condition of the next time segment:
$$v_{i+1}^*(t_{i+1}) = v_i^{*\perp}(t_{i+1}) - W_{i+1}^{\perp}(t_{i+1}) b_i, \tag{3.16}$$
where
$$b_i = W_{i+1}^{\perp}(t_{i+1})^T v_i^{*\perp}(t_{i+1}) \tag{3.17}$$
should be stored.

4. Let $i = i + 1$. Go to Step 1 unless $i = K$, in which case we proceed to Section 3.6.

Here we compute the inhomogeneous solution v_i^* and homogeneous solution W_i separately. By doing this, we can first find all positive LEs by gradually increasing M, since the computation of LE only requires homogeneous solutions. Once M is determined, we can go on to compute v^*. If we already know the number of positive LEs, then v_i^* and W_i can be computed simultaneously.

3.6 Computing v

Here we compute $\{v_i\}$ for each segment, with $v_i^{\perp}$ continuous across different segments. The minimization in Equation (2.10) becomes:

$$\sum_{i=0}^{K-1} \int_{t_i}^{t_{i+1}} \left[(v_i^{*\perp})^T v_i^{*\perp} + 2(v_i^{*\perp})^T W_i^{\perp} a_i + a_i^T (W_i^{\perp})^T W_i^{\perp} a_i \right] dt, \tag{3.18}$$

where $\{a_i \in \mathbb{R}^M, i = 0, \ldots, K-1\}$. Other than a constant contribution from $(v_i^{*\perp})^T v_i^{*\perp}$, which is independent of $\{a_i\}$, we should choose $\{a_i\}$ via

$$\min_{\{a_i\}} \sum_{i=0}^{K-1} 2d_i^T a_i + a_i^T C_i a_i. \tag{3.19}$$

The continuity of $v^{\perp}$ at t_i, across the interface between segment $i-1$ and i, can be written as

$$v_{i-1}^{*\perp}(t_i) + W_{i-1}^{\perp}(t_i) a_{i-1} = v_i^{*\perp}(t_i) + W_i^{\perp}(t_i) a_i. \tag{3.20}$$

By applying Equation (3.12), (3.13), and (3.16), we can show this is equivalent to:

$$a_i = R_{i-1} a_{i-1} + b_{i-1}. \tag{3.21}$$

Combining the minimization problem in Equation (3.19) and the continuity constraints in Equation (3.21), we obtain the NILSS problem for multiple time segments:

$$\begin{aligned} &\min_{\{a_i\}} \sum_{i=0}^{K-1} 2d_i^T a_i + a_i^T C_i a_i \\ &\text{s.t. } a_i = R_{i-1} a_{i-1} + b_{i-1} \quad i = 1, \ldots, K-1. \end{aligned} \tag{3.22}$$

Once $\{a_i\}$ is obtained, we can compute v_i within each time segment $t \in [t_i, t_{i+1}]$ via the expression

$$v_i(t) = v_i^*(t) + W_i(t) a_i. \tag{3.23}$$

3.7 Computing ξ_i

For each segment i, we define $\xi_i(t)$ by plugging v into Equation (2.11) to arrive at

$$\xi_i f = v_i - v_i^{\perp}. \tag{3.24}$$

In fact, we only need to know the value of ξ_i at the beginning and end of each segment, that is:

$$\begin{aligned} \xi_i(t_i) &= 0\ ; \\ \xi_i(t_{i+1}) &= \frac{(v_i(t_{i+1}))^T f(u(t_{i+1}))}{f(u(t_{i+1}))^T f(u(t_{i+1}))}\ . \end{aligned} \tag{3.25}$$

In Equation 3.25, we used the fact that at the beginning of each segment, v_i^* and W_i are in $V^{\perp}$, hence so is v_i.

3.8 Computing $d\langle J\rangle_\infty / ds$

Once $v(t)$ is obtained, $d\langle J\rangle_\infty / ds$ is computed via

$$\frac{1}{T}\sum_{i=0}^{K-1}\left[\int_{t_i}^{t_{i+1}} (\partial_u J\, v_i + \partial_s J)\, dt + \xi_i(t_{i+1})(\langle J\rangle_T - J(t_{i+1}))\right]. \tag{3.26}$$

The derivation of Equation (3.26) from Equation (2.12) is in Appendix E.

Alternatively, the sensitivity can be computed without explicitly determining $\{v_i(t)\}$. The sensitivity contribution of each $v_i(t)$ can be computed from a linear combination of the contributions of v_i^* and $w_{i,j}$, with a_i being the coefficients.

Chapter 4

Numerical Results

In this chapter we apply NILSS to two dynamical systems. The first is the Lorenz 63 system, which is a PDE with dimension $m = 3$. The second is a CFD simulation of a chaotic flow over a backward-facing step, which has dimension $m = 36675$. The second problem shows that NILSS can be used in engineering applications.

4.1 Numerical Results on Lorenz attractor

We apply NILSS to the Lorenz 63 system. There are three states $u = [x, y, z]$, so $m = 3$. The governing equation is:

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z. \tag{4.1}$$

In our current numerical example, $\sigma = 10, \beta = 8/3$.

The parameter of the system is ρ, which varies in range $[2, 45]$. The Lorenz 63 system has different behaviors when ρ changes [28]:

- $2 \leq \rho < 24.7$, two fixed-point attractors.
- $24.7 \leq \rho < 31$, one quasi-hyperbolic strange attractor.
- $31 \leq \rho \leq 45$, one non-hyperbolic attractor.

In none of these cases the dynamical system strictly satisfies our assumptions that there exists a full set of CLVs for all states on the attractor; however, as we shall see, NILSS still gives meaningful results.

The instantaneous objective function is $J(u) = z$, so the objective is:

$$\langle J \rangle_\infty = \lim_{T \to \infty} \frac{1}{T} \int_0^T z \, dt \,. \tag{4.2}$$

We use $\langle J \rangle_{T'}$ to approximate $\langle J \rangle_\infty$, where $T' = 500$ time units. Moreover, the initial state u_0 of each ρ is randomized.

When solving the primal solution $u = (x, y, z)^T$, we use RK-4 with time step size 0.01. Each segment has 200 steps, or 2 time units. We perform NILSS over $K = 50$ segments, i.e., $T = 100$ time units.

The LEs of the Lorenz 63 system should satisfy the following constraints [10]:

$$\begin{aligned} \lambda_1 + \lambda_2 + \lambda_3 &= -(1 + \sigma + \beta); \\ \lambda_3 &= 0. \end{aligned} \tag{4.3}$$

Here λ_3 is the LE whose corresponding CLV is parallel to du/dt. Since $\lambda_1 + \lambda_2 < 0$, there are at most 1 positive LE. Hence we set the number of homogeneous solutions to be $M = 1$.

With above setting, we compute $\langle J \rangle_\infty$ and $d \langle J \rangle_\infty / d\rho$. The results are shown in Figure 4-1 and Figure 4-2. The flaw shown in Figure 4-1 is also observed in other numerical results such as those found in [20]. This flaw corresponds to the onset of chaos around $\rho = 24.7$. For smaller ρ, the system has two fixed-points, and the sensitivity results, via NILSS, show no oscillation. When the system develops into chaos, the sensitivity results begin to oscillate because, on a finite trajectory, they depend on the random-valued initial conditions u_0 and $W_0(0)$. Nevertheless, Figure 4-1 shows that the true value of $d \langle J \rangle_\infty / d\rho$ is approximately 1 for all ρ. The sensitivities computed with NILSS agree with this observation.

