| Report Documentation Page | *Form Approved*<br>*OMB No. 0704-0188* |
|---|---|

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

| | | |
|---|---|---|
| 1. REPORT DATE<br>**13 AUG 2010** | 2. REPORT TYPE<br>**Final** | 3. DATES COVERED<br>**01-02-2007 to 30-11-2009** |

| | |
|---|---|
| 4. TITLE AND SUBTITLE<br>**Unsteady Aerodynamic Models for Flight Control of Agile Micro Air Vehicles** | 5a. CONTRACT NUMBER<br>**FA9550-07-1-0127** |
| | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S)<br>**Clarence Rowley** | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)<br>**Princeton University,Mechanical and Aerospace Engineering,Engineering Quadrangle,Princeton,NJ,08544** | 8. PERFORMING ORGANIZATION REPORT NUMBER<br>**; AFRL-OSR-VA-TR-2011-0251** |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)<br>**AFOSR, 875 North Randolph Street, Suite 325, Arlington, VA, 22203** | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S)<br>**AFRL-OSR-VA-TR-2011-0251** |

12. DISTRIBUTION/AVAILABILITY STATEMENT

**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT

**This project addresses the development of a hierarchy of models for unsteady aerodynamics, in a framework that is suitable for control design. The ultimate goal is to use these models for the design of flight controllers, for instance to accurately track a trajectory in the presence of large disturbances. Classical aerodynamic models by Theodorsen and Wagner were shown to be accurate in many cases, but break down at high angles of attack or for very rapid maneuvers. Methods were developed to perform balanced model reduction without the need for adjoint information, and for systems with periodic motions (such as vortex shedding). These were used to develop models of unsteady pitching and plunging maneuvers that are well-suited to control design techniques. The methods were developed for linearized models (small-amplitude motions) but also perform well for large-amplitude maneuvers.**

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT<br>**unclassified** | b. ABSTRACT<br>**unclassified** | c. THIS PAGE<br>**unclassified** | **Same as Report (SAR)** | **65** | |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std Z39-18

# Unsteady aerodynamic models for control of agile micro air vehicles

FA9550-07-1-0127

Clarence W. Rowley
Department of Mechanical and Aerospace Engineering
Princeton University

## Abstract

Small, lightweight aircraft are of increasing interest to the Air Force, for addressing potential threats in urban environments and other complex terrain. In order to avoid obstacles, respond to gusts, and track potentially elusive targets, highly agile maneuvers will be required, for which the standard quasi-steady aerodynamic models are not accurate. This project addresses the development of a hierarchy of models for unsteady aerodynamics, in a framework that is suitable for control design. The ultimate goal is to use these models for the design of flight controllers, for instance to accurately track a trajectory in the presence of large disturbances.

Classical aerodynamic models by Theodorsen and Wagner were shown to be accurate in many cases, but break down at high angles of attack or for very rapid maneuvers. Finite-time Lyapunov exponents (FTLE) were used to identify meaningful flow structures and elucidate the flow physics. Methods were developed to perform balanced model reduction without the need for adjoint information, and for systems with periodic motions (such as vortex shedding). These were used to develop models of unsteady pitching and plunging maneuvers that are well-suited to control design techniques. The methods were developed for linearized models (small-amplitude motions) but also perform well for large-amplitude maneuvers.

## Contents

**1 Executive summary** **3**
1.1 Techniques . . . . . 3
1.2 Summary of this report . . . . . 4

**2 Background: classical models** **5**
2.1 Theodorsen's frequency response . . . . . 5
2.2 Wagner's indicial response . . . . . 6

**3 Finite-time Lyapunov exponents for detecting breakdown of classical models** **7**
3.1 Fast computation of time-varying FTLE fields . . . . . 7
3.2 Breakdown of classical unsteady models . . . . . 16

**4 Nonlinear models for unsteady flows at fixed angle of attack** **23**
4.1 Phenomenological models . . . . . 23
4.2 Models using POD and Galerkin projection . . . . . 26

**5 Balanced model reduction** **30**
5.1 Balanced POD models without adjoints . . . . . . . . . . . . . . . . . . . . . . 31
5.2 Balanced truncation for periodic systems . . . . . . . . . . . . . . . . . . . . . 36

**6 Control-oriented extensions of classical unsteady models** **45**
6.1 Reduced-order approximations of Wagner's indicial response . . . . . . . . . . 46
6.2 Results: small-amplitude maneuvers . . . . . . . . . . . . . . . . . . . . . . . . 54
6.3 Results: large-amplitude maneuvers . . . . . . . . . . . . . . . . . . . . . . . . 56
6.4 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

# 1 Executive summary

Most models used for flight control in today's aircraft assume that the forces and moments on the aircraft are *quasi-steady:* they depend only on the velocity of the vehicle relative to the surrounding air. However, unsteady effects become increasingly important for Micro Air Vehicles, where rapidly changing gusts and large accelerations render the quasi-steady assumption invalid.

The purpose of this project is to obtain improved models for unsteady aerodynamics, in a form that is suitable for control design. Classical approaches to unsteady aerodynamics include the frequency-domain models of Theodorsen [58] and the time-domain models of [63], but these are not directly suitable for control design. A handful of more modern studies are available for rolling delta wings [4, 44] and for airfoils undergoing dynamic stall [18, 41], but these still have many limitations. For instance, the model in [18] must be calibrated to experimental data, and often does not match data it was not specifically calibrated against [9]. The model [41] is linear, and any nonlinear effects such as bifurcations and hysteresis are not captured. The focus of the present effort is to obtain models that overcome these various shortcomings, and may be used for designing flight controllers.

## 1.1 Techniques

The overall goal of this work is to provide improved models for design of flight controllers. Our focus, however, is on the aerodynamic models: predicting lift, drag, and moments from quantities such as freestream velocity and angle of attack. In traditional flight dynamic models, these lift and drag forces are typically treated as quasi-steady: for instance, lift is treated as a static function of angle of attack (equal to the steady-state value). When unsteady aerodynamics become important, this approach is not sufficient, and so we treat the aerodynamic models separately, as shown in Figure 1. The focus of the present effort is to determine models for the "Aerodynamics" block in Figure 1.

The models we are developing in this project build on a number of numerical tools, which we describe briefly here.

**Immersed boundary solver.** The starting point for many of the modeling approaches we employ is a high-fidelity direct numerical simulation (DNS). (In many cases, experimental data is equally suitable.) We use an Immersed Boundary Fractional Step method with a fast solver, developed by Colonius and Taira [13]. The studies here are for low Reynolds number (mostly at Re = 100), but the modeling techniques are equally applicable at higher Reynolds numbers.

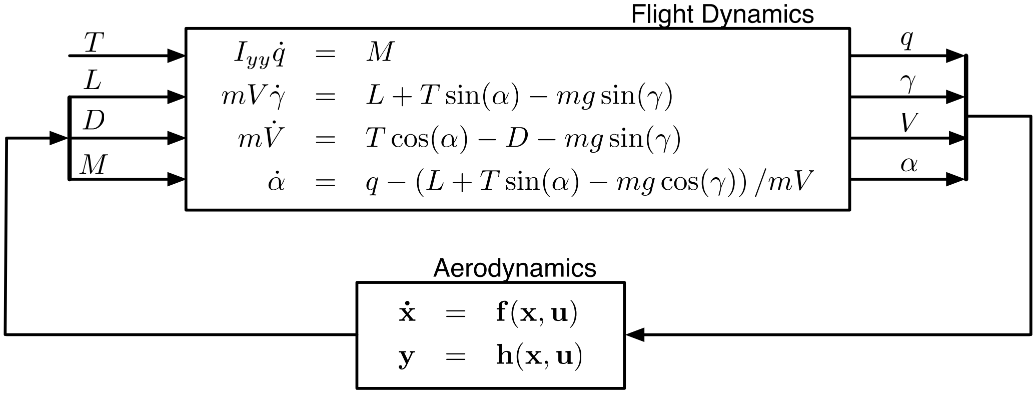


Figure 1: Overall structure for coupling of flight dynamics to unsteady aerodynamic models.

**Lagrangian Coherent Structures.** In order to identify the coherent structures that arise in these unsteady flows, we compute Lagrangian Coherent Structures [24], which precisely determine the boundaries of separation bubbles and other structures one needs to model in order to accurately describe the unsteady aerodynamics. These methods originate from the dynamical systems community, employing ideas of hyperbolic invariant manifolds and Finite-Time Lyapunov Exponents (FTLE). These techniques give a clear picture of the structure of these complex unsteady flows, and thus allow one to better understand the flow physics.

**Balanced Proper Orthogonal Decomposition.** In a related MURI project on Closed-Loop Flow Control, we have developed improved model reduction methods, based on approximations of balanced truncation that are tractable even for very large systems. This method, called Balanced Proper Orthogonal Decomposition (BPOD) has been quite successful for producing low-order models suitable for control design [2], and significantly outperforms standard POD-Galerkin methods. In the present effort, we employ this technique, and develop it further, introducing a method that does not require adjoint simulations, and extending it to systems with periodic orbits (such as periodic vortex shedding).

## 1.2 Summary of this report

This report describes the results of this three-year project, and the technical results are divided into five sections. In Section 2, we provide some necessary background on classical models for unsteady aerodynamics.

In Section 3, we describe how we use Finite-Time Lyapunov Exponents (FTLE) to identify when unsteady separation occurs, and use these quantities to gain an improved understanding of when the classical models break down. We demonstrate that the classical models fail when there is significant leading-edge separation, which occurs at large Strouhal numbers or reduced frequencies. A key result of this section is a method for fast computation of FTLE fields, when one needs to compute many fields at nearby times (as when making a movie).

Section 4 presents two different modeling procedures for predicting unsteady aerodynamics using nonlinear models. One approach is based on Proper Orthogonal Decomposition (POD) and Galerkin projection, and the other is a simple “phenomenological” model based on the phenomena observed in the real flow. The models include nonlinear behavior such as Hopf bifurcations and limit cycles, both of which are observed in the real flow. However, both modeling strategies have a serious limitation in that they need to be calibrated for a particular angle of attack, and are not immediately suitable for control design. Nevertheless, they provide a starting point for future work in developing nonlinear models for these unsteady flows.

Section 5 presents the technique of model reduction that has been most successful in this work: Balanced Proper Orthogonal Decomposition (Balanced POD). This technique is based on *linear* models, and typically requires numerical simulations of both the linearized equations and the corresponding *adjoint* equations. The need for adjoint information is a severe limitation of this method, as it prohibits it from being applied to experimental data. A major result of this project is a method for developing reduced-order models that are identical to Balanced POD, without the need for adjoint information. This procedure is presented in Section 5.1. Another extension is to flows linearized about a periodic orbit, such as periodic vortex shedding, and this is described in Section 5.2.

The model reduction methods developed in Section 5 are then applied to unsteady aerodynamics in Section 6, and we view the models in this section as the models most appropriate for design of flight controllers. These models are based on Wagner's indicial response [63], and agree with Wagner's models to an arbitrarily high degree of accuracy. While Wagner's models represent the lift as a convolution integral that is cumbersome to compute, and not suitable for control synthesis, our procedure produces state-space models that may be used directly for control design. Furthermore, our models are formulated in a way that directly builds upon standard approaches incorporating classical "stability derivatives" $C_{L\alpha}$, $C_{L\dot{\alpha}}$, $C_{L\ddot{\alpha}}$ (as in [57]), as shown in Figure 28.

The models obtained in Section 6 predict the unsteady response very well for small-amplitude maneuvers. For large amplitude maneuvers, agreement is poor when the angle of attack becomes larger than about 20°. More work is needed, however, to extend these models to the nonlinear regime.

## 2 Background: classical models

When modeling the aerodynamic forces acting on an airfoil in motion, it is natural to start with a quasi-steady approximation. Instead of dealing with the full unsteady problem, one assumes that the airfoil's center of mass, $h$, and angle of attack, $\alpha$, motions are "gradual" enough for the flow field to locally equilibrate to the motion. In this way, the unsteady terms in the flow equations are set to zero and the motion is accounted for by translating $\dot{h}$ into an effective angle of attack and $\dot{\alpha}$ into an effective camber. Finally, applying the assumption of a thin airfoil, we obtain a quasi-steady estimate for the lift coefficient

$$C_L = 2\pi\left(\alpha + \dot{h} + \frac{1}{2}\dot{\alpha}\left(\frac{1}{2} - a\right)\right) \tag{1}$$

Lengths are nondimensionalized by $2b$ and time is nondimensionalized by $2b/U_\infty$, where $U_\infty$ is the free stream velocity, $b$ is the half-chord length and $a$ is the pitch axis location with respect to the 1/2-chord (e.g., pitching about the leading edge corresponds to $a = -1$, whereas the trailing edge is $a = 1$).

### 2.1 Theodorsen's frequency response

In 1935, Theodorsen[58] went beyond the quasi-steady models and solved for the lift distribution around an idealized airfoil in purely sinusoidal pitch and plunge maneuvers. His theory is soluble with analytic techniques, relying on a number of simplifications, such as an inviscid, incompressible flow field and infinitesimal deflections of a flat plate, leaving an idealized planar wake. Because his theory was developed to handle purely sinusoidal maneuvers, it is represented in the frequency domain. Theodorsen's model predicts the unsteady lift as

$$C_L = \underbrace{\frac{\pi}{2}\left[\ddot{h} + \dot{\alpha} - \frac{a}{2}\ddot{\alpha}\right]}_{\text{Apparent Mass}} + \underbrace{2\pi\left[\alpha + \dot{h} + \frac{1}{2}\dot{\alpha}\left(\frac{1}{2} - a\right)\right]}_{\text{Circulatory}} C(k)$$

where Theodorsen's function $C(k)$ is a transfer function relating sinusoidal inputs of reduced frequency[30] $k$ to their aerodynamic response. The first set of terms represent the "apparent mass", or non-circulatory terms. The second set of terms are due to circulatory effects, and are exactly the quasi-steady forces multiplied by Theodorsen's function, which accounts for the change in magnitude and phase of these terms with changes in reduced frequency. This expression simplifies considerably for an airfoil in pure pitch or pure plunge

$$\text{pure plunge } (\alpha = 0) \qquad C_L = \frac{\pi}{2}\ddot{h} + 2\pi\dot{h}C(k) \tag{2}$$

$$\text{pure pitch} \qquad C_L = \frac{\pi}{2}\left[\dot{\alpha} - \frac{a}{2}\ddot{\alpha}\right] + 2\pi\left[\alpha + \frac{1}{2}\dot{\alpha}\left(\frac{1}{2} - a\right)\right] C(k) \tag{3}$$

## 2.2 Wagner's indicial response

Although Theodorsen's model is a powerful tool for determining unsteady lift coefficients, it is only solvable in closed form for sinusoidal forcing. The time domain method of Wagner[63] makes it possible to reconstruct the lift response to arbitrary angle-of-attack input, $\alpha(t)$, by superposition of the "indicial" lift response $C_L^S(t)$ due to a step response in angle of attack, $\dot{\alpha} = \delta(t)$:

$$C_L(t) = C_L^S(t)\alpha(0) + \int_0^t C_L^S(t-\tau)\dot{\alpha}(\tau)d\tau \tag{4}$$

Wagner originally derived the indicial response analytically, accounting for added mass and shed-wake effects in a manor similar to that of Theodorsen. However, it is possible to reduce the number of simplifying assumptions by obtaining the indicial response $C_L^S$ from experiment. Therefore, the only assumption is that of linearity; a more general approach based on functionals has been developed to extend this theory for nonlinear indicial response[59].

There are a number of interesting generalizations to these methods, such as the methods of Sears[62] and Küssner[31], which extended the methods of Theodorsen and Wagner, respectively, to the problem of moving wind direction, rather than moving airfoil. However, this section is meant to provide a brief review of the classical tools developed for modeling unsteady aerodynamic forces in the 1920s–1950s. A more complete treatment of the subject can be found in Leishman[33].

Wagner's indicial response benefits from its time domain formulation; however, the superposition integral approach is computationally expensive and does not fit nicely into a control framework. In Section 6, we present a systematic approach for obtaining computationally tractable reduced order models (ROMs) based on the indicial response. Moreover, these models take the form of low-dimensional, state-space equations, which are ideal for the application of analysis and control techniques.

# 3 Finite-time Lyapunov exponents for detecting breakdown of classical models

Classical models for unsteady aerodynamics often assume that the flow remains attached over a wing. Violation of this assumption is a common reason for classical models to break down, and therefore detecting unsteady separation is valuable. Moreover, an understanding of flow separation provides insight into the flow physics involved in these phenomena.

Separation in unsteady flows is a surprisingly subtle phenomenon. In steady flows, the criterion for separation is simple: at a separation point, the shear stress vanishes at the walls. In an unsteady flow, this criterion is not valid, either when applied to the time-averaged mean flow, or to the instantaneous unsteady flow, and more sophisticated techniques are required [64].

Finite-time Lyapunov exponents (FTLE) are a valuable tool for studying unsteady separation, and identifying separated regions. However, FTLE fields are reasonably expensive to compute, especially when one is interested in an unsteady phenomenon, and visualizing FTLE fields at many different times. In Section 3.1 we present a technique for significantly improving the speed of this computation. In Section 3.2 we use this technique to identify when classical unsteady models fail, and we demonstrate that this breakdown typically occurs when separation becomes significant. For further details on these methods, see [7, 8].

## 3.1 Fast computation of time-varying FTLE fields

The theory and computation of finite-time Lyapunov exponents (FTLE), also known as direct Lyapunov exponents (DLE), is a relatively modern development [23, 53], with extensions to 3-dimensional [20, 22] and $n$-dimensional [36] flows. FTLE analysis has been widely applied in a number of branches of fluid mechanics, including fluid transport [51, 15, 52], biopropulsion [48, 65, 19], flow over airfoils [38, 9, 7], plasmas [47], and geophysical flows [34, 35].