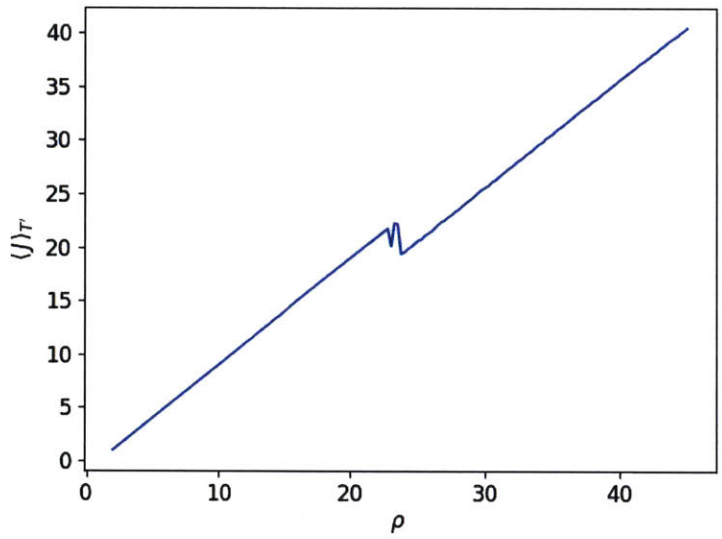

Figure 4-1: Averaged objective $\langle J \rangle_{T'}$ versus parameter ρ for the Lorenz 63 system, with $\sigma = 10, \beta = 8/3$, $T' = 500$ time units.

4.2 Numerical Results on CFD Simulation of flow over a backward-facing step

We apply NILSS to a chaotic flow over a backward-facing step. Specifically, we use the same geometry and mesh as in the PitzDaily tutorial of OpenFOAM 4.0, which is modeled from the experiment by Pitz and Daily [25]. This problem is a two-dimensional flow over a backward-facing step near the inlet and a contracting nozzle at the outlet. The geometry is shown in Figure 4-3.

For the numerical simulation, we use OpenFOAM 4.0 as the solver. We use the mesh provided in the tutorial: there are 12225 cells, as shown in Figure 4-4. We solve the incompressible Navier-Stokes equation via pisoFOAM. We use the second-order finite volume scheme and the time-integration method is PISO (Pressure Implicit with Splitting of Operator) with a time step size 1×10^{-5} second. We use dynamic one equation eddy-viscosity model as turbulence model [18]. The viscosity is $1 \times 10^{-5} m^2/s$.

We set no-slip wall conditions for all boundaries except for the inlet and outlet. The velocity at the inlet boundary takes a uniform fixed value in the x-direction, the norm of which is the parameter of this problem. For the base case, we set the

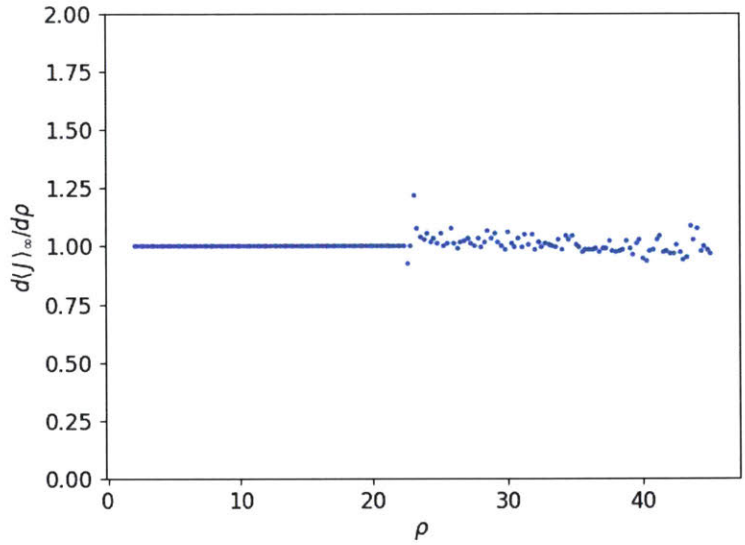

Figure 4-2: $d\langle J\rangle_\infty/d\rho$ computed for each ρ via NILSS. The time length of the trajectories is $T = 100$, which is partitioned into 50 segments of length 2. NILSS uses one homogeneous tangent solution.

inlet velocity to $U = (10, 0, 0)m/s$. For the outlet, we use the 'inletOutlet' option, which is to switch between the zero value and the zero gradient boundary condition, depending on the flow direction.

With the above settings, a typical snapshot of the flow field is shown in Figure 4-5. The flow is chaotic but not turbulent, since it is two-dimensional. Moreover, for a real-life problem, like the current one, there is no guarantee that all of our assumptions made when developing NILSS will be satisfied. However, as we shall see,

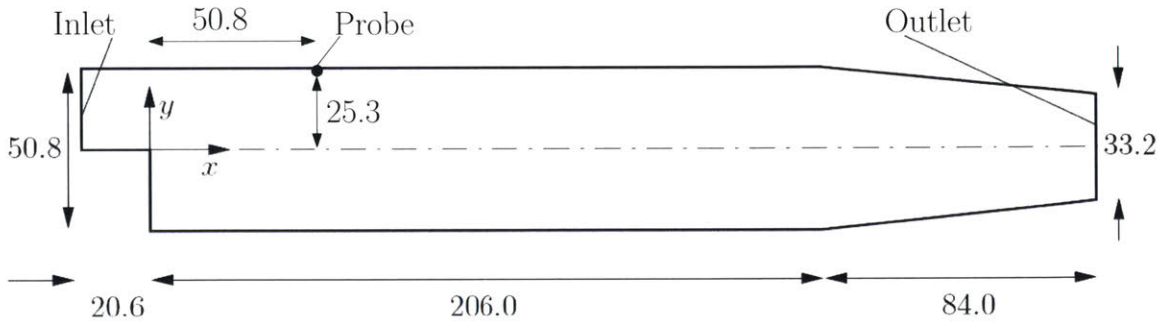

Figure 4-3: Geometry used in the simulation of a chaotic flow over a backward-facing step, dimensions in mm. All boundaries except inlet/outlet are solid walls.

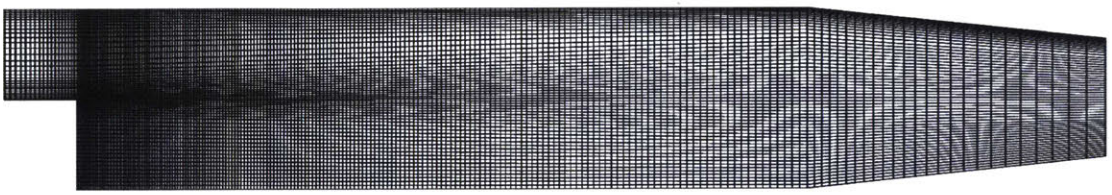

Figure 4-4: Mesh of test case, as provided in the tutorial of OpenFOAM 4.0

NILSS still gives meaningful results.