Because FTLE analysis is particularly useful for unsteady flows, it is often necessary to compute a sequence of FTLE fields in time to visualize an unsteady event. As flows become more complex, computations become increasingly expensive. In particular, FTLE calculations are expensive because a large number of particle trajectories must be integrated in order to obtain a particle flow map, often from stored velocity fields. When computing a sequence of FTLE fields in time, it is possible to speed up the computation considerably by eliminating redundant particle integrations. One approach that has been developed uses adaptive mesh refinement to reduce the number of integrations [16, 50, 54].

The approach here is to construct an approximate flow map by composing intermediate flow maps from FTLE field calculations at neighboring times. The first class of flow map approximation, denoted unidirectional composition, constructs a flow map by composing intermediate flow maps which are all aligned in the same time direction. The second class, denoted bidirectional composition, composes intermediate flow maps in both positive and negative-time. The methods are compared using analytic estimates for accumulated error and computation time as well as benchmarks on a number of example flows.

**Main results** In this section we demonstrate that the unidirectional method is both fast and accurate, although it requires significantly more memory than the bidirectional method. Orders of magnitude speed-up may be achieved over the standard method, and computational improvement scales with the desired time resolution of the FTLE animation. The bidirectional method suffers from significant error which is aligned with the opposite-time

Lagrangian coherent structures. To understand this coherent error, we provide an error analysis for both methods, and uncover an important relationship between positive-time LCS (pLCS) and negative-time LCS (nLCS). In particular, in the neighborhood of a time-dependent saddle, particles near the pLCS flow into particles near the nLCS in positive time.

#### 3.1.1 *Standard computation of FTLE*

Consider a time-dependent velocity field $\mathbf{u}$ on $\mathbb{R}^n$ and a particle trajectory $\mathbf{x}(t)$ which satisfies

$$\dot{\mathbf{x}} = \mathbf{u}\left(\mathbf{x}(t), t\right). \tag{5}$$

The velocity field, $\mathbf{u}$, may be an unsteady solution of the Navier-Stokes equation, although it is only assumed that $\mathbf{u}$ is at least $C^0$ in time and $C^1$ in space. However, to extract Lagrangian coherent structures from the Hessian of the FTLE field, $\mathbf{u}$ must be $C^2$ in space [53]. The velocity field may be analytically defined, but is more often obtained from experiments or direct numerical simulation which produce velocity field data at discrete snapshots over a finite range of time. A method of computing finite-time Lyapunov exponents (FTLE) on a finite amount of discrete velocity field data has been developed [23, 53].

Computing an FTLE field typically involves four steps. First, a grid of particles $X_0 \subset \mathbb{R}^n$ is initialized over the domain of interest. The particles are advected (i.e., integrated) with the flow from initial time 0 to final time $T$, resulting in a time-$T$ particle flow map, $\Phi_0^T$, defined as:

$$\Phi_0^T : \mathbb{R}^n \to \mathbb{R}^n; \quad \mathbf{x}(0) \mapsto \mathbf{x}(0) + \int_0^T \mathbf{u}(\mathbf{x}(\tau), \tau) d\tau. \tag{6}$$

Next, the flow map Jacobian, $\mathbf{D}\Phi_0^T$ is computed, usually by finite-differencing, to obtain the Cauchy-Green deformation tensor,

$$\Delta = \left(\mathbf{D}\Phi_0^T\right)^* \mathbf{D}\Phi_0^T \tag{7}$$

where $^*$ denotes transpose. Finally, the largest eigenvalue, $\lambda_{\max}$, of this symmetric tensor is extracted and synthesized into an FTLE field:

$$\sigma(\Phi_0^T; \mathbf{x}) = \frac{1}{|T|} \log \sqrt{\lambda_{\max}(\Delta(\mathbf{x}))}. \tag{8}$$

The bottleneck in this procedure is the large number of particle integrations required to obtain the particle flow map, $\Phi_0^T$. Moreover, if the velocity field is time-varying, it is necessary to compute a sequence of FTLE fields in time to visualize unsteady events, as shown schematically in Fig. 2.

#### 3.1.2 *Flow Map Approximation*

As seen in Fig. 2, the standard method of computing a sequence of FTLE fields involves inefficient re-integration of particles. The unidirectional and bidirectional methods outlined below streamline the computation of neighboring FTLE fields by approximating the time-$T$ flow map, $\Phi_{t_0}^{t_0+T}$, which can be written as:

$$\Phi_{t_0}^{t_0+T} \quad = \quad \Phi_{t_{N-1}}^{t_N} \circ \cdots \circ \Phi_{t_1}^{t_2} \circ \Phi_{t_0}^{t_1} \tag{9}$$

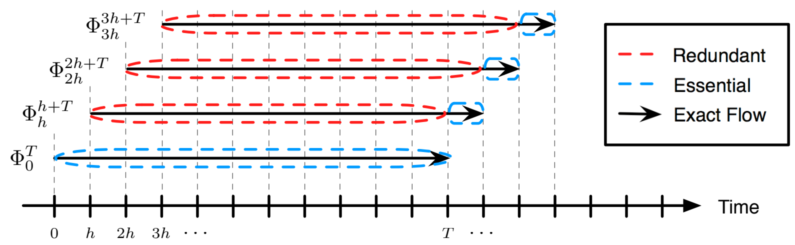


Figure 2: The standard method for computing FTLE. Exact flow maps $\Phi_{kh}^{kh+T}$ for $k \in \{0,1,2,3\}$ are shown (solid black arrow). Essential (blue) and redundant (red) particle integrations are outlined in dashed ovals.

where $t_N = t_0 + T$.

Because the flow maps are obtained numerically on a discrete grid of points, $X_0 \subset \mathbb{R}^n$, it is necessary to interpolate the map at points $\mathbf{x} \notin X_0$. Consider a flow map $\Phi : \mathbb{R}^n \to \mathbb{R}^n$, defined on $\mathbb{R}^n$, and the same flow map restricted to $X_0$, $\Phi|_{X_0} : X_0 \to \mathbb{R}^n$. The interpolation operator $\mathcal{I}$ takes the discrete map $\Phi|_{X_0}$ and returns the map, $\Phi$, defined on all of $\mathbb{R}^n$:

$$\mathcal{I} : \Phi|_{X_0} \mapsto \Phi \tag{10}$$

Using the shorthand $\mathcal{I}\Phi \triangleq \mathcal{I}\left(\Phi|_{X_0}\right)$, we obtain the approximate flow map:

$$\begin{aligned}\tilde{\Phi}_{t_0}^{t_0+T}(X_0) &= \mathcal{I}\Phi_{t_{N-1}}^{t_N} \circ \cdots \circ \mathcal{I}\Phi_{t_1}^{t_2} \circ \Phi_{t_0}^{t_1}(X_0) \\ &\approx \Phi_{t_0}^{t_0+T}(X_0)\end{aligned} \tag{11}$$

The bidirectional method approximates the time-$T$ flow map $\Phi_{t_0}^{t_0+T}$ by first integrating backward to a reference time, $t = 0$, then interpolating forward through a previously computed time-$T$ map, $\Phi_0^T$, and finally integrating forward to time $t_0 + T$. The unidirectional method approximates the time-$T$ flow map using a number of smaller time flow maps, $\Phi_{t_i}^{t_i+h}$, which all have the same time direction. Additionally, the chain rule may be applied to each of the methods, resulting in an approximation to the flow map Jacobian, $\mathbf{D}\Phi_{t_0}^{t_0+T}$.

### *3.1.3 Bidirectional Composition*

Bidirectional approximation eliminates redundancy from neighboring FTLE field computations by using the information from a known flow map at a given time, $\Phi_0^T$, to calculate an approximation to the flow map at future times, $\Phi_{t_0}^{t_0+T}$. First, $X_0$ is integrated backward from $t_0$ to the reference time 0. The distorted grid $\Phi_{t_0}^0(X_0)$ is then flowed forward through the interpolated map, $\mathcal{I}\Phi_0^T$, and finally integrated forward an amount $t_0$ to the desired time $t_0 + T$, as in Fig. 3:

$$\Phi_{t_0}^{t_0+T} \quad = \quad \Phi_T^{t_0+T} \circ \mathcal{I}\Phi_0^T \circ \Phi_0^{-t_0}. \tag{12}$$

The flow $\Phi_0^T$ is stored as a reference solution to compute an approximation to the flow map at later times $\tilde{\Phi}_{kh}^{kh+T} \approx \Phi_{kh}^{kh+T}$ by

$$\tilde{\Phi}_{kh}^{kh+T} \quad = \Phi_T^{kh+T} \circ \mathcal{I}\Phi_0^T \circ \Phi_{kh}^0 \qquad k \in \mathbb{Z} \tag{13}$$

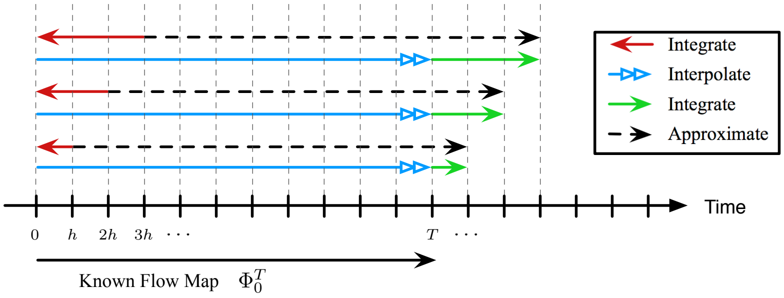


Figure 3: Schematic for bidirectional method (a). Given a known flow map $\Phi_0^T$ (solid black arrow), it is possible to approximate the flow map at later times $\tilde{\Phi}_{kh}^{kh+T}$ (dashed black arrow) by integrating backward in time to $t = 0$ (red arrow), flowing forward through the interpolated map $\mathcal{I}\Phi_0^T$ which was already computed (blue double arrow), and integrating trajectories forward to the correct final time (green arrow).

Instead of using $\Phi_0^T$ as the reference solution for every future time, it is convenient to use the new approximate flow map $\tilde{\Phi}_h^{h+T}$ as the reference solution for the next iteration:

$$\tilde{\Phi}_{2h}^{2h+T} \quad = \Phi_{h+T}^{2h+T} \circ \mathcal{I}\tilde{\Phi}_h^{h+T} \circ \Phi_{2h}^h \tag{14}$$

This method may be continued, using $\tilde{\Phi}_{kh}^{kh+T}$ to approximate $\tilde{\Phi}_{(k+1)h}^{(k+1)h+T}$:

$$\tilde{\Phi}_{(k+1)h}^{(k+1)h+T} \quad = \Phi_{kh+T}^{(k+1)h+T} \circ \mathcal{I}\tilde{\Phi}_{kh}^{kh+T} \circ \Phi_{(k+1)h}^{kh}. \tag{15}$$

Errors will compound more quickly since we are using approximate flow maps as the reference solutions for later approximations, as seen in Fig. 4.

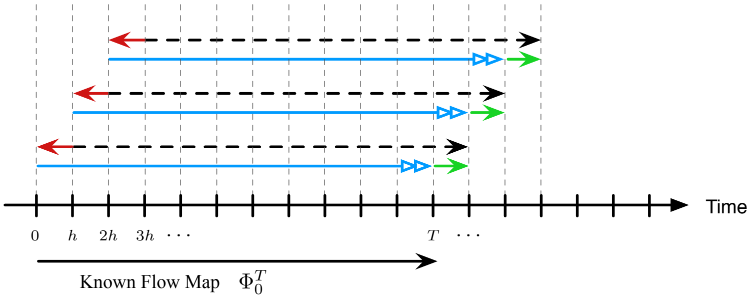


Figure 4: Schematic for bidirectional method (b). As in Fig. 3, a known flow map (solid black arrow) is used to approximate the flow map at a later time $\tilde{\Phi}_{kh}^{kh+T}$ (dashed black arrow). The approximate flow map is used as the known map for the next step (dashed black arrow).

### *3.1.4 Unidirectional Composition*

The basis of the unidirectional method is to eliminate redundant particle integrations by only integrating particle trajectories through a given velocity field a single time. If a sequence of FTLE snapshots is desired at a time spacing of $h$, for example as frames in an animation, then it is convenient to break up the time-$T$ flow map into smaller time-$h$ flow maps, where $T = kh$:

$$\tilde{\Phi}_0^{kh} \quad = \quad \mathcal{I}\Phi_{(k-1)h}^{kh} \circ \cdots \circ \mathcal{I}\Phi_h^{2h} \circ \Phi_0^h \tag{16}$$

This method is called unidirectional because particle flow maps of the same time direction are used, as opposed to the bidirectional method which composes both positive-time and negative-time flow maps.

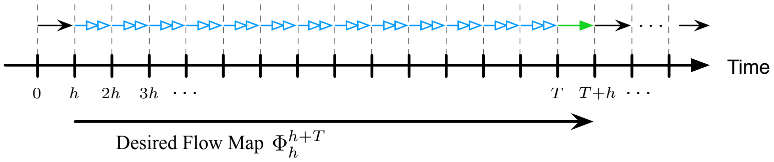


Figure 5: Schematic for unidirectional method. Time-$h$ flow maps (short blue arrows) are stored and composed to approximate the time-$T$ flow map (long black arrow). The next flow map only requires integrating one new time-$h$ flow map (green arrow).

The simplest approach is to compute a number of time-$h$ flow maps and store them in memory. Then, to construct an approximate $\Phi_{t_0}^{t_0+T}$, it remains only to compose the sequence of interpolated time-$h$ flow maps. The next iteration involves integrating one more time-$h$ flow map and composing the next sequence, as in Fig. 5.

To further improve efficiency by reducing the total number of flow map compositions, it is possible to construct a multi-tiered hierarchy of flow maps for reuse in neighboring flow map constructions. Given enough memory, it is possible to reduce the number of interpolated compositions by increasing the number of tiers of flow maps, each tier being constructed as the composition of two of the flow maps in the next tier lower, as in Fig. 6.

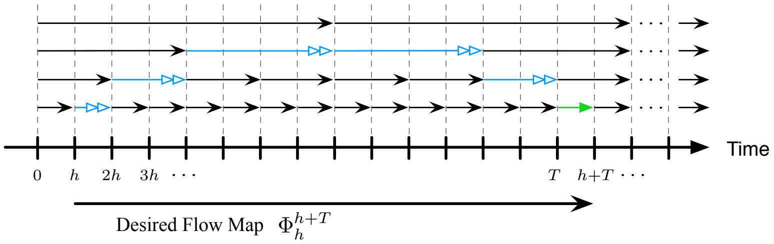


Figure 6: Schematic for unidirectional method with multiple tiers. The bottom tier of time-$h$ flow maps is computed as in Fig. 5. Pairs are composed to form the next tier of time-$2h$ flow maps, and so on. This method requires more storage, but fewer total compositions when computing a series of FTLE fields for an animation.

### *3.1.5 Chain Rule of Compositions*

As seen in Eq. (7), once the flow map $\Phi_{t_0}^{t_0+T}$ is obtained, it is necessary to compute the flow map Jacobian in order to extract the FTLE. Applying the chain rule to Eq. (9), it is possible

| Problem | Resolution | $T/h$ | Frames | Method | Mem. (GB) | Speed-up | Accurate |
|---|---|---|---|---|---|---|---|
| Double Gyre | $1024 \times 512$ | 15 | 30 | Standard | .05 | 1 | Yes |
| | | | | Unidirectional | .36 | 10 | Yes |
| | | | | Bidirectional | .14 | 6.2 | No |
| Pitching plate | $1024 \times 512$ | 15 | 30 | Standard | .48 | 1 | Yes |
| | | | | Unidirectional | .70 | 8.2 | Yes |
| | | | | Bidirectional | .50 | 5.4 | No |
| Pitching plate | $600 \times 300$ | 150 | 192 | Standard | .48 | 1 | Yes |
| | | | | Unidirectional | 1.8 | 67 | Yes |
| | | | | Bidirectional | .48 | 54 | No |
| ABC flow | $128^3$ | 20 | 40 | Standard | .48 | 1 | Yes |
| | | | | Unidirectional | 2.6 | 6.8 | Yes |
| | | | | Bidirectional | .73 | 7.3 | No |

Table 1: Comparison of methods on various examples fluid flows. The unidirectional method both fast and accurate, but requires more memory than the other methods, providing one or two orders of magnitude computational improvement over the standard method.

to express the flow map Jacobian as a product of the Jacobians of intermediate flow maps:

$$\begin{aligned}\mathbf{D}(\Phi_{t_0}^{t_N})(\mathbf{x}) &= \mathbf{D}\left(\Phi_{t_{N-1}}^{t_N} \circ \cdots \circ \Phi_{t_1}^{t_2} \circ \Phi_{t_0}^{t_1}\right)(\mathbf{x}) \\ &= \mathbf{D}\Phi_{t_{N-1}}^{t_N}\left(\Phi_{t_0}^{t_{N-1}}(\mathbf{x})\right) \times \cdots \times \mathbf{D}\Phi_{t_0}^{t_1}(\mathbf{x})\end{aligned} \tag{17}$$

Applied to the bidirectional methods, this yields:

$$\begin{aligned} \Phi_h^{h+T} =& \Phi_T^{h+T} \circ \Phi_0^T \circ \Phi_h^0 \\ \implies \quad \mathbf{D}\Phi_h^{h+T}(\mathbf{x}) =& \mathbf{D}\Phi_T^{h+T}\left(\Phi_0^T \circ \Phi_h^0\right)(\mathbf{x}) \times \\ & \times \mathbf{D}\Phi_0^T\left(\Phi_h^0\right)(\mathbf{x}) \circ \mathbf{D}\Phi_h^0(\mathbf{x}), \end{aligned} \tag{18}$$

and applied to the unidirectional methods, this yields:

$$\begin{aligned} \Phi_0^T =& \Phi_{T-h}^T \circ \cdots \circ \Phi_h^{2h} \circ \Phi_0^h \\ \implies \quad \mathbf{D}\Phi_0^T(\mathbf{x}) =& \mathbf{D}\Phi_{T-h}^T\left(\Phi_0^{T-h}(\mathbf{x})\right) \times \cdots \\ & \cdots \times \mathbf{D}\Phi_h^{2h}\left(\Phi_0^h(\mathbf{x})\right) \times \mathbf{D}\Phi_0^h(\mathbf{x}). \end{aligned} \tag{19}$$

#### 3.1.6 Comparison of Methods

Each method from Section 3.1.2 is implemented and tested on three example problems: the periodic double gyre, 2D flow over a pitching flat plate at Reynolds number 100, and 3D unsteady ABC flow. These examples are chosen because they cover a range of features including 2D and 3D vector fields, which are either defined analytically or obtained from data files from DNS on either open, closed, or periodic domains. Each example problem is discussed more in Appendix B of [8], including details such as how the velocity field is defined, and on what domain. In the pitching plate example, velocity field snapshots are all loaded up-front before applying the methods.

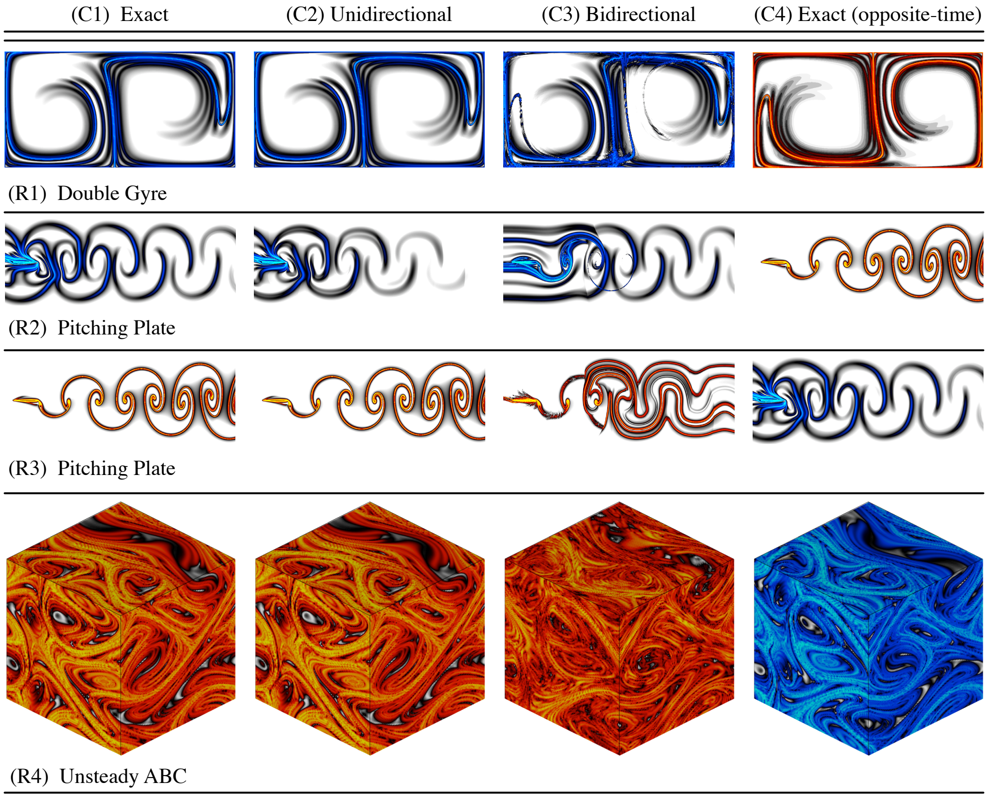


Figure 7: Graphical comparison of each method on four examples: (top row) positive-time FTLE of double gyre, (second row) positive-time FTLE of 2D pitching plate, (third row) negative-time FTLE of 2D pitching plate, (bottom row) negative-time FTLE of 3D ABC flow. Each figure shows the FTLE field after a number of iterations of the given method. The column of FTLE fields calculated using unidirectional composition agree well with the exact FTLE fields computed using the standard method. The column of FTLE fields calculated using bidirectional composition all have significant error which is aligned with the opposite-time coherent structures. The opposite-time FTLE fields are shown in the rightmost column for comparison with the bidirectional method. FTLE fields computed for positive-time flow maps are blue and those computed for negative-time flow maps are red.

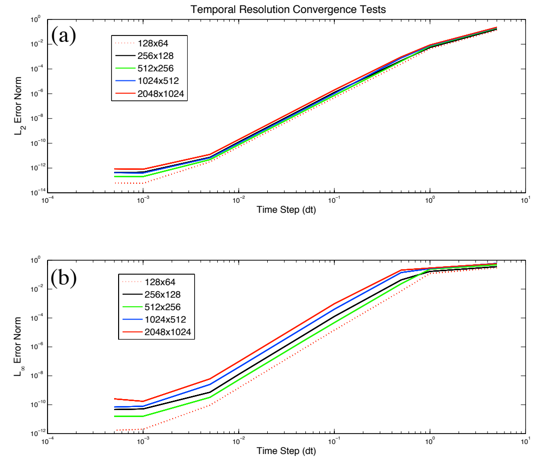


Figure 8: Convergence tests for FTLE field error vs. integration time-step and grid spacing on double-gyre.

Table 1 summarizes the results comparing each method on the three example fluid flows. In each comparison, the standard, unidirectional and bidirectional methods are used to compute a sequence of FTLE fields which are frames in an unsteady animation. The flow map duration used to compute an FTLE field is $T$, and the time-spacing between neighboring FTLE fields is $h$, so the number of animation frames per flow map duration is $T/h$. As demonstrated in Section 3.1.8, this is an upper bound on the speed-up of the unidirectional method.

In each comparison, the unidirectional method is accurate and offers the greatest speed-up over the standard method. However, it also requires more memory than any other method. The bidirectional method is fast and uses less memory than the unidirectional method, but is prone to large errors in the approximate flow map and does not accurately reproduce the FTLE field.

Contour plots of the FTLE fields computed after a number of iterations of each method are shown in Fig. 7. The FTLE fields computed with the unidirectional method agree with those computed using the standard method, as seen by comparing the first and second column of Fig. 7. FTLE fields computed using the bidirectional method, shown in the third column, have large errors. It is interesting to note that these errors are aligned with coherent structures found in the opposite-time FTLE field, shown in the fourth column. An analysis of this coherent error is provided in Section V of [8].

### *3.1.7 Example - Double Gyre*

Figure 8 shows the $L_2$ and $L_\infty$ error of the forward-time FTLE field for the double gyre computed using the standard method with $T = 16$, as time-step $\Delta t$ and grid spacing $\Delta x$ are varied. At a given grid spacing, a reference FTLE field is computed using a sufficiently small time-step, $\Delta t = 10^{-4}$, so that the FTLE field may be considered exact. For small enough time-step $\Delta t \approx .001$, the FTLE field error converges. All integrations are performed using a fixed time-step, fourth order Runge-Kutta scheme.

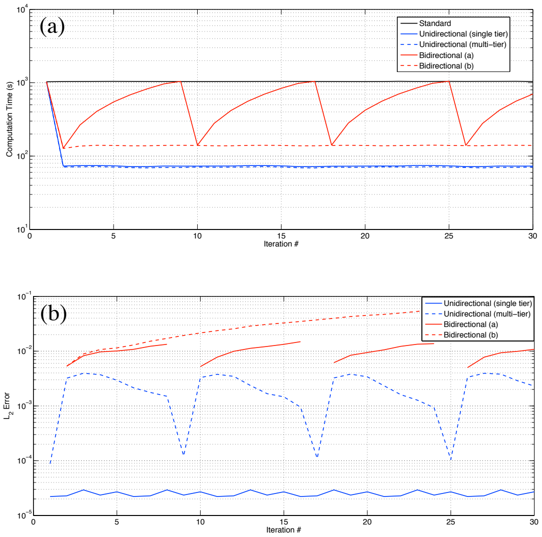


Figure 9: Computational time vs. Iteration (top) and $L_2$ error vs. Iteration (bottom) for FTLE fields of the double gyre with resolution $1024 \times 512$.

The flow map approximation methods are only faster than the standard method when used to compute a sequence of FTLE fields in time, as in the construction of frames for a movie. Figure 9 compares computation time and $L_2$ error vs. frame number (iteration #) for a sequence of FTLE fields of the double gyre, computed using the standard, unidirectional, and bidirectional methods. Each iteration produces an FTLE field which is a single frame in an animation of the unsteady FTLE field. In this example, the flow map duration is $T = 16$, the time spacing between each FTLE field is $h = 1$, and the time-step of integration is $\Delta t = .01$. The multi-tier unidirectional method uses four tiers.

The first FTLE field takes roughly the same time to compute using each of the methods. However, for subsequent iterations, the unidirectional and bidirectional methods are significantly faster. The computation time of bidirectional method (a) increases with the number of iterations, $k$, because integrating back from $t = kh$ to the reference time $t = 0$ becomes more costly as $k$ increases, as seen in Fig. 4. After $T/2h = 8$ iterations of bidirectional method (a), it is advantageous to compute a new reference flow map using the standard method. This explains the breaks in the solid red curve in part (b) of Fig. 9, as the bidirectional method is exact at these iterations. Bidirectional method (b) overcomes this increasing cost vs. iteration by using the flow map from the current iteration as the reference flow map at the next iteration. However, using an approximate flow map to compute the next approximation causes bidirectional method (b) to accumulate error more quickly than method (a). The unidirectional method is both the fastest and most accurate method in this comparison.

### *3.1.8 Computational Resources*

Again, consider a sequence of time-$T$ flow maps spaced $h$ apart, as might be required for an unsteady visualization. When there are many integration time-steps of size $\Delta t$ between each neighboring flow map, i.e. $\Delta t \ll h$, then the added cost of flow map composition becomes relatively small compared with the cost of integrating a time-$h$ flow map.

The standard method involves $(T/h) \times (h/\Delta t)$ integration steps for each new FTLE field, whereas the unidirectional method only requires $h/\Delta t$ integration steps, and the bidirectional method requires $2h/\Delta t$ integration steps. Assuming $\Delta t \ll h$, the speed-up of the unidirectional method over the standard method will increase as the number of frames in the animation per flow map duration, $T$. In other words, as $\Delta t/h \to 0$, the computation of $\Phi_{t_0}^{t_0+T}$ using the unidirectional method is $T/h$ times faster than using the standard method, and twice as fast as the bidirectional method.

In the examples above, all intermediate flow maps were stored in memory until no longer useful for future computations. Regardless of any parameters of the FTLE field animation, the standard and bidirectional methods must store a fixed number of flow maps. The standard method stores the single flow map $\Phi_{t_0}^{t_0+T}$, while the bidirectional method stores three maps: $\Phi_{t_0}^{0}$, $\Phi_{0}^{T}$, and $\tilde{\Phi}_{t_0}^{t_0+T}$. The unidirectional method, however, stores of every intermediate time-$h$ flow map $\Phi_{(k-1)h}^{kh}$, of which there are $T/h$. Therefore, the memory requirement of the unidirectional method scales linearly with the upper-bound on its speed-up, $T/h$.

The memory usage of the unidirectional method scales with the dimension of the flow $D$, the spatial resolution $R$, and the possible computational speed up of the method $S$, given by $T/h$:

$$\text{Memory (GB)} \sim S \times D \times R^D \tag{20}$$

$$= \frac{8\ \text{B/double}}{1024^3\ \text{B/GB}} \times \frac{T}{h} \times D \times R^D \tag{21}$$

For example, a series of two dimensional, high-definition (1920 × 1080 resolution) FTLE fields may be computed using the unidirectional method with up to 100× speed up using approximately 3.1 GB of RAM. A three dimensional FTLE field with resolution 512×256×64 may be computed with up to 100× speed up with approximately 19 GB of RAM.

In the pitching plate example, velocity fields are obtained from data files which are the output of a direct numerical simulation. Because loading velocity fields which are stored on disk is slow, it is important to minimize the number of file loads. In the pitching plate example, all of the velocity fields are loaded up-front and stored in memory throughout the computation. However, velocity fields are often too large to store them all in memory, for example in large 2D or 3D simulations. After the unidirectional method is initialized, subsequent iterations of the method only require loading velocity fields involved in the computation of a single, time-$h$ flow map. The standard method, however, must load velocity fields relevant to the entire time-$T$ flow map, requiring $T/h$ times as many file loads as the unidirectional method.

### 3.2 Breakdown of classical unsteady models

Here, we examine the effectiveness of the classical models described in Section 2, for an airfoil undergoing sinusoidal pitching or plunging. The simulations here are for a flat plate at low Reynolds number (Re = 100), and are calculated using the immersed boundary method of [13]. For details of the simulation, see [7].

For this low-Reynolds-number flat plate, the steady-state lift is shown as a function of angle

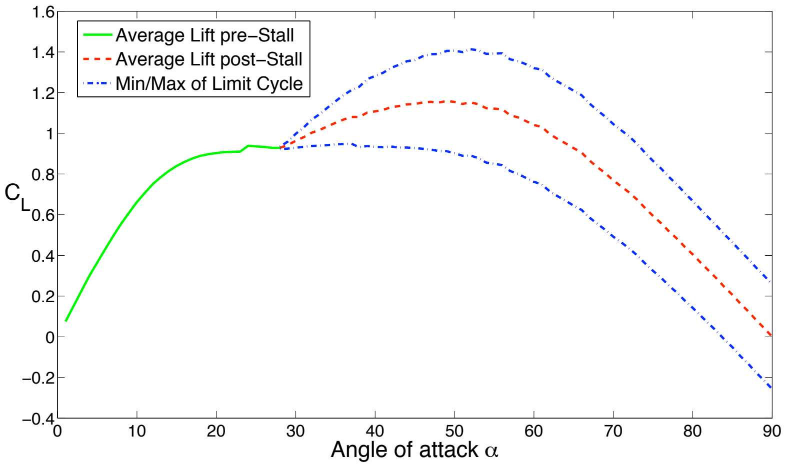


Figure 10: Lift coefficient vs. angle of attack for fixed plate at Re = 100. A supercritical Hopf bifurcation occurs at $\alpha \approx 28°$. The dashed line represents average post-stall lift and the dotted lines represent minimum and maximum post-stall lift.

of attack in Figure 10. At $\alpha_c \approx 28°$, a Hopf bifurcation occurs[2], and for $\alpha > \alpha_c$, the flow is unsteady, corresponding to periodic vortex shedding from the leading and trailing edges.

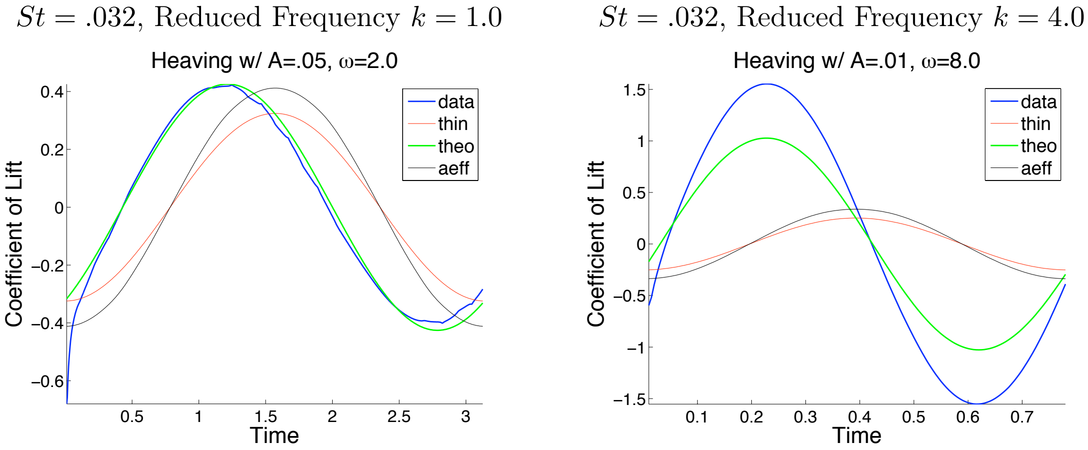
### *3.2.1 Theodorsen's model — sinusoidally plunging airfoil*

Using equations (1) and (2) it is possible to compare the thin airfoil theory and Theodorsen's model with the simulated response of a flat plate to sinusoidal plunging in Reynolds number 100 flow. It is also compared with an *effective* angle of attack approximation using a look up table for the lift coefficient at the static angle of attack $\alpha_{\text{eff}}(t) = \tan^{-1}(-\dot{h}(t)/U_\infty)$ for each point in the maneuver; this approximation may be classified as quasi-steady. The plunging motion is specified by the center of mass motion $h(t) = -A\sin(\omega t)$.

It has been shown[9] that for reasonable Strouhal numbers and reduced frequencies $k = \pi f c/U_\infty$ less than 0.5, the effective angle of attack approximation agrees well with DNS. However, for the same range of Strouhal numbers $\text{St} \in \{.032, .064, .128\}$, we find that Theodorsen's theory agrees with DNS up to reduced frequencies of 2.0, shown in Figure 11. This is consistent, since at larger reduced frequencies, quasi-steady assumptions break down and it becomes important to consider flow acceleration terms.

For larger Strouhal numbers, $\text{St} \in \{.256, .512\}$, Theodorsen's model disagrees with DNS even for small reduced frequencies, shown in Figure 12. This is particularly interesting, because these large Strouhal numbers correspond to maximum effective angles of attack which are larger than the critical stall angle shown in Figure 10. At these Strouhal numbers the effective angle of attack approximation plateaus due to $\alpha_{\text{eff}} > \alpha_c$. In addition to disagreeing with DNS in magnitude and phase, Theodorsen's model does not describe the higher-frequency components in the response.