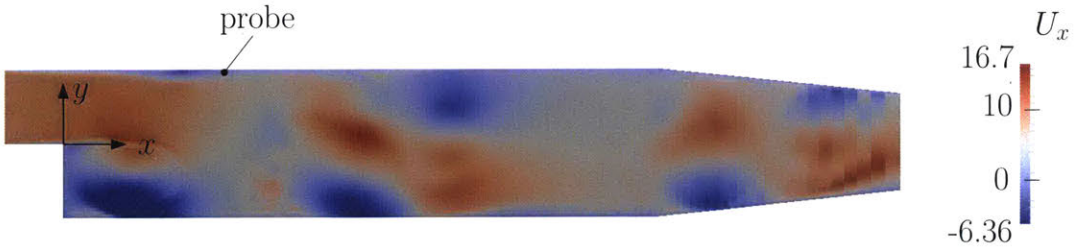

Figure 4-5: Flow field at time 0.091. Plotted by x-directional velocity U_x.

The parameter in this problem is the x-directional velocity at the inlet, U_{x0}. We use four different objectives: the long-time average of $U_x/10$, $(U_x/10)^2$, $(U_x/10)^4$, and $(U_x/10)^8$, where U_x is the x-direction velocity at a probe at coordinate (50.8 mm, 25.3 mm). The location of the probe is very close to the upper surface, as shown in Figure 4-5.

Each objective $\langle J \rangle_\infty$ is approximated by $\langle J \rangle_{T'}$, which is the average of the instantaneous objectives $J(t)$ over 2×10^5 time steps, or $T' = 2$ seconds. Since $J(t)$ exhibits aperiodic oscillations, $\langle J \rangle_{T'}$ has uncertainty. To get the uncertainty, we divide the history of $J(t)$ into 5 equally long parts in time.. Denote the objectives averaged over each of the five parts by $J_1, ...J_5$. The corrected sample standard deviation between them are:

$$\sigma' = \sqrt{\frac{1}{4}\sum_{k=1}^{5}(J_k - \langle J \rangle_{T'})^2}. \tag{4.4}$$

Here we assume that the standard deviation of $\langle J \rangle_{T'}$ is proportional to $T'^{-0.5}$. Thus, we use $\sigma = \sigma'/\sqrt{5}$ as the standard deviation of $\langle J \rangle_{T'}$. We further assume $\pm 2\sigma$ yields the 95% confidence interval for $\langle J \rangle_{T'}$. Objectives for different parameters in the range

[9,11] are shown in the right column of Figure 4-7, where the bars indicate the 95% confidence interval.

We use the finite difference NILSS as explained in Appendix D. Each segment has 250 time steps, or a period of 0.0025 second. To compute the sensitivity, we run NILSS over $K = 200$ segments, or $T = 0.5$ seconds.

To determine the number of homogeneous solutions, M, we compute LEs by the method described in Section 3.2. For a particular LE, denoted by λ, its computed value changes with the length of the trajectory, or the number of segments, provided that the segment length ΔT is fixed. We use λ_i to denote the LE value computed using data from segments $1, 2, ..., i$. To determine the uncertainty in the computed LE, we compute the smallest interval that converges at rate $i^{-0.5}$ and contains all $\{\lambda_i\}$. Specifically, we assume that $\{\lambda_i\}$ converges to some λ_0 as we increase i and its confidence interval is proportional to $i^{-0.5}$. To find λ_0, we first define $C(\lambda)$ as:

$$C(\lambda) = \min\{C' \mid |\lambda - \lambda_i| \leq C' i^{-0.5}, \text{for all } i \leq K\}, \tag{4.5}$$

where K is the number of segments. We define λ_0 as such that the corresponding $C(\lambda_0)$ is smallest:

$$\lambda_0 = \arg\min_{\lambda}\{C(\lambda)\}. \tag{4.6}$$

We regard $CK^{-0.5}$ as the confidence interval for λ_0. The convergence history of the largest 16 LEs are shown in the left of Figure 4-6. The λ_0 and confidence intervals for each LE are shown in the right of Figure 4-6. The total number of positive LEs is smaller than 16. So we set $M = 16$.

By using the settings listed above, the cost of NILSS is mainly in integrating the primal solution over $200 \times 250 \times 18 = 9 \times 10^5$ time steps. Here 200 is the number of segments, 250 is the number of time steps in each segment, and 18 is the number of primal solutions computed. In finite difference NILSS, we need one v^* and 16 $\{w_j\}$. Each tangent solution is approximated by a finite difference between a perturbed solution and the same base solution: that is 18 primal solutions in total.

We want to give confidence intervals for the sensitivities computed by NILSS.

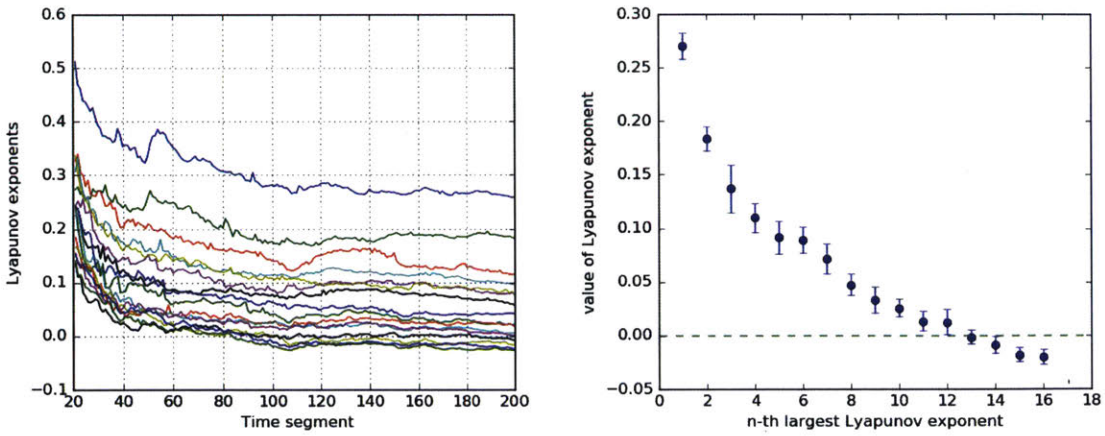

Figure 4-6: Lyapunov exponents (LE). Left: the convergence history of 16 different LEs as the trajectory length increases, where the trajectory length is represented by the number of segments. Right: confidence interval of the largest 16 LEs.