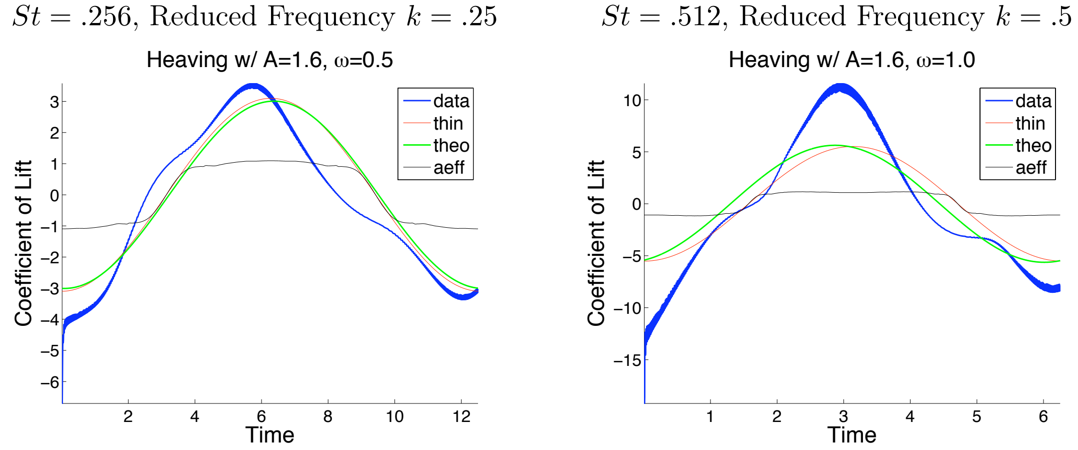
The table below shows the maximum effective angle of attack associated with each Strouhal number used above. Notice that for Strouhal numbers .256 and .512, the maximum effective



$St = .032$, Reduced Frequency $k = 1.0$

$St = .032$, Reduced Frequency $k = 4.0$

Figure 11: Each curve is a plot of lift coefficient vs. time for a sinusoidally plunging flat plate at Re = 100 and Strouhal number .032. The blue curve is the $C_L$ from DNS, the red curve is computed using thin airfoil theory, Eq. (1), the black curve uses an effective angle of attack approximation, and the green curve is Theodorsen's prediction, Eq. (2). Theodorsen's model agrees well with DNS for reduced frequencies $k < 2.0$ as long as the Strouhal number is small enough that the maximum effective angle of attack is less than the stall angle.



$St = .256$, Reduced Frequency $k = .25$

$St = .512$, Reduced Frequency $k = .5$

Figure 12: Each curve is a plot of lift coefficient vs. time for a sinusoidally plunging flat plate at Re = 100 and Strouhal numbers .256 and .512. The blue curve is the $C_L$ from DNS, the red curve is computed using thin airfoil theory, Eq. (1), the black curve uses an effective angle of attack approximation, and the green curve is Theodorsen's prediction, Eq. (2). For large Strouhal numbers, Theodorsen's model does not agree well with DNS even at low reduced frequency. Also, there are higher frequency components which are not captured by the models.

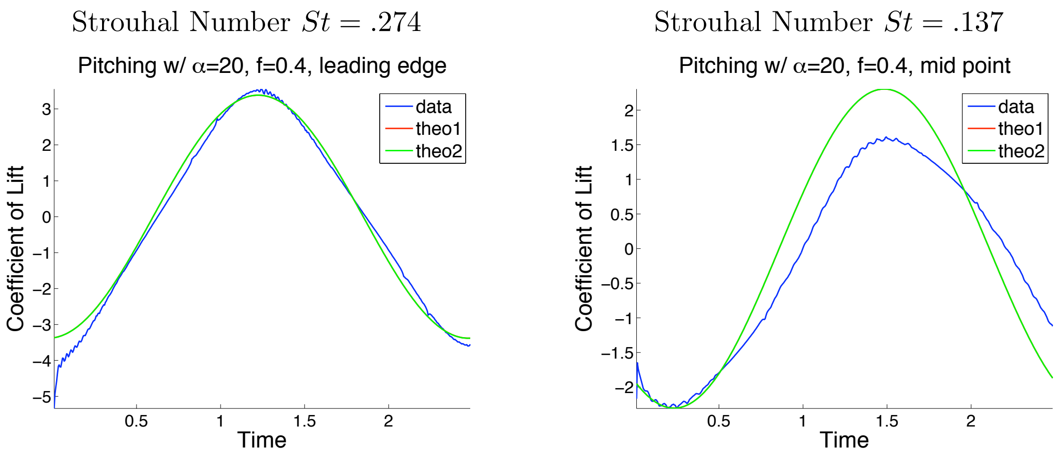


Figure 13: Each curve is a plot of lift coefficient vs. time for a sinusoidally pitching flat plate at Re = 100 and reduced frequency $k = 1.26$. The blue curve is the $C_L$ from DNS, and the green curve is Theodorsen's prediction, Eq. (3). Theodorsen's model matches DNS for a flat plate pitching to an amplitude of 20° about the leading edge and not the half-chord, even though the Strouhal number is larger for pitching about the leading edge.

angle of attack is larger than the critical stall angle $\alpha_c \approx 28°$.

| Strouhal number (St) | .032 | .064 | .128 | .256 | .512 |
|---|---|---|---|---|---|
| Max effective aoa ($\alpha_{\text{eff}}^{\max}$) | 5.74° | 11.37° | 21.91° | 38.81° | 58.13° |

### *3.2.2 Theodorsen's model — sinusoidally pitching airfoil*

Similar to the case of a sinusoidally plunging airfoil, with equation (3) it is possible to compare Theodorsen's model with DNS for a sinusoidally pitching flat plate. Three pitching amplitudes $\alpha \in \{20°, 27.1°, 43.2°\}$ are examined, and for each amplitude the pitching point is varied along the chord from the leading-edge to the trailing-edge at every quarter-chord in between. In each case, the reduced frequency is $k = 1.26$. Interestingly, the agreement between Theodorsen's model and DNS does not depend so much on reduced frequency and Strouhal number as in the case of sinusoidal plunging, but instead depends much more on raw pitching amplitudes and pitching point. In Figure 13 we see that for the same angle of attack excursion Theodorsen's model agrees better with DNS when the plate pitches about the leading edge, even though the Strouhal number is larger.

This is consistent with the fact that Theodorsen's theory depends on the assumption of attached flow over the wing surface, and pitching about the mid-chord promotes leading-edge separation much more than pitching about the leading edge. This effect is shown in Figure 14, where the effect of leading edge separation is exaggerated due to the large pitch amplitude $\alpha_{\max} = 27.1°$.

When the pitch amplitude is larger than the stall angle $\alpha_c \approx 28°$, Theodorsen's model does not agree with DNS even when the plate is pitched about the leading edge; see Figure 15. This is interesting because there is a similar observation in the case of sinusoidal plung-

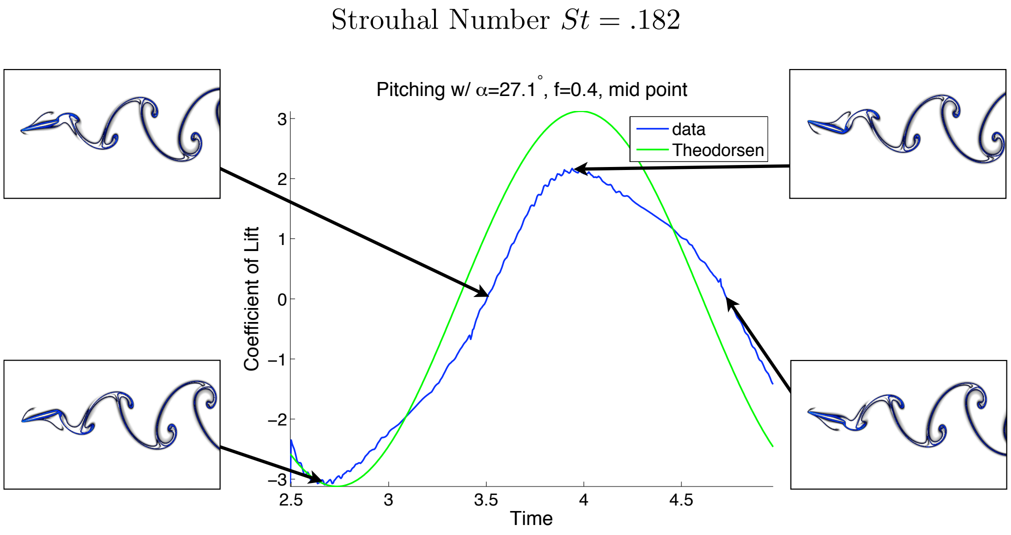

Strouhal Number $St = .182$

Figure 14: Leading edge separation is visualized using FTLE fields for a plate pitching to an amplitude of 27.1° about the mid chord.

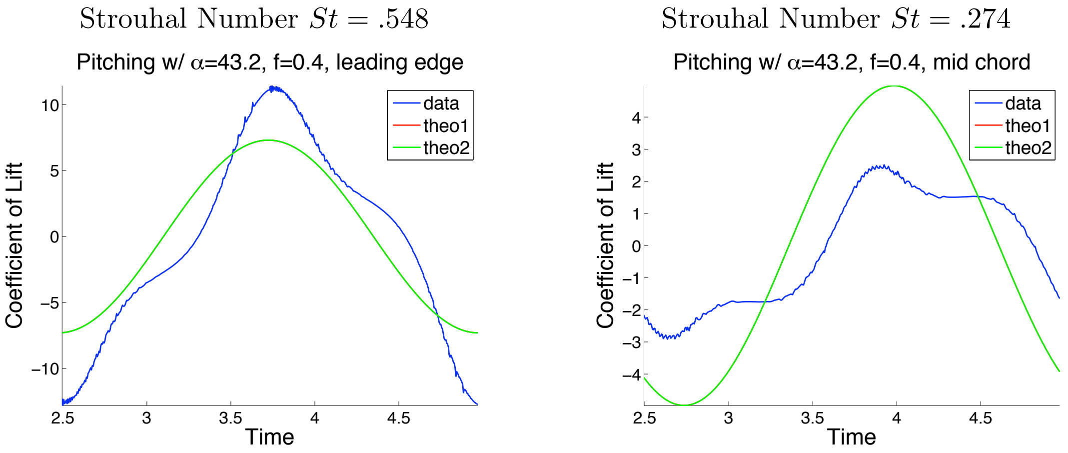
ing, where Theodorsen's model begins to disagree at Strouhal numbers which correspond to effective angles of attack larger than $\alpha_c$. This should not be surprising, however, since Theodorsen's theory is developed for infinitesimal oscillations where the flow is never separated and the wake is assumed to be planar. Under these assumptions, Theodorsen's model cannot possibly take high angle of attack vortex shedding effects into account.

In addition to the lack of agreement between Theodorsen's model and DNS for large pitching amplitudes, there are $C_L$ variations at twice the pitching frequency, as there was in the case of sinusoidal plunging for Strouhal numbers $\text{St} \in \{.256, .512\}$. Using FTLE to visualize the flow structures, it is possible to see not only leading edge separation, but also a distortion of the FTLE near the plate near the mid chord, as shown in Figure 16. It is likely that natural vortex shedding at this high angle of attack is interacting with the separation due to the airfoils pitching motion.

#### *3.2.3 Indicial response*

Compared with Theodorsen's method of predicting response forces, indicial response, equation (4) is an empirical method which relies on knowing only the response in lift to a small step in angle of attack. For the simulations below, the step in $\alpha$ was approximated by a steep sigmoidal step of 1°. The indicial response roughly predicts the initial peak observed in DNS for fast pitch-up maneuvers of moderate amplitude, $\alpha = 8°$ and $\alpha = 16°$, shown in Figure 17. However, superposition of a number of small steps fails to reproduce transient oscillations as the initial peak dies off.

For larger angle of attack pitch-up maneuvers, say $\alpha = 32°$, the indicial response predicts the rough form of transient lift, but doesn't capture the jagged peak which is observed in DNS. Because the method of indicial response involves staggered superposition of a number


Strouhal Number $St = .548$

Strouhal Number $St = .274$

Figure 15: Each curve is a plot of lift coefficient vs. time for a sinusoidally pitching flat plate at Re $= 100$ and reduced frequency $k = 1.26$. The blue curve is the $C_L$ from DNS, and the green curve is Theodorsen's prediction, Eq. (3). For pitching amplitude $\alpha_{\max} = 43.2° > \alpha_c$, Theodorsen's model doesn't agree with DNS, despite the pitching point.

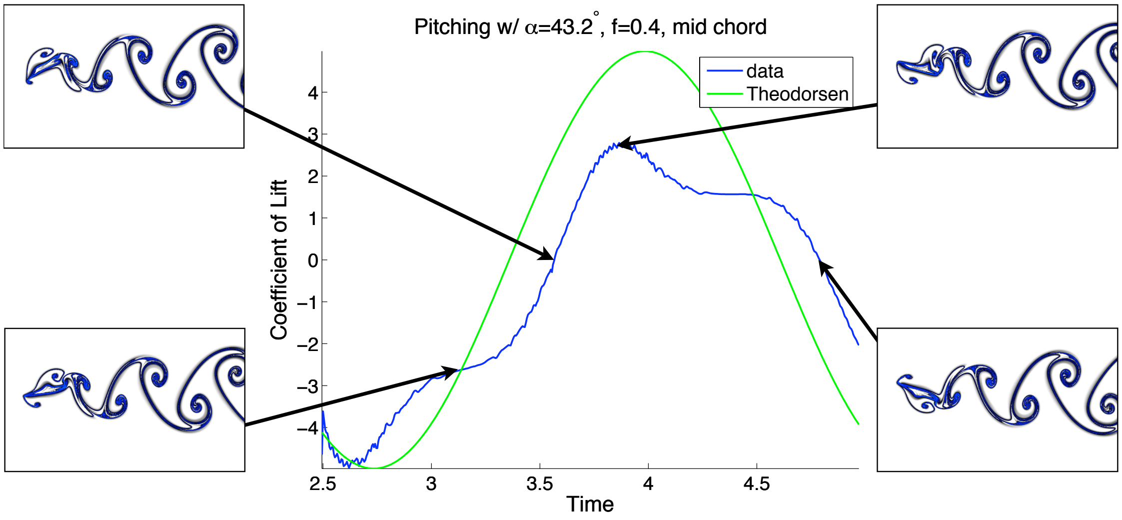


Figure 16: Leading edge separation is visualized using FTLE fields for a plate pitching to an amplitude of 43.2° about the mid chord.

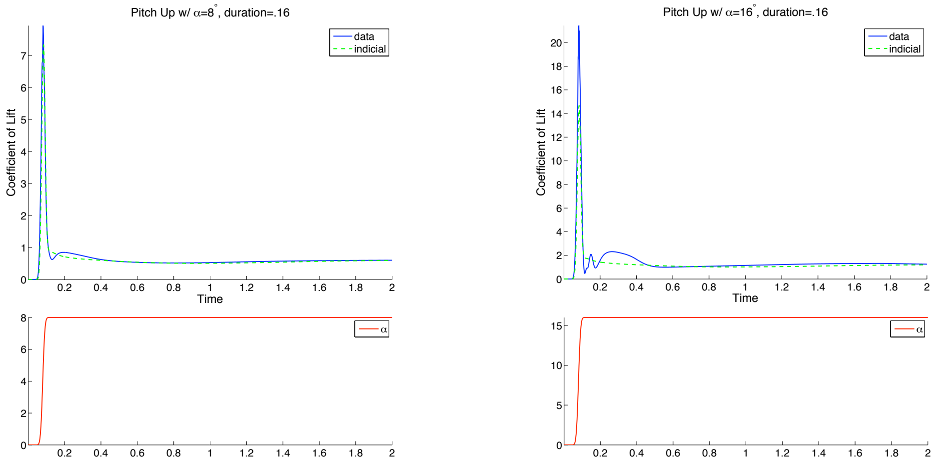


Figure 17: (top) Comparison of lift coefficient, $C_L$, between DNS and Indicial response for a flat plate in pitch-up maneuver at Re = 300. (bottom) Angle of attack vs. time. (left) Pitch-up of 8° with duration of 16 time steps. (right) Pitch-up of 16° with duration 16 time steps

of small steps to reconstruct a large step, it is not possible for this method to predict the periodic vortex shedding which is characteristic of high angles of attack. Therefore, even if it is able to predict transient lifts, it is not useful for steady state prediction at large angle of attack, as shown in Figure 18.

### *3.2.4 Conclusions*

In this section, we have investigated the unsteady aerodynamic forces on low-Reynolds number wings at high angle of attack and in pitch and plunge maneuvers using 2D direct numerical simulations. The classical theories of Theodorsen and Wagner have been compared with DNS for a number of pitch and plunge maneuvers of varying Strouhal number, reduced frequency, pitch amplitude and center. In addition to determining when these theories break down, the flow field is investigated using FTLE to visualize relevant flow structures to determine *how* the theories break down, indicating possible improvements to the models.

Comparison of Theodorsen's model for the lift of a sinusoidally plunging flat plate with forces from DNS showed agreement for moderate reduced frequencies $k < 2.0$ for a range of Strouhal numbers for which the maximum effective angle of attack is smaller than the critical stall angle. For the case of a sinusoidally pitching plate, agreement between Theodorsen's model and DNS was less dependent on Strouhal number than the position of the pitch axis along the chord. Pitching about the mid-chord, while resulting in a smaller Strouhal number than pitching about the leading edge, promotes leading edge separation and dynamic stall effects which are not captured by Theodorsen's model. However, Theodorsen's model does not agree with DNS for any pitch point if the angle of attack excursion is large enough to cause periodic vortex shedding. This is an important relationship between the theory for pitching and plunging airfoils; in particular, the theory breaks down in both cases when the angle of attack (resp. effective angle of attack) excursion exceeds the critical stall angle.