Similar to the case of LE, the value of dJ/ds changes with T, or equivalently, the number of segments. We use $(dJ/ds)_i$ to denote the sensitivity computed using data from segments $1, 2, ..., i$. In this case, we assume that $\{(dJ/ds)_i\}$ converges to some $(dJ/ds)_0$ as we increase i, and its confidence interval is proportional to $i^{-0.5}$. To find $(dJ/ds)_0$, we first define $C(dJ/ds)$ as

$$C\left(\frac{dJ}{ds}\right) = \min\left\{C' \;\middle|\; \left|\frac{dJ}{ds} - \left(\frac{dJ}{ds}s\right)_i\right| \leq C' i^{-0.5}, \text{for all } i \leq K\right\}. \tag{4.7}$$

We define $(dJ/ds)_0$ such that the corresponding $C((dJ/ds)_0)$ is the smallest:

$$\left(\frac{dJ}{ds}\right)_0 = \arg\min_{dJ/ds}\left\{C\left(\frac{dJ}{ds}\right)\right\}. \tag{4.8}$$

We regard $CK^{-0.5}$ as the confidence interval for $(dJ/ds)_0$. The left column in Figure 4-7 is a log-log plot of $|(dJ/ds)_0 - (dJ/ds)_i|$ versus i for $U_{x0} = 10$, where the lines indicate $Ci^{-0.5}$. Similarly, we find the confidence interval of the sensitivity at $U_{x0} = 11$. In the right column of Figure 4-7, the wedges indicate the confidence intervals of the sensitivities.

As we can see in Figure 4-7, in the last three rows, the sensitivities computed

by NILSS correctly reflect the trend in long-time averaged objectives. However, for the first row, the averaged objectives themselves have large uncertainties. This is because a function oscillating near zero usually has large variance in comparison to its average. In this scenario, since the primal simulation does not suggest a trend, we cannot tell if NILSS gives a meaningful derivative.

In our current example, the cost of NILSS is roughly the same as that of the conventional finite difference method. For chaotic systems, with fixed u_0 and T', the relation $\langle J \rangle_{T'} \sim s$ has many local fluctuations [34]. To smooth out these local fluctuations, we perform a linear regression over 5 parameters within the interval [9,11]. In the conventional finite difference method, the total cost comes from integrating the primal system for $5 \times 2 \times 10^5 = 1 \times 10^6$ steps. This cost is similar to NILSS, which integrates for 9×10^5 steps.

However, here we may be making a comparison in favor of the conventional finite difference. In Figure 4-7, the range span of parameters is 2; it is too large for the last two objectives, since the relations between objectives and parameters are not linear. In these cases, if we want to reduce the error in linearly approximating a nonlinear function, the parameter range should be smaller. However, this requires the confidence intervals of the objectives to be reduced as well. Otherwise, the uncertainties in the objectives are divided by a smaller parameter range; this would give rise to larger uncertainties in the sensitivities. To obtain smaller confidence intervals for the objectives, we require longer trajectories, which means larger computational cost for the conventional finite difference method.

When there are multiple parameters, the cost of NILSS is even lower than the conventional finite difference method. For a tangent NILSS, Equation (2.2) has a right-hand side $\partial_s f$, which states that v^* would change if we have a new parameter; however, w_j does not depend on the parameter s, so they could be reused for the new parameter. The marginal cost of adding a new parameter is only the cost to compute a new v^*. In our finite difference NILSS for this problem, 18 trajectories were computed: one is a base trajectory, one has a perturbed parameter, 16 have perturbed initial conditions. Only the trajectories with perturbed parameter should be recomputed

for an additional parameter. So the marginal cost of another parameter is only 1/18 of the cost of the first parameter. On the other hand, for the conventional finite difference method, 5 trajectories are computed: one is a base trajectory and 4 have perturbed parameters. As a result, 4 trajectories should be recomputed for a new choice of parameter. This suggests that the marginal cost of another parameter is 4/5 of the cost of the first parameter, which is higher than that of finite difference NILSS.

The cost of NILSS is lower than that of the conventional LSS method. The number of states in our problem is $12225 \times 3 = 36675$. If we perform the conventional LSS over the same time span of 5×10^4 steps, the LSS method would require solving a linear equation system with 1.8×10^9 variables. This would be a very large cost in both computation time and computer storage.

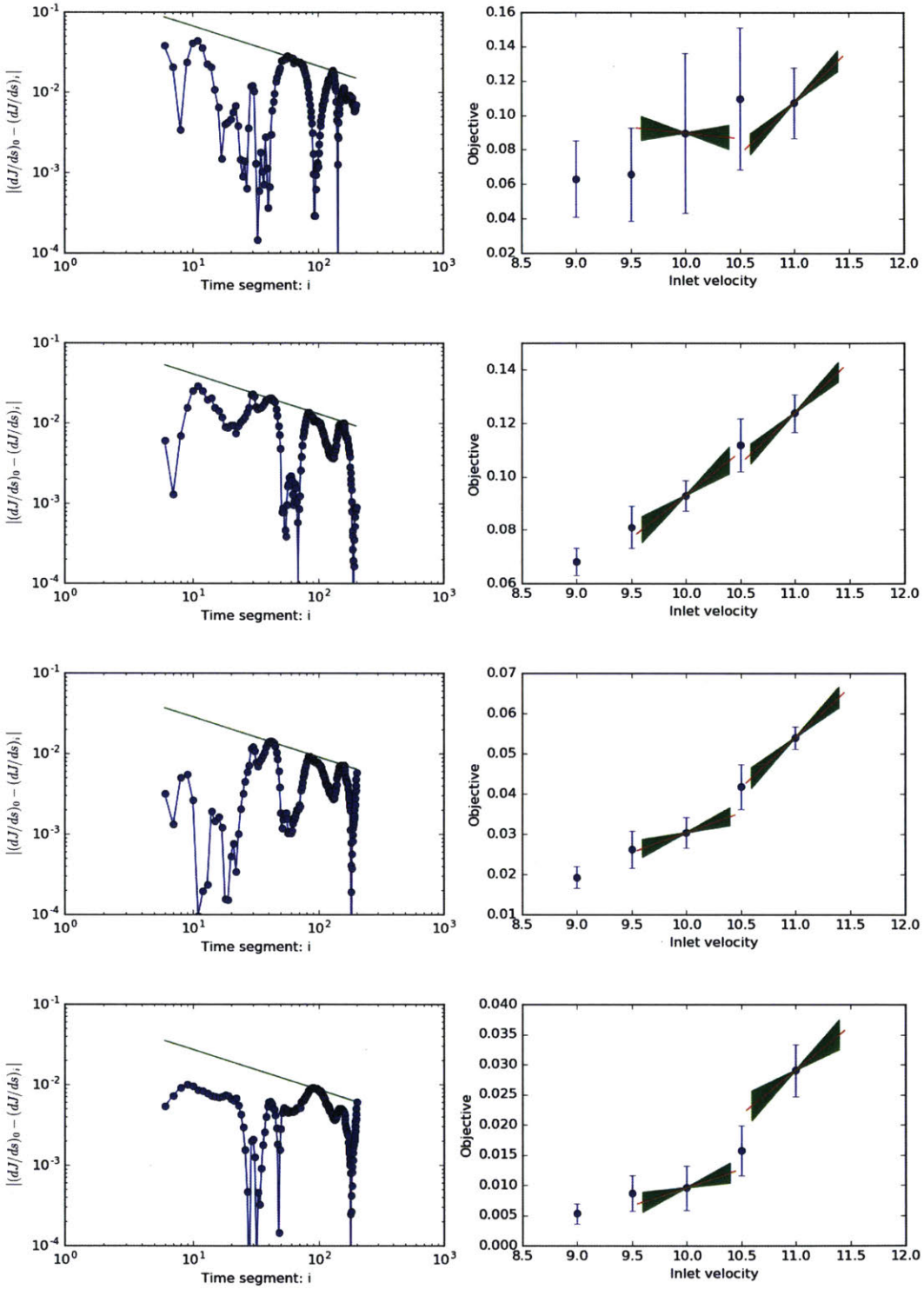

Figure 4-7: Sensitivity computed by NILSS. From top to bottom, the objective function is the long-time average of $U_x/10$, $(U_x/10)^2$, $(U_x/10)^4$, and $(U_x/10)^8$. Left column: sensitivity computed by an increasing number of segments, the lines indicates confidence intervals for sensitivities. Right column: sensitivity plotted with objectives for adjacent parameters, the bars and wedges indicate confidence intervals of the objectives and sensitivities, respectively.

Chapter 5

Conclusions

The Least Squares Shadowing (LSS) method [34] can compute meaningful sensitivities of long-time averaged quantities in chaotic systems. In comparison to other approaches, the advantage of LSS is in its relative simplicity and robustness. LSS gives one of the first applications of sensitivity analysis on CFD problems. However, the cost of LSS is still too high for real-life engineering problems, where the dimension of the system can be hundreds of thousands.

This thesis develops the Non-Intrusive Least Squares Shadowing (NILSS) method. In NILSS, we want to represent the transient effect by homogeneous tangent solutions, and subtract it from the conventional inhomogeneous tangent, which represents the two effects of the parameter perturbation. The leftover long-time effect can then be used to compute the sensitivity of interest. The homogeneous tangent we want can be constructed as the linear combination of unstable modes, which can be approximated by the same number of homogeneous tangents with random initial conditions. Then we solve the NILSS problem, which is to minimize the L^2 norm of a linear combination among the conventional inhomogeneous tangent and several homogeneous tangents. This minimization finds the inhomogeneous tangents that represents the long-time effect.

The first advantage of NILSS is in its computational cost. Now the minimization problem in NILSS is small and almost costless; the main cost is in preparing the minimization problem, that is, to compute the conventional inhomogeneous tangent

and several homogeneous tangents. Hence for problems with a few unstable modes, which is the case for many engineering applications, NILSS has low computational cost.

The major coding workload for implementing NILSS is to have a homogeneous tangent solver which takes arbitrary initial conditions. Although most linear solvers are inhomogeneous and takes only zero initial conditions, these changes could be done relatively easily. Specifically, NILSS does not involve manipulating information that most existing linear solvers do not provide, such as the Jacobian matrix. Moreover, for the finite difference NILSS, only a primal simulation software is required, and we even do not need a linear solver.

NILSS has been demonstrated on the Lorenz 63 system and a CFD simulation for a flow over a backward-facing step. In both cases, the sensitivities provided by NILSS reflects the trend between objectives and parameters. For the latter case, NILSS has a similar computational cost as the conventional finite difference method. We further argue that NILSS would be computationally cheaper than the conventional finite difference method if the relationship between objectives and parameters is nonlinear or if we are interested in multiple parameters. We also verified that the latter test case has a low-dimensional attractor, with less than 16 positive Lyapunov exponents.

There are many avenues for future work of NILSS. For example, we should extend our theory to systems which are not uniform-hyperbolic or not ergodic. On the application side, the adjoint formulation of NILSS is of interest. Nevertheless, NILSS reduces the computational cost to the same level of the primal simulation, which gives it potential to make sensitivity analysis accessible to real-life engineering applications.

Appendix A

Showing $f(u)$ is a CLV with a zero LE

We assume the attractor Λ is bounded with a positive lower bound for $f(u)$, i.e., there exists $C_1^0 > 0$, such that

$$\|f(u)\| \geq C_1^0, \quad \text{for all } u \in \Lambda. \tag{A.1}$$

Since $f(u, s)$ is a continuous function, then $f(u)$ is continuous for fixed s. Together with the assumption that Λ is bounded, we see that the $f(\Lambda)$ is also bounded, i.e., there exists $C_2^0 > 0$, such that

$$\|f(u)\| \leq C_2^0, \quad \text{for all } u \in \Lambda. \tag{A.2}$$

We check that for a fixed s, $f(u)$ is a homogeneous tangent solution that satisfies

$$\frac{df(u)}{dt} = \frac{\partial f}{\partial u}\frac{du}{dt} = \partial_u f f, \tag{A.3}$$

where the last equality is due to Equation (1.1).

Next, we denote $C_1 = C_1^0/\|f(u(0))\|$, $C_2 = C_2^0/\|f(u(0))\|$, then $f(u)$ is a CLV whose LE is 0, since

$$C_1 e^{0t}\|f(u(0))\| \leq \|f(u(t))\| \leq C_2 e^{0t}\|f(u(0))\|, \tag{A.4}$$

which satisfies Equation (2.7).

Appendix B

Showing $\{\|\zeta_j^{\perp}\|\}$ behave like exponentials

Here we show that the norm of the orthogonal projection of stable and unstable modes, $\{\|\zeta_j^{\perp}\|\}$, behave like exponentials.

We assume that all CLVs are uniformly bounded away from each other. First, we define the angle $\alpha_{ij}(u)$ between two CLVs,

$$\alpha_{ij}(u) = \arccos \frac{\zeta_i(u)^T \zeta_j(u)}{\|\zeta_i(u)\| \|\zeta_j(u)\|}, \quad i \neq j. \tag{B.1}$$

The assumption means that there is an $\alpha_0 > 0$ such that:

$$\alpha_{ij}(u) > \alpha_0, \quad \text{for all } i \neq j, u \in \Lambda\ , \tag{B.2}$$

where Λ is the attractor.

Since $f(u)$ is also a CLV, the angles between $\{\zeta_j\}$ and $f(u)$ are all greater than α_0 and the angles between $\{\zeta_j\}$ and $V^{\perp}$ are smaller than $\pi/2 - \alpha_0$. Hence, by using the C_1 and C_2 provided by Equation (2.7), we arrive at

$$\|\zeta_j^{\perp}(u(t))\| \geq \sin(\alpha_0)\|\zeta_j(u(t))\| \geq \sin(\alpha_0) e^{\lambda_j t} C_1 \|\zeta_j(u(0))\| \geq C_1' e^{\lambda_j t} \|\zeta_j^{\perp}(u(0))\|\ , \tag{B.3}$$

where $C_1' = \sin(\alpha_0)C_1$. On the other hand, we know that

$$\|\zeta_j^{\perp}(u(t))\| \leq \|\zeta_j(u(t))\| \leq C_2 e^{\lambda_j t}\|\zeta_j(u(0))\| \leq C_2' e^{\lambda_j t}\|\zeta_j^{\perp}(u(0))\| \,, \tag{B.4}$$

where $C_2' = \sin(\alpha_0)C_2$. To summarize, there is $C_1', C_2' > 0$, such that

$$C_1' e^{\lambda_j t}\|\zeta_j^{\perp}(u(0))\| \leq \|\zeta_j^{\perp}(u(t))\| \leq C_2' e^{\lambda_j t}\|\zeta_j^{\perp}(u(0))\| \,. \tag{B.5}$$

Here all $\lambda_j \neq 0$, since they correspond to either stable or unstable modes, but not the neutral mode.