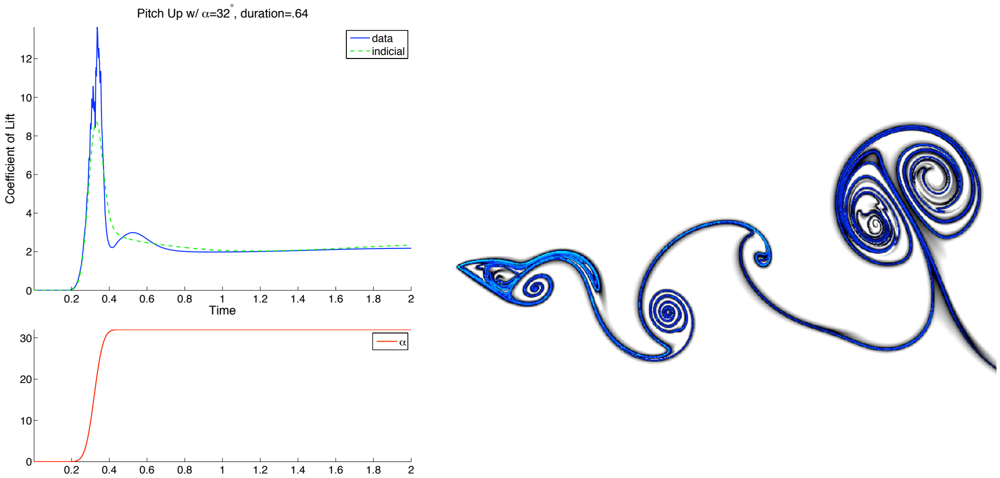


Figure 18: (left) Comparison of lift coefficient, $C_L$, between DNS and Indicial response for a flat plate in pitch-up at Re = 300 and sinusoidal pitch maneuver at Re = 100. (right) FTLE field visualization of the periodic laminar vortex shedding which takes place after the transients die down.

This observation is supported by the method of indicial response, where agreement between model and data begins to break down for large pitch-up maneuvers. The inability to capture unsteady effects due to high angle of attack is a fundamental limitation of both methods.

## 4 Nonlinear models for unsteady flows at fixed angle of attack

A first step towards developing models for unsteady aerodynamics is to develop models valid at a fixed angle of attack. In this section, we summarize our results using two approaches. In the first approach, we develop phenomenological models, which capture the essential features observed, namely a Hopf bifurcation at a critical angle of attack, as shown in Figure 10. In this model, the angle of attack $\alpha$ appears explicitly, so it is straightforward to see how it would generalize it to other angles of attack. However, in principle, such a generalization is difficult, as the model contains three empirical parameters which would need to be adjusted.

In the second approach, we develop more systematic models using Proper Orthogonal Decomposition (POD) and Galerkin projection. These models are very accurate for the conditions they are calibrated for, but they do not generalize well to other angles of attack, or more complex unsteady maneuvers. Thus, while the models in this section are interesting, we expect the models of Section 6 to be more useful for actual implementation.

### 4.1 Phenomenological models

Recall from Figure 10 that the flow over a wing at low Reynolds number exhibits a Hopf bifurcation at a critical angle of attack, at which the flow transitions from steady separation to unsteady vortex shedding. The simplest system of differential equations that describes this behavior is given by the normal form of a Hopf bifurcation [21]. In this section, we model the transient and steady-state lift associated with an impulsively started 2D plate using this simple model, along with a decoupled first-order lag:

$$\left.\begin{aligned}\dot{x} &= (\alpha-\alpha_c)\mu x - \omega y - ax(x^2+y^2)\\ \dot{y} &= (\alpha-\alpha_c)\mu y + \omega x - ay(x^2+y^2)\\ \dot{z} &= -\lambda z\end{aligned}\right\} \implies \begin{aligned}\dot{r} &= r\left[(\alpha-\alpha_c)\mu - ar^2\right]\\ \dot{\theta} &= \omega\\ \dot{z} &= -\lambda z\end{aligned} \tag{22}$$

The $z$ direction is decoupled and represents the exponential decay of transient lift generated from the impulsive start. Transforming the $(x, y)$ system into polar coordinates, it becomes clear that there is a fixed point at $r = 0$. This fixed point undergoes a subcritical Hopf bifurcation at $\alpha = \alpha_c$ resulting in an unstable fixed point at $r = 0$ and a stable limit cycle with radius $R = \sqrt{(\alpha-\alpha_c)\mu/a}$. The limit cycle represents the fluctuations in lift due to periodic vortex shedding of a plate at an angle of attack which is larger than the stall angle. Thus, at a particular angle of attack $\alpha$, the unsteady coefficient of lift $C_L$ is constructed from the average lift $\bar{C}_L$ and the state variables $y$ and $z$ as follows

$$C_L = \bar{C}_L + y + z$$

With knowledge of the actual lift vs. time from numerical experiment, it is possible to tune the parameters $(\mu, \lambda, \omega, a)$ with the rates of decay, period of shedding and amplitude of stable lift fluctuations. Initial conditions of the model are chosen to start the system with the right transient lift and phase. By properly tuning the constants with experimental data, this model will closely reproduce the transient lift dynamics of a stationary plate for a wide range of angles of attack $\alpha$. Figure 19 shows a typical example, for $\alpha = 35°$.

It is important to note that the model (22) is specifically chosen to exhibit a supercritical Hopf bifurcation as the angle of attack $\alpha$ varies through $\alpha_c$. Such a bifurcation is observed in simulations and experiments (Figure 10), so for a *fixed* set of parameters, the model (22) may be tuned to match the observed behavior over a range of angle of attack $\alpha$. Theoretically, one would expect such a model to be valid only for values of $\alpha$ close to the bifurcation value $\alpha_c$, but in practice we have observed that, even for fixed parameters, the model (22) may be tuned to match simulations reasonably closely over a wide range of parameter values, from $\alpha = 0$ to at least 35°.

It also is possible to model the transient lift from the impulsive start by coupling $z$ and $(x, y)$:

$$\left.\begin{aligned}\dot{x} &= (\alpha-\alpha_c)\mu x - \omega y - axz\\ \dot{y} &= (\alpha-\alpha_c)\mu y + \omega x - ayz\\ \epsilon\dot{z} &= -z + (x^2+y^2)\end{aligned}\right\} \implies \begin{aligned}\dot{r} &= r\left[(\alpha-\alpha_c)\mu - az\right]\\ \dot{\theta} &= \omega\\ \epsilon\dot{z} &= -z + r^2\end{aligned} \tag{23}$$

For $\epsilon \ll 1$, trajectories quickly settle to the "slow" manifold $z = x^2 + y^2$, which reduces to our original dynamics. This method has been useful in characterizing transients in the wake of a cylinder[45]. Because the flat plate at high angles of attack is a bluff body, its wake topology should be structurally equivalent to the wake behind a cylinder.

The models (22) and (23) capture reasonable dynamic behavior, but they have a severe limitation for the present application, since the initial values of the states $(x, y, z)$ need to be carefully chosen in order to match the transient lift response. In practice, this transient response is precisely what we are interested in modeling, but these initial conditions are not known. Instead, these states are excited by external factors, such as disturbances or

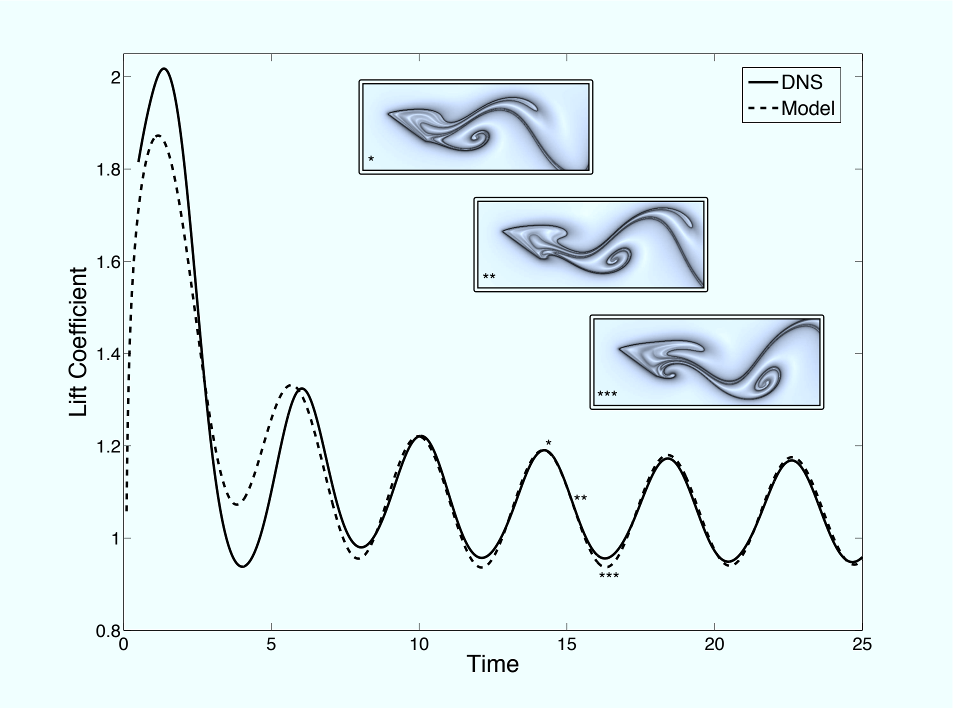


Figure 19: Coefficient of lift vs. time for stationary plate at post-stall angle $\alpha = 35°$. The solid line represents the oscillating lift curve obtained from direct numerical simulation and the dashed line is a the output of a low order ODE model with a Hopf bifurcation and a lag. The inset panels show coherent structures in the flow field at instances of high, moderate and low lift.

changes in angle of attack $\alpha$ (which can lead to the formation of leading-edge vortices, as stated earlier). One idea for incorporating these effects is to add impulsive forcing to the model $\dot{\mathbf{x}} = \ldots + \mathbf{f}(\alpha, \dot{\alpha})$, so that the terms $\mathbf{f}$ can excite the states $\mathbf{x}$ through a change of angle of attack. However, this approach was not pursued in this project, as the approaches of Sections 5 and 6 seemed more fruitful.

## 4.2 Models using POD and Galerkin projection

In this section, we summarize our results determining unsteady aerodynamic models using Proper Orthogonal Decomposition (POD) and Galerkin projection. These should not be confused with the models obtained using Balanced POD (Section 5), which yielded the superior models in Section 6.

The evolution of a flow field via direct numerical simulation may be viewed as a high dimensional dynamical system $\dot{u} = X(u)$, where $u$ is a state variable representing the velocity components at each spatial location, arranged in a long vector. Therefore, $u \in \mathbb{R}^N$ where $N$ is the number of grid points times the number of flow variables.

To obtain a reduced order model, it is first necessary to construct a low dimensional subspace $S \subset \mathbb{R}^N$ on which the dynamics may be projected. Given a time sequence of unsteady velocity fields $\{u_k \in \mathbb{R}^N\}_{k=1}^M$ from DNS, we seek a projection $P_S : \mathbb{R}^N \to S$ so that the projection error $\frac{1}{M}\sum_{k=1}^M(\|u_k - P_S u_k\|)$ is minimized. It has been shown that this minimization problem is equivalent to the eigenvalue problem

$$R\varphi = \lambda\varphi \qquad \text{where} \quad R = XX^* \qquad \text{and} \quad X = \begin{bmatrix} \vdots & \vdots & & \vdots \\ u_1 & u_2 & \ldots & u_M \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

$R$ is a real, symmetric matrix of dimension $N \times N$ with at most $M$ nonzero eigenvalues. Therefore, the eigenvalue problem $R\varphi = \lambda\varphi$ is equivalent to a simpler eigenvalue problem

$$U\varphi = \lambda\varphi \qquad \text{where} \quad U = X^*X$$

of dimension $M \times M$. For $M \ll N$, this greatly reduces the computation and is known as the method of snapshots. $\{\varphi_k\}_{k=1}^m$ are eigenfunctions associated with the $m$ largest eigenvalues of $R$ (or $U$) and are known as POD modes. POD modes are typically computed after subtracting the mean flow from each of the snapshots $u_i$. Because of the form of the minimization problem posed, the first $k$ POD modes are the $k$ most Energy-containing modes. Although energetic modes are important, it has been shown that modes including only a small fraction of the total energy can be dynamically important[26].

Given dynamics $\dot{u} = X(u)$ and a projection $P_S$ onto a low dimensional subspace $S \subset \mathbb{R}^N$, it is now possible to project the discretized Navier-Stokes equations onto the subspace $S$, resulting in a low order dynamical system model for the full equations of motion:

$$\dot{r} = P_S X(r), \quad \text{where} \quad r(t) = \bar{\varphi} + \sum_{k=1}^{m} a_k(t)\varphi_k$$

The dynamics are now captured as a low dimensional ODE with the POD mode amplitudes

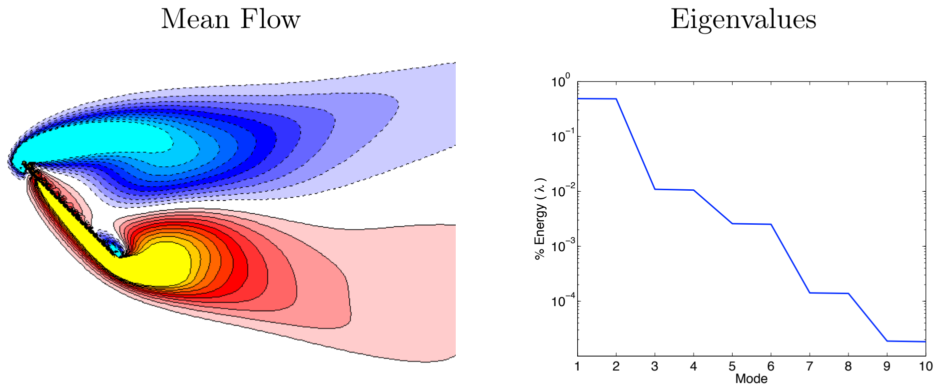


Figure 20: Mean flow and eigenvalues of POD modes for stationary flat plate at $\alpha = 45°$ and Re$= 100$.

as variables:

$$\langle \dot{r} - X(r), \varphi_k \rangle = 0 \quad \implies \quad \langle \dot{a}_j \varphi_j, \varphi_k \rangle - \langle X(r), \varphi_k \rangle = 0$$

$$\implies \quad \dot{a}_k = \langle X(r), \varphi_k \rangle \quad \text{for} \quad k = 1, \ldots, m$$

In the current study, $\dot{u} = X(u)$ is the momentum equation:

$$\dot{u} = \underbrace{-(u \cdot \nabla)u + \nu \nabla^2 u - \nabla p}_{X(u)}$$

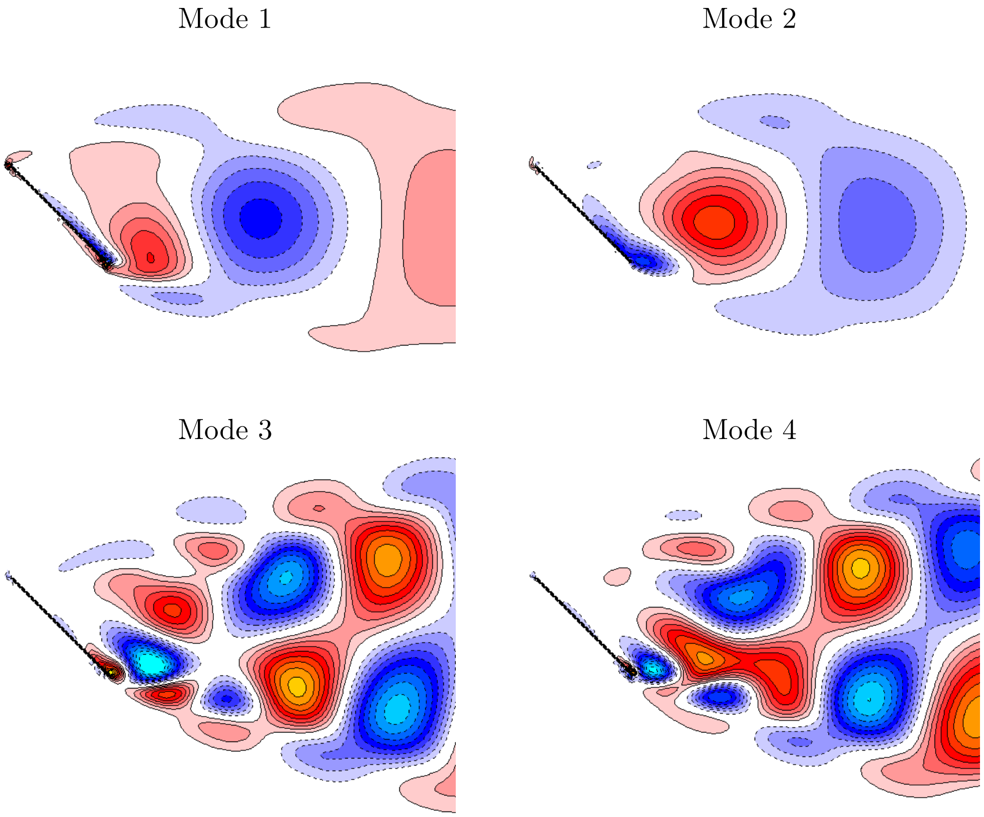
### *4.2.1 POD/Galerkin model, $\alpha = 45°$*

From simulation data of unsteady flow around a fixed plate at Re $= 100$ and $\alpha = 45°$ which are allowed to reach steady state vortex shedding, POD modes are computed and shown in Figure 21. In this configuration the plate sheds vorticity periodically from the leading and trailing edges. By first subtracting the mean flow, it is possible to obtain POD modes which are in energetic pairs as seen in the eigenvalue plot, Figure 20. Because of an approximate convective symmetry in the periodic shedding case, these POD modes come in pairs which appear to be phase shifted by $\pi/2$.