Appendix C

Derivation of $d\langle J\rangle_\infty/ds$

By applying an infinitesimal perturbation in s, the governing equation for u is:

$$\frac{d(u+\delta u)}{dt} = f(u+\delta u, s+\delta s)\ . \tag{C.1}$$

After subtracting it by the unperturbed ODE, we get the governing equation for δu

$$\frac{d(\delta u)}{dt} = \partial_u f \delta u + \partial_s f \delta s\ . \tag{C.2}$$

As shown in fig C-1, we assume that at time t, the difference of the new trajectory from the original one is itself perpendicular to f, or $\delta u(t) = \delta u^\perp(t)$. After δt, this difference is no longer perpendicular to f, and thus it becomes

$$\delta u(t+\delta t) = \delta u^\perp(t) + (\partial_u f \delta u^\perp(t) + \partial_s f \delta s)\delta t\ . \tag{C.3}$$

We denote the projection of $\delta u(t+\delta t)$ onto the direction of $f(u(t+\delta t))$ by $-\eta f \delta t \delta s$, or

$$-\eta f \delta t \delta s = \frac{f^T\left[\delta u^\perp(t+\delta t)\right]}{f^T f} f\ . \tag{C.4}$$

On the other hand, the projection of $\delta u(t+\delta t)$ onto $V^\perp$ is denoted by $\delta u^\perp(t+\delta t)$, as defined in Equation (2.5). Thus, in Equation (C.3), $\delta u(t+\delta t)$ can be represented

as the summation of two orthogonal projections:

$$\delta u^{\perp}(t) + (\partial_u f \delta u^{\perp}(t) + \partial_s f \delta s)\delta t = \delta u^{\perp}(t + \delta t) - \eta f \delta t \delta s \ . \tag{C.5}$$

We recall our definition that $v = \delta u/\delta s$, $v^{\perp} = \delta u^{\perp}/\delta s$, we obtain:

$$\frac{dv^{\perp}}{dt} = \partial_u f v^{\perp} + \partial_s f + \eta f \ . \tag{C.6}$$

Here v is the tangent solution of Equation (2.2); $v^{\perp}$ is the orthogonal projection of v according to Equation (2.5). Only η is unknown, so we can also view Equation (C.6) as the definition of η.

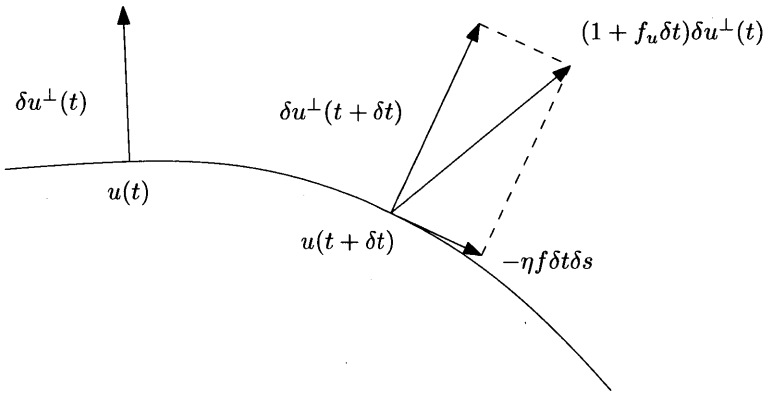

Figure C-1: Perturbation of the trajectory due to a perturbation on the parameter.

Recall ξ is the scalar such that $\xi f = v - v^{\perp}$, as defined in Equation (2.11). We can show that:

$$\eta = -\frac{d\xi}{dt} \ , \tag{C.7}$$

To see this, first subtract Equation (C.6) from (2.2). This yields

$$\frac{d(v - v^{\perp})}{dt} = \partial_u f (v - v^{\perp}) - \eta f \ . \tag{C.8}$$

Using our definition of ξ, we arrive at

$$\frac{d(\xi f)}{dt} = \partial_u f (\xi f) - \eta f \ . \tag{C.9}$$

By the rule for differentiating the product of two functions,

$$\frac{d(\xi f)}{dt} = \xi \frac{df}{dt} + \frac{d\xi}{dt} f \ . \tag{C.10}$$

Equation (C.7) is obtained by recalling the chain rule for the differential:

$$\partial_u f(\xi f) = \xi \ \partial_u f(f) = \xi(\partial_u f \frac{du}{dt}) = \xi \frac{df}{dt} \ . \tag{C.11}$$

To know the difference between the perturbed trajectory and the base trajectory, we need to define a correspondence between the states on the two trajectories. That is, we should define which state on the base trajectory should be subtracted by which state on the perturbed trajectory.

Instead of comparing the two trajectories in the same time frame, we vary the length of infinitesimal time steps so that the corresponding states of the two trajectories remain perpendicular to f. In time δt, the new trajectory moves a length of $f\delta t - \eta f \delta t \delta s$. So the new speed is $(1 - \eta\delta s)f$. Hence the new trajectory needs time $\delta t/(1 - \eta\delta s) \approx \delta t(1 + \eta\delta s)$ to cross length $f\delta t$, which is the length of the base trajectory. If we compare the point on base trajectory at time $(t+\delta t)$ with the point on the perturbed trajectory at time $t+\delta t(1+\eta\delta s)$, their difference will remain perpendicular to f, which is $\delta u^{\perp}(t + \delta t)$.

The $J_{new}\delta t_{new}$ on this small section of new trajectory is:

$$\begin{aligned} & J_{new}\delta t_{new} \\ =&(J + \partial_u J\delta u^{\perp})(1 + \eta\delta s)\delta t \\ =&J\delta t + \partial_u J\delta u^{\perp}\delta t + J\eta\delta s\delta t \ . \end{aligned} \tag{C.12}$$

To compute the difference between the average J, we first write down its definition:

$$\begin{aligned}
&\frac{1}{T_{new}}\int_0^{T_{new}} J_{new}dt - \frac{1}{T}\int_0^T Jdt \\
=&\frac{1}{\int_0^T(1+\eta\delta s)dt}\int_0^T \left(J + \partial_u J\delta u^\perp + J\eta\delta s\right)dt - \frac{1}{T}\int_0^T Jdt \\
=&\frac{\delta s}{T}\int_0^T \left[\partial_u J\, v^\perp + \partial_s J + \eta(J - \langle J\rangle)\right]\, dt\,,
\end{aligned} \tag{C.13}$$

where we used the definition $\delta u^\perp(t) = v^\perp \delta s$. If we divide by δs, we arrive at:

$$\frac{d}{ds}\left(\frac{1}{T}\int_0^T Jdt\right) = \frac{1}{T}\int_0^T \left[\partial_u J\, v^\perp + \partial_s J + \eta(J - \langle J\rangle)\right]\, dt\,. \tag{C.14}$$

Notice that here the ending time T also depends on s.

First we use the shadowing direction v^∞ as v in equation (C.14). Since $v^{\infty\perp}(u)$ is uniformly bounded for all u on the attractor, we can interchange the procedure of differentiating by s and letting T go to infinity:

$$\begin{aligned}
\frac{d}{ds}\langle J\rangle_\infty &= \frac{d}{ds}\left(\lim_{T\to\infty}\frac{1}{T}\int_0^T Jdt\right) \\
&= \lim_{T\to\infty}\frac{1}{T}\int_0^T \left[\partial_u J\, v^{\perp\infty} + \partial_s J + \eta(J - \langle J\rangle)\right]\, dt\,,
\end{aligned} \tag{C.15}$$

where η is computed by substituting v^∞ into Equation (C.6). In fact, it is exactly the commutation between differentiation and T going to infinity that requires $v^{\infty\perp}$ to be uniformly bounded. The mathematical proof that justifies the interchange of two procedures can be found in [33, 11].

For infinite T, only $v^{\infty\perp}$ can make C.15 holds. However, for finite T, we can use the NILSS solution v to approximate v^∞ and arrive at:

$$\frac{d\langle J\rangle_\infty}{ds} \approx \frac{1}{T}\int_0^T \left[\partial_u J\, v^\perp + \partial_s J + \eta(J - \langle J\rangle_T)\right]\, dt\,. \tag{C.16}$$

The proof of this approximation can be accomplished similarly to that in [33, 11].

We can replace the requirement for computing η, by computing ξ at the two ends

of the trajectory. To achieve this, we first apply Equation (C.7) and integrate by parts:

$$\begin{aligned}\frac{d\langle J\rangle_\infty}{ds} &\approx \frac{1}{T}\int_0^T \left[\partial_u J\, v^\perp + \partial_s J - \frac{d\xi}{dt}(J - \langle J\rangle_T)\right] dt \\ &= \frac{1}{T}\left[\int_0^T \left(\partial_u J\, v^\perp + \partial_s J\right) dt - (\xi J)\,|_0^T + \xi|_0^T \langle J\rangle_T + \int_0^T \xi \frac{dJ}{dt} dt\right].\end{aligned} \tag{C.17}$$

Next, we apply the fact that

$$\frac{dJ}{dt} = \partial_u J \frac{du}{dt} = \partial_u J\, f, \tag{C.18}$$

and that $v = v^\perp + \xi f$ into Equation (C.17). Thus, we have:

$$\begin{aligned}\frac{d\langle J\rangle_\infty}{ds} &\approx \frac{1}{T}\left[\int_0^T \left(\partial_u J\, v^\perp + \partial_s J + \xi \partial_u J f\right) dt - (\xi J)\,|_0^T + \xi|_0^T \langle J\rangle_T\right], \\ &= \frac{1}{T}\left[\int_0^T \left(\partial_u J\, v + \partial_s J\right) dt + \xi|_0^T \langle J\rangle_T - (\xi J)|_0^T\right]\end{aligned} \tag{C.19}$$

This is exactly Equation (2.12).

Appendix D

Finite difference NILSS method

We can use finite difference results to approximate all the tangent solutions used in NILSS, which consist of v^* and $W = \{w_j\}_{j=1}^M$. Once their approximations are obtained, we can compute the NILSS solution v and then use this v to compute sensitivity. To achieve this, we first compute a baseline primal solution u_b, which satisfies Equation (1.1) with the initial condition u_0 on the attractor.

To approximate the homogeneous solution w with initial condition w_0, we compute primal solution u^w by keeping the same s but using initial conditions $u_0 + \epsilon w_0$. The approximation for w is thus

$$w \approx \frac{u^w - u_b}{\epsilon} . \tag{D.1}$$

For NILSS on a single time segment, v^* is the conventional inhomogeneous tangent which has a zero initial condition. To approximate it, we first change s to $s+\epsilon$, where ϵ is a small number. Then, we compute primal solution u^*, which satisfies the perturbed governing equation with the same initial condition u_0. The approximation for v^* is thus

$$v^* \approx \frac{u^* - u_b}{\epsilon} . \tag{D.2}$$

For NILSS over multiple time segments, v_i^* can have non-zero initial conditions for segments $i \neq 0$. To approximate v_i^* with initial condition v_{i0}^*, we compute primal solution u_i^* with parameter $s + \epsilon$ and initial condition $u_0 + \epsilon w_0$. The approximation

for v_i^* is thus

$$v_i^* \approx \frac{u_i^* - u_b}{\epsilon} . \tag{D.3}$$

The benefit of this finite difference version of NILSS is that it is truly non-intrusive. In fact, it no longer requires a tangent solver, all it needs is a simulation software which can solve the primal solution.

The downside is that, the approximation by finite difference may incur additional error. Also, when deciding the time segment length ΔT, there is an extra requirement: the perturbation cannot grow out of the linear region, in which case the finite difference no longer approximates the tangent solution.

Appendix E

Derivation of $d\langle J\rangle_\infty/ds$ on multiple segments

To derive Equation (3.26) from Equation (2.12), first we recover a continuous tangent solution v from $\{v_i^\perp\}$ and $\{\xi_i\}$ on each segment:

$$v(t) = v^\perp(t) + \xi(t) f(t) \,, \tag{E.1}$$

where

$$\begin{cases} v^\perp(t) = v_i^\perp(t), \\ \xi(t) = \xi_i(t) + \sum_{i'=0}^{i-1} \xi_{i'}(t_{i'+1}), \end{cases} \qquad t \in [t_i, t_{i+1}] \,, \tag{E.2}$$

where $\{v_i^\perp(t)\}$ are given by Equation (3.23), $\xi_i(t)$ are given by Equation (3.24). The definition of ξ can be viewed as 'accumulating' ξ_i from previous segments. Applying this definition, we have:

$$\xi(0) = 0, \quad \xi(T) = \sum_{i=0}^{K-1} \xi_i(t_{i+1}). \tag{E.3}$$

The continuity of v follows from the continuity of $v^\perp$ and ξ. $v^\perp(t)$ is continuous because of the continuity condition in Equation (3.20). $\xi(t)$ is continuous because $\xi_i(t_i) = 0$, as shown in Equation (3.25).

To see that v is a tangent solution of Equation (2.2), we first notice that on

segment i, v(t) is characterized by

$$v(t) = v_i(t) + \xi_i^* f(t), \quad t \in [t_i, t_{i+1}], \tag{E.4}$$

where $\xi_i^* = \sum_{i'=0}^{i-1} \xi_{i'}(t_{i'+1})$. Taking the time derivative of v, we have:

$$\begin{aligned} \frac{dv}{dt} &= \frac{dv_i}{dt} + \xi_i^* \frac{df}{dt} = \partial_u f v_i + \partial_s f + \xi_i^* \partial_u f f \\ &= \partial_u f (v_i + \xi_i^* f) + \partial_s f = \partial_u f v + \partial_s f. \end{aligned} \tag{E.5}$$