### *4.2.2 POD/Galerkin model, $\alpha = 30°$*

Figure 23 shows POD modes for the unsteady flow around a fixed plate at Re $= 100$ and $\alpha = 30°$. The eigenvalue plot, Figure 22, is similar to the $\alpha = 45°$ case, except that the pairs are not as closely matched.

A low-order model is obtained by Galerkin projecting the Navier-Stokes equations onto these modes. Using the projected dynamics, the POD mode coefficients are integrated forward in time and used to reconstruct an approximate flow field. From the approximate velocity field

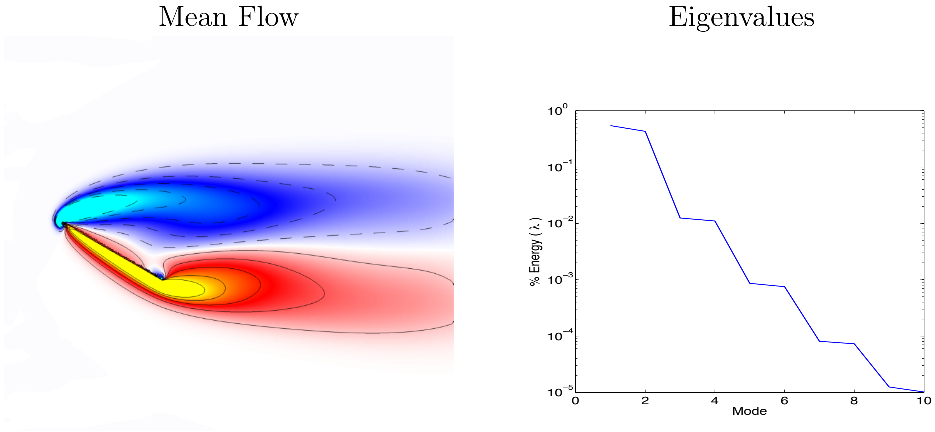

Mode 1

Mode 2

Mode 3

Mode 4

Figure 21: POD modes for stationary flat plate at $\alpha = 45°$ and Re$= 100$.


Mean Flow

Eigenvalues

Figure 22: Mean flow and eigenvalues of POD modes for stationary flat plate at $\alpha = 30°$ and Re$= 100$.

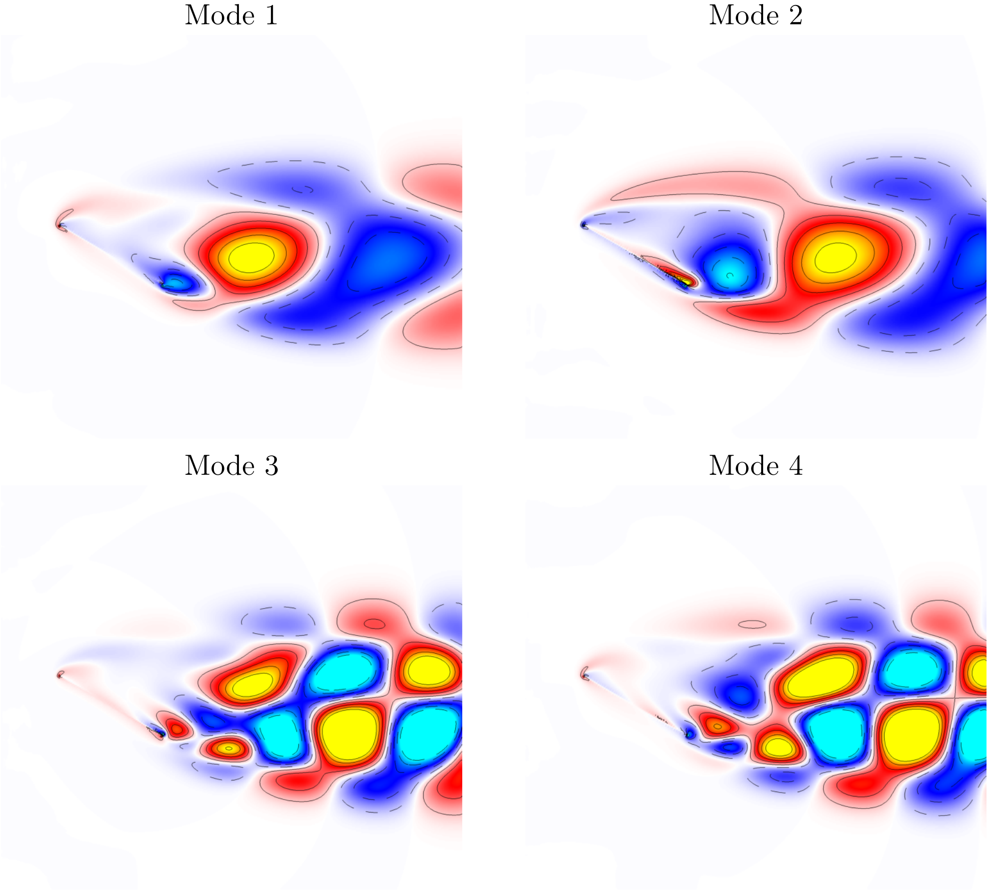


Figure 23: POD modes for stationary flat plate at $\alpha = 30^\circ$ and Re$= 100$.

reconstruction, an FTLE field is computed which agrees very well with the FTLE field from DNS, as can be seen in Figure 24. The comparison of POD mode amplitudes between DNS and projected dynamics is shown in Figure 25. It is interesting that the model agrees well with data for the first two modes, but not for the higher order modes. However, the fact that the Lagrangian coherent structures are preserved suggests that the first two modes are sufficient for reconstructing an accurate model. This is not entirely surprising considering the very simple sinusoidal vortex shedding pattern and force in time. For higher Reynolds number flows, the lift distribution and wake structures involve higher frequency oscillations which would presumably require more modes to approximate.

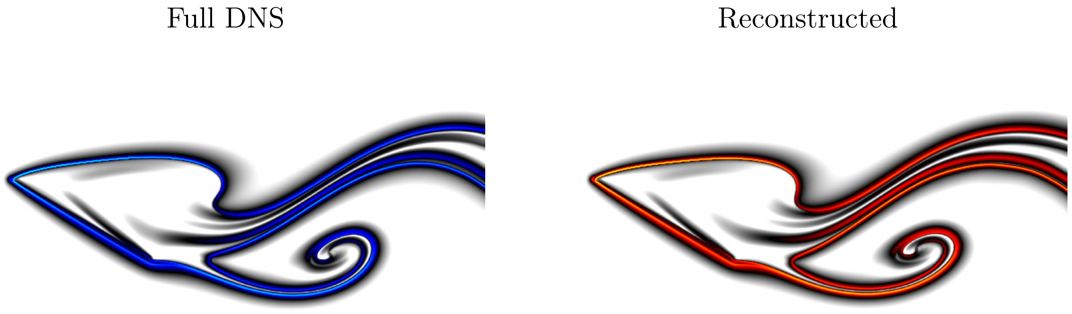


Figure 24: FTLE field for DNS vs. reconstruction from Galerkin projected dynamics for stationary flat plate at $\alpha = 30°$ and Re$= 100$.

# 5 Balanced model reduction

The models developed in the previous section demonstrate good agreement for the conditions at which they were calibrated: namely, they predict the unsteady lift forces at a single angle of attack. However, the phenomenological models of section 4.1 are obtained in an ad hoc fashion, and while the POD modes of section 4.2 are more systematic, it is not clear how to extend them to more general situations, for instance in which one has a gust that changes the angle of attack or speed of the freestream.

In order to represent these more complex, agile maneuvers, it is desirable to use a model reduction procedure that includes *inputs* and *outputs*: for instance, disturbances such as the instantaneous angle of attack or freestream speed can be regarded as inputs, and desired quantities such as lift and drag forces may be viewed as outputs. Furthermore, when cast in an input-output setting, these models may be readily used for control design.

Balanced truncation is an effective technique for obtaining reduced-order models of input-output systems. However, it is intractable for large systems such as fluids problems. An approximation of balanced truncation, called Balanced POD, was developed by Rowley [49], and has been shown to produce much more accurate models than standard POD/Galerkin models [26]. However, a criticism of this technique is that in order to obtain the models, one must simulate the *adjoint equations*, and adjoint simulations are not always readily available. (Of course, they are never available for experiments.)

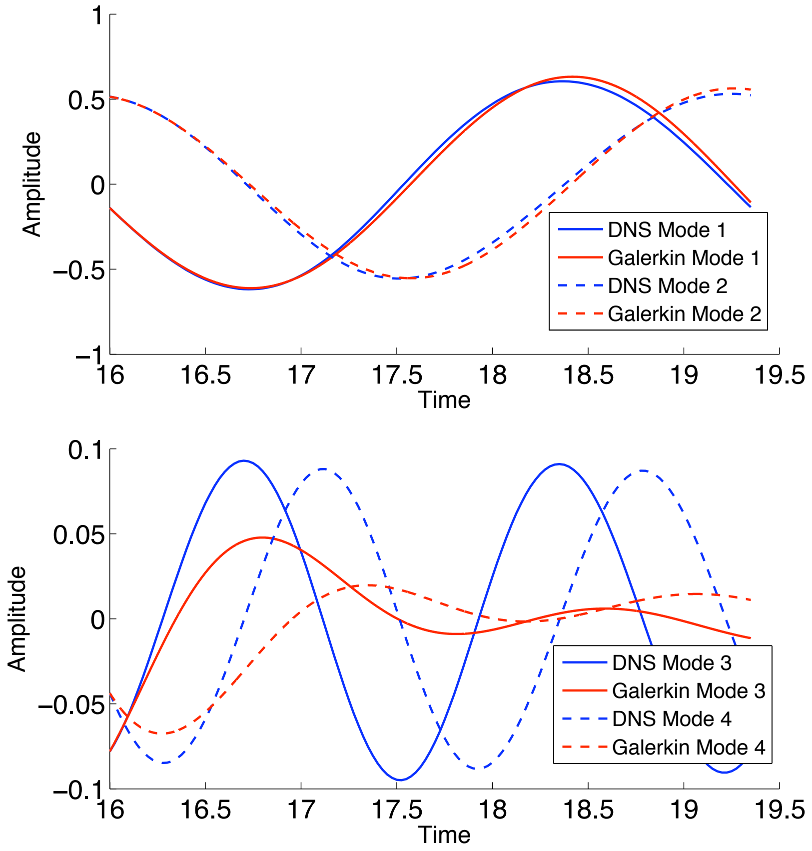


Figure 25: Mode amplitudes of DNS vs. reconstruction from Galerkin projected dynamics for stationary flat plate at $\alpha = 30°$ and Re= 100.

In the following subsection, we show how the Balanced POD method may be applied even when adjoint information is not available. A major result is that the models obtained by Balanced POD are precisely those obtained by an existing system identification method called the Eigensystem Realization Algorithm (ERA).

## 5.1 Balanced POD models without adjoints

In this section, we summarize the steps involved in approximate balanced truncation (balanced POD), and the Eigensystem Realization Algorithm, and show that they are equivalent.

Balanced truncation involves first constructing a a coordinate transformation that "balances" a linear input-output system, in the sense that certain measures of controllability and observability (the Gramian matrices) become diagonal and identical [43]. A reduced-order model is then obtained by truncating the least controllable and observable states, which correspond to the smallest diagonal entries in the transformed system. Unfortunately, the exact balanced truncation algorithm is not tractable for the large state dimensions encountered in fluid mechanics. However, an approximate, snapshot-based balanced truncation algorithm, referred to as Balanced Proper Orthogonal Decomposition (balanced POD) was proposed in [49], and has been used successfully in several examples [27, 1, 5].

The second technique, the eigensystem realization algorithm (ERA), has been used both for system identification and for model reduction, and it is well known that the models produced by ERA are approximately balanced [17, 29]. Here we show further that, theoretically, ERA produces exactly the same reduced order models as balanced POD. This equivalence indicates that ERA can be regarded as an approximate balanced truncation method, in the sense that, before truncation, it implicitly realizes a coordinate transformation under which a pair of approximate controllability and observability Gramians are exactly balanced. This feature distinguishes ERA from other model reduction methods that first realize truncations and then balance the reduced order models. Note that in ERA the Gramians, and the balancing transformation itself, are never explicitly calculated, as we will also show in the following discussions.

For both techniques, we will consider a high-dimensional, stable, discrete-time linear system, described by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k), \end{aligned} \tag{24}$$

where $k \in \mathbb{Z}$ is the time step index, $u(k) \in \mathbb{R}^p$ denotes a vector of inputs (for instance, actuators or disturbances), $y(k) \in \mathbb{R}^q$ a vector of outputs (for instance, sensor measurements, or simply quantities that one wishes to model), and $x(k) \in \mathbb{R}^n$ denotes the state variable (for instance, flow variables at all gridpoints of a simulation). These equations may arise, for instance, by discretizing the Navier-Stokes equations in time and space, and linearizing about a steady solution. The goal is to obtain an approximate model that captures the same relationship between inputs $u$ and outputs $y$, but with a much smaller state dimension:

$$\begin{aligned} x_r(k+1) &= A_r x_r(k) + B_r u(k) \\ y(k) &= C_r x_r(k) \end{aligned} \tag{25}$$

where the reduced state variable $x_r(k) \in \mathbb{R}^r$, $r \ll n$. We consider the discrete-time setting, because we are primarily interested in discrete-time data from simulations or experiments.

### *5.1.1 Balanced POD*

Here, we give only a brief overview of the balanced POD algorithm, and for details of the method, we refer the reader to [49]. The algorithm involves three main steps:

- **Step 1:** *Collect snapshots.* Run impulse-response simulations of the primal system (24) and collect $m_c + 1$ snapshots of *states* $x(k)$ in $m_c P + 1$ steps:

  $$X = \begin{bmatrix} B & A^P B & A^{2P} B & \cdots & A^{m_c P} B \end{bmatrix}, \tag{26}$$

  where $P$ is the sampling period. In addition, run impulse-response simulations for the adjoint system

  $$z(k+1) = A^* z(k) + C^* v(k) \tag{27}$$

  where the asterisk $^*$ stands for adjoint of a matrix, and collect $m_o + 1$ snapshots of states $z(k)$ in $m_o P + 1$ steps:

  $$Y = \left[ C^* \quad (A^*)^P C^* \quad (A^*)^{2P} C^* \cdots \quad (A^*)^{m_o P} C^* \right]. \tag{28}$$

  Calculate the generalized Hankel matrix,

  $$H = Y^* X. \tag{29}$$

- **Step 2:** *Compute modes.* Compute the singular value decomposition of $H$:

$$H = U\Sigma V^* = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} = U_1 \Sigma_1 V_1^* \tag{30}$$

  where the diagonal matrix $\Sigma_1 \in \mathbb{R}^{n_1 \times n_1}$ is invertible and includes all non-zero singular values of $H$, $n_1 = \text{rank}(H)$, and $U_1^* U_1 = V_1^* V_1 = I_{n_1 \times n_1}$. Choose $r \leq n_1$. Let $U_r$ and $V_r$ denote the sub-matrices of $U_1$ and $V_1$ that include their first $r$ columns, and $\Sigma_r$ the first $r \times r$ diagonal block of $\Sigma_1$. Calculate

$$\Phi_r = X V_r \Sigma_r^{-\frac{1}{2}}; \quad \Psi_r = Y U_r \Sigma_r^{-\frac{1}{2}}. \tag{31}$$

  where the columns of $\Phi_r$ and $\Psi_r$ are respectively the first $r$ primal and adjoint modes of system (24). The two sets of modes are bi-orthogonal: $\Psi_r^* \Phi_r = I_{r \times r}$.

- **Step 3:** *Project dynamics.* The system matrices in the reduced order model (25) are

$$A_r = \Psi_r^* A \Phi_r; \quad B_r = \Psi_r^* B; \quad C_r = C \Phi_r. \tag{32}$$

Note that the $n \times n$ controllability/observability Gramians are approximated by the matrices $XX^*$ and $YY^*$. The reduced-order model (25) is obtained by considering a subspace $x = \Phi_r x_r$, and projecting the dynamics (24) onto this subspace using the adjoint modes given by $\Psi_r$. It was shown in [49] that $\Phi_r$ and $\Psi_r$ respectively form the first $r$ columns of the balancing transformation/inverse transformation that *exactly* balance the approximate controllability/observability Gramians $XX^*$ and $YY^*$.

### *5.1.2 The eigensystem realization algorithm*

The eigensystem realization algorithm (ERA) was proposed in [28] as a system identification and model reduction technique for linear systems. The algorithm follows three main steps [28, 29]:

- **Step 1:** Run impulse-response simulations/experiments of the system (24) for $(m_c + m_o)P + 2$ steps, where $m_c$ and $m_o$ respectively reflect how much effect is taken for considering controllability and observability, and $P$ again is the sampling period. Collect the snapshots of the *outputs* $y$ in the following pattern:

$$\begin{aligned} \big(&CB, \quad CAB, \quad CA^P B, \quad CA^{P+1}B, \quad \ldots \\ &CA^{m_c P}B, \quad CA^{m_c P+1}B, \quad \ldots \quad CA^{(m_c+m_o)P}B, \quad CA^{(m_{c+m_o})P+1}B\big). \end{aligned} \tag{33}$$

  The terms $CA^k B$ are commonly called *Markov parameters.* Construct a generalized Hankel matrix $H \in \mathbb{R}^{q(m_o+1) \times p(m_c+1)}$

$$H = \begin{bmatrix} CB & CA^P B & \cdots & CA^{m_c P}B \\ CA^P B & CA^{2P}B & \cdots & CA^{(m_c+1)P}B \\ \vdots & \vdots & \ddots & \vdots \\ CA^{m_o P}B & CA^{(m_o+1)P}B & \cdots & CA^{(m_c+m_o)P}B \end{bmatrix}. \tag{34}$$

- **Step 2:** Compute SVD of $H$, exactly as in (30), to obtain $U_1$, $V_1$, $\Sigma_1$. Let $r \leq \text{rank}(H)$. Let $U_r$ and $V_r$ denote the sub-matrices of $U_1$ and $V_1$ that include their first $r$ columns, and $\Sigma_r$ the first $r \times r$ diagonal block of $\Sigma_1$.

- **Step 3:** The reduced $A_r$, $B_r$ and $C_r$ in (25) are then defined as

$$\begin{aligned} A_r &= \Sigma_r^{-\frac{1}{2}} U_r^* H' V_r \Sigma_r^{-\frac{1}{2}}; \\ B_r &= \text{the first } p \text{ columns of } \Sigma_r^{\frac{1}{2}} V_1^*; \\ C_r &= \text{the first } q \text{ rows of } U_r \Sigma_r^{\frac{1}{2}} \end{aligned} \tag{35}$$

where

$$H' = \begin{bmatrix} CAB & CA^{P+1}B & \cdots & CA^{m_c P+1}B \\ \vdots & \vdots & \ddots & \vdots \\ CA^{m_o P+1}B & CA^{(m_o+1)P+1}B & \cdots & CA^{(m_c+m_o)P+1}B \end{bmatrix}, \tag{36}$$

which can again be constructed directly from the collected snapshots (33).

### 5.1.3 *Theoretical equivalence between ERA and balanced POD*

The first observation is that, with $X$ and $Y$ given by (26) and (28), the generalized Hankel matrices obtained in balanced POD and ERA, respectively by (29) and (34), are theoretically identical. The theoretical equivalence between the two algorithms then follows immediately: First, $H'$ given in (36) satisfies $H' = Y^*AX$, which implies the matrices $A_r$ obtained in the two algorithms are identical. To show the equivalence of $B_r$, first note that the SVD (30) leads to $\Sigma_1^{-\frac{1}{2}} U_1^* H = \Sigma_1^{\frac{1}{2}} V_1^*$, which, by definition of $U_r$, $V_r$, $\Sigma_r$, implies $\Sigma_r^{-\frac{1}{2}} U_r^* H = \Sigma_r^{\frac{1}{2}} V_r^*$. (Note that it does *not* imply $H = U_r \Sigma_r V_r^*$, since $U_r U_r^*$ is not the identity.) Thus, in balanced POD, $B_r = \Psi_r^* B = \Sigma_r^{-\frac{1}{2}} U_r^* Y^* B$, which equals the first $p$ columns of $\Sigma_r^{-\frac{1}{2}} U_r^* H = \Sigma_r^{\frac{1}{2}} V_r^*$, which is the $B_r$ given by ERA. Similarly, the SVD (30) leads to $H V_1 \Sigma_1^{-\frac{1}{2}} = U_1 \Sigma_1^{\frac{1}{2}}$ and then $H V_r \Sigma_r^{-\frac{1}{2}} = U_r \Sigma_r^{\frac{1}{2}}$. Thus, in balanced POD, $C_r = C\Phi_r = C X V_r \Sigma_r^{-\frac{1}{2}}$, which equals the first $q$ rows of $H V_r \Sigma_r^{-\frac{1}{2}} = U_r \Sigma_r^{\frac{1}{2}}$, the $C_r$ given by ERA. In summary, we have:

**Main result.** *The reduced system matrices $A_r$, $B_r$ and $C_r$ generated in balanced POD and ERA, respectively by (32) and (35), are theoretically identical.*

In practice, these two algorithms may generate slightly different reduced order models, because the Hankel matrices calculated in the two algorithms are usually not exactly the same, due to small numerical inaccuracies in adjoint simulations, and/or in matrix multiplications needed to compute the sub-blocks in the Hankel matrices. In the following discussions, we compare these two algorithms in more detail.

### 5.1.4 *Comparison between ERA and balanced POD*

While ERA and balanced POD produce theoretically identical reduced-order models, the techniques differ in several important ways, both conceptually and computationally. Neither ERA nor balanced POD calculate Gramians explicitly, but balanced POD does construct approximate controllability and observability matrices $X$ and $Y^*$, from which one calculates the generalized Hankel matrix $H$ and balancing transformation. Balanced POD thus incurs additional computational cost, because one needs to construct the adjoint system (27), run adjoint simulations for $Y$, and then calculate each block of $H$ by matrix multiplication. Thus we see that the advantages of ERA include:

| Steps in computing reduced-order models | Approximate time (CPU hours) | |
|---|---|---|
| | balanced POD | ERA |
| 1. Linearized impulse response | 2 | 4 |
| 2. Computation of POD modes | 2 | 2 |
| 3. Adjoint impulse responses (10 in number) | 30 | - |
| 4. Computation of the Hankel matrix | 7 | 0.2 |
| 5. Singular value decomposition | 0.05 | 0.05 |
| 6. Computation of modes | 1 | - |
| 7. Computation of models | 0.02 | 0.02 |

Table 2: Comparison of the computational times required for various steps of the algorithms using balanced POD and ERA. The times are given for a 10-mode output projected system. The Hankel matrix is constructed using (a) 200 state-snapshots from each linearized and adjoint simulations for balanced POD, and (b) 400 Markov parameters (outputs) for ERA.

1. **Adjoint-free:** ERA is a feasible balanced truncation method for experiments, since it needs only the output measurements from the response to an impulsive input. Note that ERA has been successfully applied in several flow control experiments [11, 10], as a system-identification technique rather than a balanced-truncation method. In practice, input-output sensor responses are often collected by applying a broadband signal to the inputs, and the ARMARKOV method [3, 37] can then be used to identify the Markov parameters, or even directly the generalized Hankel matrix, from the input-output data history.

2. **Computational efficiency:** For large problems, typically the most computationally expensive component of computing balanced POD is constructing the generalized Hankel matrix $H$ in (29), as this involves computing inner products of all of the (large) primal and adjoint snapshots with each other. ERA is significantly more efficient at constructing the matrix $H$ in (34), since only the first row and last column of block matrices, i.e., the $(m_c + m_o + 1)$ Markov parameters, need be obtained by matrix multiplication. All the other $m_c \times m_o$ block matrices in $H$ are copies of other blocks, and need not be recomputed. For balanced POD, the matrix $H$ is obtained by computing all the $(m_c + 1) \times (m_o + 1)$ matrix multiplications (inner products) between corresponding blocks in $Y^*$ and $X$ in (29). Thus, for example, if $m_c = m_0 = 200$, the computing time needed for constructing $H$ in ERA will be about only 1% of that in balanced POD. See Table 2 for a detailed comparison on computational efficiency between balanced POD and ERA in an example of the flow past an inclined flat plate, used in [39].

At the same time, balanced POD also provides its own advantages:

1. **Sets of bi-orthogonal primal/adjoint modes:** Balanced POD provides sets of bi-orthogonal primal/adjoint modes, the columns of $\Phi_r$ and $\Psi_r$. In comparison, without the adjoint system, ERA cannot provide the primal and adjoint modes. At best, the primal modes may be computed, using the first equation in (31), if the matrix $X$ (26) is stored (in addition to the Markov parameters). But the adjoint modes cannot be

computed without solutions of the adjoint system. In this sense, balanced POD incorporates more of the physics of the system (the two sets of bi-orthogonal modes), while ERA is purely based on input-output data of the system. The primal/adjoint modes together can be useful for system analysis and controller/observer design purposes in several ways: for instance, in flow control applications, a large-amplitude region from the most observable mode (the leading adjoint mode) can be a good location for actuator placement. Also, although balanced POD is a linear method, a nonlinear system can be projected onto these sets of modes to obtain a nonlinear low-dimensional model. For instance, the transformation $x = \Phi_r x_r$, $x_r = \Psi_r^* x$ can be employed to reduce a full-dimensional nonlinear model $\dot{x} = f(x)$ to a low-dimensional system $\dot{x}_r = \Psi_r^* f(\Phi_r x_r)$. Finally, if parameters (such as Reynolds number or Mach number) are present in the original equations, balanced POD can retain these parameters in the reduced-order models. When the values of parameters change, the reduced order model by balanced POD may still be valid and perform well; see [27] for an application to linearized channel flow.

2. **Unstable systems:** Balanced POD has been extended to neutrally stable [40] and unstable systems [1]. In those cases, one first calculates the right/left eigenvectors corresponding to the neutral/unstable eigenvalues of the state-transition matrix $A$, using direct/adjoint simulations. Using these eigenvectors, the system is projected onto a stable subspace and then balanced truncation is realized for the stable subsystem. ERA for unstable systems is still an open problem, if adjoint operators are not available. However, we note that, once the stable subsystem is obtained, ERA can still be applied to it and efficiently realize its approximate balanced truncation.

**ERA for systems with high-dimensional outputs.** The method of output projection proposed in [49] makes it computationally feasible to realize approximate balanced truncation for systems with high-dimensional outputs—for instance, if one wishes to model the entire state $x$, say the flow field in the entire computational or experimental domain. This method involves projecting the outputs onto a small number of POD modes, determined from snapshots of $y$ from the impulse-response dataset. This method can be directly incorporated into ERA as follows: First, run impulse response simulations of the original system and collect Markov parameters as usual. Then, compute the leading POD modes of the dataset of Markov parameters and stack them as columns of a matrix $\Theta$. Left multiply those Markov parameters by $\Theta^*$ to project the outputs onto these POD modes. A generalized Hankel matrix is then constructed using these modified Markov parameters, and the usual steps of ERA follow.

## 5.2 Balanced truncation for periodic systems

In order to model vortex shedding phenomena, as arise for wings at high angles of attack, the techniques for balanced model reduction need to be extended to *periodic systems*. This extension is straightforward, but somewhat complex, and is the subject of the current section. In this context, it is most convenient to consider *discrete-time systems*, which may be viewed as a temporal discretization of the Navier-Stokes equations.

In particular, we consider linear discrete-time periodic systems of the form

$$x(k+1) = A(k)x(k) + B(k)u(k); \quad y(k) = C(k)x(k), \tag{37}$$

with state $x \in \mathbb{C}^n$, input $u \in \mathbb{C}^p$, output $y \in \mathbb{C}^q$, and $T$-periodic matrix coefficients $A(\cdot), B(\cdot), C(\cdot)$. The transition matrix in (37) is $F_{(j,i)} := A(j-1)A(j-2)\cdots A(i)$ for $j > i$, where $F_{(i,i)} = I_{n\times n}$. Periodicity implies that the eigenvalues of $F_{(j+T,j)}$ are independent of $j$. The *neutrally stable* case where the spectral radius $\rho(F_{(j+T,j)}) = 1$ will be discussed later. For now, assume the system is *exponentially stable*, i.e. $\rho(F_{(j+T,j)}) < 1$. The *controllability* and *observability* Gramians of (37) are then well defined and are $T$-periodic in $j$ [60]:

$$\begin{aligned} W_c(j) &:= \sum_{i=-\infty}^{j-1} F_{(j,i+1)}B(i)B(i)^* F^*_{(j,i+1)}; \\ W_o(j) &:= \sum_{i=j}^{\infty} F^*_{(i,j)}C(i)^*C(i)F_{(i,j)}, \end{aligned} \tag{38}$$

where $^*$ denotes the adjoint operator.

A standard *lifting* procedure [42] recasts (37) in $T$ input-output (I/O) equivalent LTI forms:

$$\begin{aligned} \tilde{x}_j(t+1) &= \tilde{A}_j\tilde{x}_j(t) + \tilde{B}_j\tilde{u}_j(t); \\ \tilde{y}_j(t) &= \tilde{C}_j\tilde{x}_j(t) + \tilde{D}_j\tilde{u}_j(t), \end{aligned} \tag{39}$$

with $j = 1, \ldots, T$, where $t$ is the time variable, $j$ parameterizes the lifted systems, the state $\tilde{x}_j(t) = x(j + tT)$ is periodically sampled from (37), the original inputs and outputs over each period are arranged as $\mathbb{C}^{pT}$ and $\mathbb{C}^{qT}$ column vectors $\tilde{u}_j(t) = [u(j + tT + i)]_{i=0}^{T-1}$ and $\tilde{y}_j(t) = [y(j + tT + i)]_{i=0}^{T-1}$, and the definitions of the constant matrices $\tilde{A}_j$, $\tilde{B}_j$, $\tilde{C}_j$ and $\tilde{D}_j$ readily follow from the variations of parameters formula in (37), e.g., $\tilde{A}_j = F_{(j+T,j)}$. Assuming exponential stability, the controllability and observability Gramians of the $j$-th lifted LTI system are

$$\tilde{W}_{jc} := \sum_{i=0}^{\infty} \tilde{A}^i_j\tilde{B}_j\tilde{B}^*_j\left(\tilde{A}^i_j\right)^*; \tilde{W}_{jo} := \sum_{i=0}^{\infty}\left(\tilde{A}^i_j\right)^*\tilde{C}^*_j\tilde{C}_j\tilde{A}^i_j. \tag{40}$$

The following statement follows from the periodicity of (37).

**Proposition 5.1.** *$\tilde{W}_{jc} = W_c(j)$ and $\tilde{W}_{jo} = W_o(j)$ for all $j = 1, \ldots, T$.*

Proposition 5.1 enables us to enjoy the best of both worlds: Whereas lifting enables an appeal to LTI balanced truncation in the lifted domain, as discussed through the remainder of the paper, Gramian computations can be carried in the original periodic setting, where the dimensions of the input and output spaces are much lower: $p$ and $q$ instead of $Tp$ and $Tq$.

#### *5.2.1 Factorization of empirical Gramians using snapshot-based matrices*

In snapshot-based methods [32, 49], the exact Gramians are substituted by approximate *empirical Gramians* where the infinite series in (38) are truncated [12, 61, 55] at a finite $m < \infty$:

$$\begin{aligned} W_{ce}(j;m) &:= \sum_{i=j-m}^{j-1} F_{(j,i+1)}B(i)B(i)^*F^*_{(j,i+1)}; \\ W_{oe}(j;m) &:= \sum_{i=j}^{j+m-1} F^*_{(i,j)}C(i)^*C(i)F_{(i,j)}. \end{aligned} \tag{41}$$

When the system is exponentially stable, truncation is justified by an induced norm bound on the truncation error, obtained by a geometric series argument and an appeal to Proposition 5.1:

**Lemma 5.2.** *Assume that the linear periodic system (37) is exponentially stable and let $m$ be an integer multiple of the period, $m = l\,T$. Then the following induced norm error bounds hold:*

$$\begin{aligned} \frac{\|W_c(j) - W_{ce}(j;m)\|}{\|W_c(j)\|} &\leqslant \|F^l_{(j+T,j)}\|^2; \\ \frac{\|W_o(j) - W_{oe}(j;m)\|}{\|W_o(j)\|} &\leqslant \|F^l_{(j+T,j)}\|^2. \end{aligned} \tag{42}$$

Empirical Gramians can be factorized using snapshot-based matrices.

**Proposition 5.3.** *Let $B^{(i)}$, $i = 1, \ldots, p$, denote the $i$-th column of $B$, and let $X^{(i)} \in \mathbb{C}^{n\times m}$ be defined as*

$$\begin{aligned} X^{(i)}(j;m) := \Big[&F_{(j,j-m+1)}B^{(i)}(j-m), \\ &F_{(j,j-m+2)}B^{(i)}(j-m+1), \ldots, B^{(i)}(j-1)\Big] \end{aligned}$$

*for each $j = 1, \ldots, T$ and the horizon length $m$. Finally, define the matrix of snapshots*

$$X(j;m) := \Big[X^{(1)}(j;m), \ldots, X^{(p)}(j;m)\Big] \in \mathbb{C}^{n\times m\,p}.$$

*Then $W_{ce}(j;m) = X(j;m)X(j;m)^*$.*

As illustrated in Figure 26(a), the columns of $X(j;m)$ are snapshots of impulse-response simulations of the system (37), justifying the term *empirical Gramian*: Invoking the $T$-periodicity of $B(\cdot)$ and $F(\cdot,\cdot)$(e.g. $F_{(j,j-m+T+1)} = F_{(j-T,j-m+1)}$), one observes that the $m$ columns of $X^{(i)}(m;j)$ are samples at times $j - kT$, $k = 0, \ldots, l-1$ of trajectories of simulations initiated at $x(j-m+t) = B^{(i)}(j-m+t-1)$, $t = 1, \ldots, T$, assuming $m = lT$. In total, $Tp$ simulations and $mp$ snapshots are needed to construct $X(j;m)$.

An analogous observation applies to the empirical observability Gramian.