To conclude, v is a continuous tangent solution over the entire trajectory and the L^2 norm of $v^\perp$ is minimized, i.e., v is the solution of NILSS problem on a single time segment. Hence we can substitute v into Equation (2.12), which means that, together with Equation (E.3), we obtain:

$$\begin{aligned} \frac{d\langle J\rangle_\infty}{ds} &\approx \frac{1}{T}\left[\int_0^T (\partial_u J\, v + \partial_s J)\, dt + \xi|_0^T \langle J\rangle_T - (\xi J)|_0^T\right] \\ &= \frac{1}{T}\left[\sum_{i=0}^{K-1} \int_{t_i}^{t_{i+1}} (\partial_u J\, v_i + \partial_s J + \xi_i^* \partial_u J\, f)\, dt\right] + \frac{1}{T}\left[\xi(T)(\langle J\rangle_T - J(T))\right] \\ &= \frac{1}{T}\left[\sum_{i=0}^{K-1} \int_{t_i}^{t_{i+1}} \left(\partial_u J\, v_i + \partial_s J + \xi_i^* \frac{dJ}{dt}\right) dt\right] + \frac{1}{T}\left[\xi(T)(\langle J\rangle_T - J(T))\right] \\ &= \frac{1}{T}\sum_{i=0}^{K-1}\left[\int_{t_i}^{t_{i+1}} (\partial_u J\, v_i + \partial_s J)\, dt + \xi_i(t_{i+1})(\langle J\rangle_T - J(t_{i+1}))\right]. \end{aligned} \tag{E.6}$$

This yields Equation (3.26).

Bibliography

[1] Rafail V Abramov and Andrew J Majda. Blended response algorithms for linear fluctuation-dissipation for complex nonlinear dynamical systems. *Nonlinearity*, 20(12):2793, 2007.

[2] Rafail V. Abramov and Andrew J. Majda. New Approximations and Tests of Linear Fluctuation-Response for Chaotic Nonlinear Forced-Dissipative Dynamical Systems. *Journal of Nonlinear Science*, 18(3):303–341, 2008.

[3] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*, 10, may 2001.

[4] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov Characteristic Exponents for smooth dynamical systems and for hamiltonian systems; A method for computing all of them. Part 2: Numerical application. *Meccanica*, 15(1):21–30, 1980.

[5] Thomas R Bewley. Flow control: new challenges for a new Renaissance. *Progress in Aerospace Sciences*, 37:21–58, 2001.

[6] Thomas R. Bewley, Parviz Moin, and Roger Temam. DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *Journal of Fluid Mechanics*, 447:179–225, 2001.

[7] Patrick Blonigan, Steven Gomez, and Qiqi Wang. Least Squares Shadowing for sensitivity analysis of turbulent fluid flows. In *52nd Aerospace Sciences Meeting*, pages 1–24, 2014.

[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, Qiqi Wang, Eric J Nielsen, and Boris Diskin. Least Squares Shadowing Sensitivity Analysis of Chaotic Flow around a Two-Dimensional Airfoil. In *54th AIAA Aerospace Sciences Meeting*, number January, pages 1–28, 2016.

[10] Jo Bovy. Lyapunov exponents and strange attractors in discrete and continuous dynamical systems. Technical report, KU Leuven University, Theoretical Physics Project, 2004.

[11] Mario Chater, Angxiu Ni, Patrick J. Blonigan, and Qiqi Wang. Least Squares Shadowing method for sensitivity analysis of differential equations. *sumitted to SIAM Journal on Numerical Analysis, arXiv:1509.02882*, 2015.

[12] Philippe Courtier, John Derber, Ron Errico, JF Louis, and T Vukićević. Important literature on the use of adjoint, variational methods and the Kalman filter in meteorology, oct 1993.

[13] EH Dowell. Flutter of a buckled plate as an example of chaotic motion of a deterministic autonomous system. *Journal of Sound and Vibration*, pages 333–344, 1982.

[14] G L Eyink, T W N Haine, and D J Lea. Ruelle's linear response formula, ensemble adjoint schemes and Lévy flights. *Nonlinearity*, 17(5):1867, 2004.

[15] Krzysztof J Fidkowski and David L Darmofal. Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics.

[16] Michael B. Giles and Endre Süli. Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, 11, jan 2002.

[17] Antony Jameson. Aerodynamic design via control theory. *Journal of scientific computing*, 3(3):233–260, 1988.

[18] Won-Wook Kim and Suresh Menon. A new dynamic one-equation subgrid-scale model for large eddy simulations. In *AIAA, 33 rd Aerospace Sciences Meeting and Exhibit, Reno, NV*, 1995.

[19] A N Kolmogorov. The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Numbers. *Proceedings: Mathematical and Physical Sciences*, 434(1890):9–13, 1991.

[20] D J Lea, M R Allen, and T W N Haine. Sensitivity analysis of the climate of a chaotic system. *Tellus Series a-Dynamic Meteorology and Oceanography*, 52(5):523–532, 2000.

[21] C. E. Leith. Climate Response and Fluctuation Dissipation. *Journal of the Atmospheric Sciences*, 32(10):2022–2026, oct 1975.

[22] Edward N. Lorenz. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, mar 1963.

[23] T. N. Palmer. A nonlinear dynamical perspective on model error: A proposal for non-local stochastic-dynamic parametrization in weather and climate prediction models. *Quarterly Journal of the Royal Meteorological Society*, 127(572):279–304, 2001.

[24] Sergei Yu. Pilyugin. *Shadowing in Dynamical Systems*. Lecture Notes in Mathematics.

[25] R. W. Pitz and J.W. Daily. Combustion in a turbulent mixing layer formed at a rearward facing step. *AIAA Journal*, 21(11):1565–1570, 1983.

[26] James J Reuther, Antony Jameson, Juan J. Alonso, Mark J Rimllnger, and David Saunders. Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 2. *Journal of Aircraft*, 36(1):61–74, 1999.

[27] David Ruelle. Ergodic theory of differentiable dynamical systems. *Publications Math{é}matiques de l'Institut des Hautes {É}tudes Scientifiques*, 50(1):27–58, 1979.

[28] Colin Sparrow. *The Lorenz equations: bifurcations, chaos, and strange attractors*, volume 41. Springer Science & Business Media, 2012.

[29] Jean-Noéul Thepaut and Philippe Courtier. Four-dimensional variational data assimilation using the adjoint of a multilevel primitive-equation model. *Quarterly Journal of the Royal Meteorological Society*, 117(502):1225–1254, oct 1991.

[30] J. Thuburn. Climate sensitivities via a FokkerâĂŞPlanck adjoint approach. *Quarterly Journal of the Royal Meteorological Society*, 131(605):73–92, 2005.

[31] Jeroen Tromp, Carl Tape, and Qinya Liu. Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophys. J. Int.*

[32] Peter Walters. *An introduction to ergodic theory*, volume 79. Springer Science & Business Media, 2000.

[33] Qiqi Wang. Convergence of the Least Squares Shadowing Method for Computing Derivative of Ergodic Averages. *SIAM Journal on Numerical Analysis*, 52(1):156–170, 2014.

[34] Qiqi Wang, Rui Hu, and Patrick Blonigan. Least Squares Shadowing sensitivity analysis of chaotic limit cycle oscillations. *Journal of Computational Physics*, 267:210–224, 2014.

[35] Lai-Sang Young. What are SRB measures, and which dynamical systems have them? *Journal of Statistical Physics*, 108(5):733–754, 2002.