**Proposition 5.4.** *Let $C^{(i)}$, $i = 1, \ldots, q$, denote the $i$-th row of $C$, and let $Y^{(i)} \in \mathbb{C}^{n\times m}$ be defined as*

$$\begin{aligned} Y^{(i)}(j;m) := \Big[&F^*_{(j+m-1,j)}C^{(i)}(j+m-1)^*, \\ &F^*_{(j+m-2,j)}C^{(i)}(j+m-2)^*, \ldots, C^{i}(j)^*\Big] \end{aligned}$$

*for each $j = 1, \ldots, T$ and the horizon length $m$. Finally, let*

$$Y(j;m) := \Big[Y^{(1)}(j;m), \ldots, Y^{(q)}(j;m)\Big] \in \mathbb{C}^{n\times m\,q}.$$

*Then $W_{oe}(j;m) = Y(j;m)Y(j;m)^*$.*

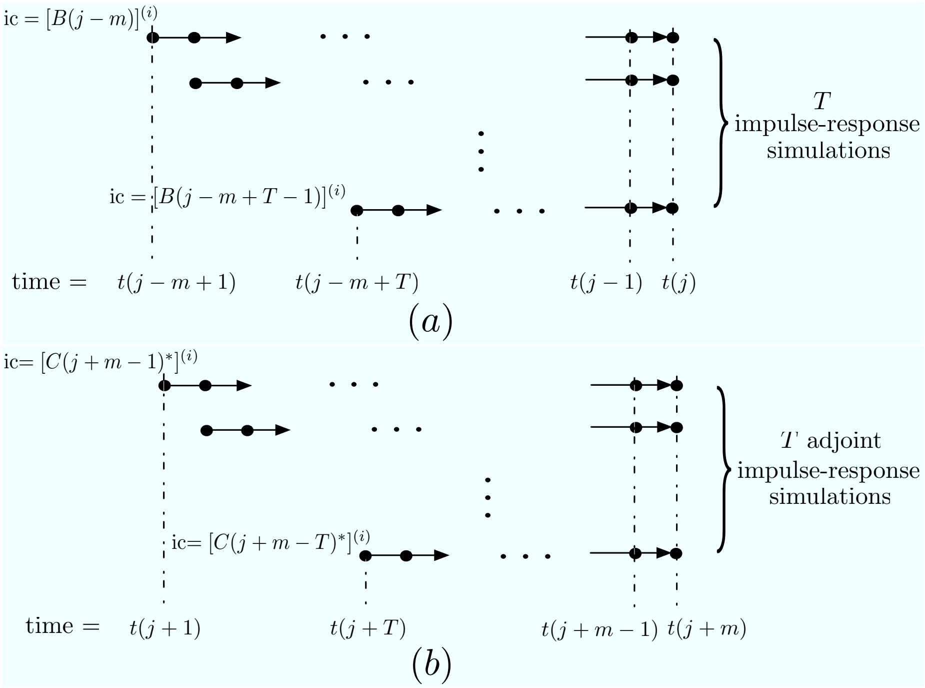


Figure 26: (a) The $T$ impulse-response simulations corresponding to the $i$-th control input. (b) The $T$ adjoint impulse-response simulations corresponding to the $i$-th adjoint control input.

As illustrated in Figure 26(b), $Y(j,m)$ can be obtained from simulations of the *adjoint system*

$$z(k+1) = \hat{A}(k)z(k) + \hat{C}(k)v(k) \tag{43}$$

where $k = j, \ldots, j+m-1$, $z \in \mathbb{C}^n$, $v \in \mathbb{C}^q$, $\hat{A}(k) := A(2j+m-k-1)^*$ and $\hat{C}(k) := C(2j+m-k-1)^*$. By periodicity, $Tq$ adjoint simulations, and in total $mq$ snapshots taken at time $j+kT$, $k = 1, \ldots, l$ are needed to construct $Y(j;m)$.

### 5.2.2 *Balanced truncation using the method of snapshots*

Fix a time $1 \leqslant j \leqslant T$. Justified by Propositions 5.3 and 5.4 and Lemma 5.2, let $X(j;m_c)$ and $Y(j;m_o)$ be computed, allowing $m_c \neq m_o$, as factors of the empirical Gramians $W_{ce}(j;m_c), W_{oe}(j;m_o)$. By Proposition 5.1, they can be also used as factors of the empirical Gramians of the $j$-th *lifted* system (39). The method of snapshots presented in [49] then leads to approximate balanced truncations in the lifted LTI setting, as follows: Compute the SVD $Y(j;m_o)^*X(j;m_c) = U\Sigma V^*$, and the transformations $\Phi$, $\Psi$ that *exactly* balance the empirical Gramians of the lifted system

$$\Phi = X(j;m_c)V\Sigma^{-1/2}; \ \Psi = Y(j;m_o)U\Sigma^{-1/2}. \tag{44}$$

Let $\Phi_r, \Psi_r$ be the first $r$ columns of $\Phi$ and $\Psi$, comprising the leading bi-orthogonal balancing and adjoint modes of the $j$-th lifted system. ( Note that to simplify notation, the dependence of $\Phi$, $\Psi$, $\Phi_r$, $\Psi_r$ on $j$ is suppressed. ) The reduced state $\tilde{z}_j(t) \in \mathbb{C}^r$ is defined by the projection $\tilde{z}_j(t) = \Psi_r^*\tilde{x}_j(t) = \Psi_r^*x(j+tT)$ and the estimated full state $x(j+tT) \approx \Phi_r\tilde{z}_j(t)$.

The reduced model of order $r$, in the lifted setting, reads

$$\begin{aligned} \tilde{z}_j(t+1) &= \Psi_r^* \tilde{A}_j \Phi_r \tilde{z}_j(t) + \Psi_r^* \tilde{B}_j \tilde{u}_j(t); \\ \tilde{y}_j(t) &= \tilde{C}_j \Phi_r \tilde{z}_j(t) + \tilde{D}_j \tilde{u}_j(t). \end{aligned} \tag{45}$$

I/O equivalence of (37) to the lifted (39) means that the reduced-order system provides the sought I/O approximation of (37). Note that improved numerical stability of the computations above can be achieved by first representing each of the factors $X(j; m_c)$ and $Y(j; m_o)$ in terms of leading orthogonal bases, obtained, e.g., by SVD or by Krylov methods.

We comment in closing on the possibility to "un-lift" the reduced-order lifted system. As discussed in [60], the exact Gramians solve an allied periodic Lyapunov equation, thus providing an exact periodic balancing and an "un-lifted" balanced truncation in the periodic setting. Using the method of snapshots, There are two computational shortcoming to that approach in the current problem. First, the computational burden is high when $T \gg 1$. Second, the truncated empirical Gramians used here do not form an exact solution of the periodic Lyapunov equation. Un-lifting is nonetheless a simple task if the balancing requirement is limited to the periodically sampled system (i.e., to a lifted system for one, fixed $j$). The following inductive procedure is one possible solution: Fix $\Phi(j) := \Phi_r$ and $\Psi(j+T-1) := \Psi_r$. Let $P(j+i)$ be the rank-$r$ orthogonal projection on $\mathcal{I}m(F_{(j+i,j)}\Phi(j))$ and let $\Phi(j+i+1) = \Psi(j+i) \in \mathbb{C}^{n\times r}$, $i = 0, \ldots, T-2$, satisfy $P(j+i) = \Phi(j+i+1)\Psi(j+i)^*$. Then a periodic realization of the reduced order system is defined with $A_r(k) := \Psi(k)^* A(k)\Phi(k)$, $B_r(k) := \Psi(k)^* B(k)$ and $C_r(k) := C(k)\Phi(k)$.

### 5.2.3 *Output projection method*

The computations delineated above require an untenable number of adjoint simulations when very high dimensional outputs are considered; e.g., when the output is set identical to the state, such that one can use state response data in design of an optimal controller (e.g. linear-quadratic regulator) or to analyze system dynamics in detail. In the LTI case [49] proposed to project the output on the (few) leading POD modes of the dataset formed by the impulse response simulations. Thus one invokes the *kinematic* significance of POD modes, to reduce the dimension of the output space, but avoids the weakness of standard POD models that use them as *dynamic* states. Here we extend the output projection method to periodic systems.

The I/O map of the $j$-th lifted LTI system (39) is determined by the $Tq \times Tp$ dimensional impulse-response matrices $\{\tilde{G}_j(t)\}$. The *output-projected* lifted system

$$\begin{aligned} \tilde{x}_j(t+1) &= \tilde{A}_j \tilde{x}_j(t) + \tilde{B}_j \tilde{u}_j(t); \\ \tilde{y}_j(t)_P &= \tilde{P}_j \left( \tilde{C}_j \tilde{x}_j(t) + \tilde{D}_j \tilde{u}_j(t) \right), \end{aligned} \tag{46}$$

is designed to best approximate the exact impulse response of the original lifted system. Ideally, the low-rank orthogonal projection matrix $\tilde{P}_j$ should thus satisfy

$$\tilde{P}_j = \underset{\{\tilde{P}_j \in \mathcal{P}_{\tilde{r}_{op}}\}}{\operatorname{argmin}} \left( \sum_{t=0}^{\infty} ||\tilde{G}_j(t) - \tilde{P}_j \tilde{G}_j(t)||^2 \right), \tag{47}$$

where $\mathcal{P}_{\tilde{r}_{op}}$ is the space of orthogonal projections of rank $\tilde{r}_{op} \ll Tq$. When the Frobenius norm $||\cdot||_F$ is used in (47), it becomes a standard projection problem. Its solution is

$\tilde{P}_j = \tilde{\Theta}_j \tilde{\Theta}_j^*$, where the columns of $\tilde{\Theta}_j$ are the leading $\tilde{r}_{op}$ POD modes of the datasets $\{\tilde{G}_j(i)\}_{i=0}^{\infty}$.

As described above, the optimal $\tilde{P}_j$ is generically a full matrix. Thus, $\tilde{y}_j(t)_P = \tilde{P}_j \tilde{y}_j(t)$ is no longer the lifted representation of the output of a periodic system, and the projected system cannot be “un-lifted". Rather, for each $t$, the value of $\tilde{y}_j(t)_P$ is determined by the original response along an entire period. In particular, we lose the ability to compute the Gramian in the original periodic setting. To avoid this problem we impose on (47) the additional condition that the projection has a block diagonal form

$$\tilde{P}_j = \operatorname{diag}\left[\tilde{P}_j(1), \cdots, \tilde{P}_j(T)\right] \tag{48}$$

where each $q \times q$ diagonal block is a rank-$r_{op}$ orthogonal projection with $\tilde{r}_{op} = r_{op}T$. This enables to un-lift the projected lifted system (46) to an *output-projected time-periodic system*

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k); \\ y(k)_P &= P(k)C(k)x(k), \end{aligned} \tag{49}$$

where the $T$-periodic, rank-$r_{op}$ orthogonal projection $P$ is defined by $P(j+tT+i) = P(j+i) := \tilde{P}_j(i+1), \quad i = 0, \dots, T-1$. The constrained optimization problem (47)-(48) is solved as an equivalent set of unconstrained problems in the periodic setting, invoking the correspondence of the $T$, $q \times pT$ dimensional blocks of $\tilde{G}_j(t)$, $G(j+tT+i, j)$, $i = 0, \dots, T-1$, to the impulse response of (37), as detailed in [6]:

**Proposition 5.5.** *Using the Frobenius norm, the solution of the constrained optimization problem (47) and (48) is equivalent to the combined solution of the problems*

$$\begin{aligned} \tilde{P}_j(i+1) = \underset{\{\tilde{P}_j(i+1) \in \mathcal{P}_{r_{op}}\}}{\operatorname{argmin}} \Bigg( \sum_{t=0}^{\infty} \Big\| & G(j+tT+i, j) \\ & - \tilde{P}_j(i+1) G(j+tT+i, j) \Big\|_F^2 \Bigg), \end{aligned}$$

*for* $i = 0, \dots, T-1$.

*Proof.* By a reduction to a standard projection problem. ☐

The computation of the structurally constrained optimal $\tilde{P}_j$ of the form (48) is thus reduced to $T$ unconstrained optimization problems for each $P(k)$, $k = j, \cdots, j+T-1$, in the periodic setting. Following standard POD rationale, the solutions are $P(k) = \Theta(k)\Theta(k)^*$, where the $r_{op}$ columns of $\Theta(k)$ are the leading POD modes of the dataset $\{G(tT+k, j)\}_{t=0}^{\infty}$, and the approximation error between the output-projected system and the original system is

$$\begin{aligned} & \sum_{t=0}^{\infty} ||\tilde{G}_j(t) - \tilde{P}_j \tilde{G}_j(t)||_F^2 \\ & = \sum_{i=j}^{j+T-1} \sum_{t=0}^{\infty} \left\| G(j+tT+i, j) - \tilde{P}_j(i+1) G(j+tT+i, j) \right\|_F^2 \\ & = \sum_{i=j}^{j+T-1} \sum_{m=r_{op}+1}^{q} \lambda(i)_m \end{aligned}$$

where for each $i$, $\lambda(i)_1, \ldots, \lambda(i)_q$ are the descending-ordered eigenvalues $\sum_{t=0}^{\infty} G(tT+i,j)G(tT+i,j)^*$. The POD modes can be computed by the method of snapshots [56], applied to datasets comprising the columns of the impulse-response matrices $\{G(tT+i,j)\}_{t=0}^{s}$. Conveniently, provided that $m_c \geqslant (s+1)T$, periodicity implies that data required to compute these snapshots have already been obtained during the computation of $X(j;m_c)$, as described in § 5.2.1. For instance, the matrix $C(j)X(j;m_c)$ includes the columns of matrices $\{G(j+tT,j)\}_{t=1}^{m_c/T}$.

The empirical factor $Y(j;m_o)$ of the corresponding observability Gramian

$$W_{oP}(j) = \sum_{i=j}^{\infty} F_{(i,j)}^{*} C(i)^{*} \Theta(i) \Theta(i)^{*} C(i) F_{(i,j)}$$

is needed in order to realize the snapshot-based approximate balanced truncation for the output-projected system (49). This is accomplished with only $Tr_{op}$ ($r_{op} \ll q$) impulse-response simulations of the adjoint time-periodic system corresponding to the output-projected system (49), whose control input is $r_{op}$-dimensional.

In closing we note that, for additional simplicity and a requirement of a single SVD computation, one can also use a single, time-invariant output projection. Under this constraint, the optimal selection is $P = \Theta\Theta^*$, where the columns of $\Theta$ are the leading POD modes of the entire impulse-response $\{\{G(tT+k,j)\}_{t=0}^{s}\}_{k=j}^{j+T-1}$ of (37). This stronger constraint implies further reduction in matching, when compared with the optimal solution in the lifted domain.

### 5.2.4 Summary: procedures of balanced POD for periodic systems

Following the terminology in [49], the approximate balanced truncation method for linear, time-periodic systems is termed a *lifted balanced POD*. Its main steps include:

- Step 0: Fix a time $j$, $1 \leqslant j \leqslant T$, as the time point for lifting.
- Step 1: Run $Tp$ impulse-response simulations to obtain $m_c p$ snapshots and form the $n \times m_c p$ dimensional $X(j;m_c)$ as described in § 5.2.1.
- Step 2: Compute $y = Cx$ from stored states in simulations carried to compute $X(j;m_c)$. Solve for the POD problems for the periodically sampled $y(j+tT+i)$, to obtain the output-projection matrices $\Theta(j+i)$, $i = 0, \cdots, T-1$.
- Step 3: Run $Tr_{op}$ impulse-response simulations of the adjoint output-proejcted system, to form the $n \times m_o r_{op}$ dimensional matrix $Y(j;m_o)$ as described in § 5.2.1.
- Step 4: Compute the SVD of $Y(j;m_o)^* X(j;m_c)$ and the balancing modes for the lifted system given by (44).
- Step 5: Compute the reduced lifted system (45).

Variants include skipping Step 2, when the output dimension $q$ is small, and using a single, time-invariant output projection, as discussed in § 5.2.3. The reduced system can be unliftted to a periodic system, e.g., as described in closing § 5.2.2. As in [49], an obvious dual version of the algorithm addresses the case of a high-dimensional input space, with only few outputs. This case is motivated by systems susceptible to distributed disturbances, simultaneously effecting the entire state (e.g., $B = I$).

### 5.2.5 *The neutrally stable case*

Consider a linear periodic system (37) that arises from linearization of a system around an asymptotically stable periodic orbit. By Floquet theory [25], in this case $\tilde{A}_j = F_{(j+T,j)}$ is only neutrally stable, due to *one* unity eigenvalue that corresponds to persisting perturbations along the periodic orbit in the linearization. Balanced truncation cannot be directly applied to a neutrally stable system, as the infinite series used to define Gramians may diverge.

[1] presented an extended version of balanced POD for unstable LTI systems that have small unstable dimensions. Following the idea presented in [66], it decomposes the system dynamics into stable and unstable parts. Then it applies approximate balanced truncation to the stable dynamics while keeping the unstable dynamics exactly. This method is conceptually applied here to periodic systems through the lifted setting, with all computations executed in the periodic setting. First, for a given lifting time $j$, define a projection onto the stable subspace $E^s\left(\tilde{A}_j\right)$ by $\mathbb{P}_j = I_{n\times n} - \frac{v_j w_j^*}{w_j^* v_j}$ where $w_j, v_j \in \mathbb{C}^n$ are the left/right eigenvectors of $\tilde{A}_j$ corresponding to the unity eigenvalue. Dynamics of the neutrally stable lifted system (39) is thus restricted to the stable subspace of $\tilde{A}_j$:

$$\begin{aligned} \tilde{x}_j(t+1)_s &= \tilde{A}_j \tilde{x}_j(t)_s + \mathbb{P}_j \tilde{B}_j \tilde{u}_j(t); \\ \tilde{y}_j(t)_s &= \tilde{C}_j \mathbb{P}_j \tilde{x}_j(t)_s + \tilde{D}_j \tilde{u}_j(t), \end{aligned} \tag{50}$$

where $\tilde{x}_j(t)_s = \mathbb{P}_j \tilde{x}_j(t)$. Lifted balanced POD can be realized to this projected system describing stable dynamics. Let $\Phi^s_{r_s}$ and $\Psi^s_{r_s}$ be the matrices including the leading $r_s$ balancing and adjoint modes of the projected system (50). Then, a reduced model of order $r$, $r = r_s + 1$, for the neutrally stable lifted system (39) can be obtained in the form of (45), where now $\Phi_r = \begin{bmatrix}\Phi^s_{r_s} & v_j\end{bmatrix}$; $\Psi_r = \begin{bmatrix}\Psi^s_{r_s} & \frac{w_j}{w_j^* v_j}\end{bmatrix}$. The reduced system keeps the one-dimensional neutrally stable dynamics exactly, while the exponentially stable dynamics is reduced to the order of $r_s$.

Numerically, the neutrally stable eigenvectors of $\tilde{A}_j$ can be calculated using a Krylov method, or even the power method: By running a control-free simulation of the periodic system (37) with an arbitrary initial condition $x_{(j)} \notin E^s\left(\tilde{A}_j\right)$, one can approximate $v_j$ by $x(j+lT)$, with a large $l$. Similarly, a long-time control-free simulation of the adjoint periodic system (43) is needed to approximate $w_j$. Then, when computing the transformations $\Phi^s_{r_s}$ and $\Psi^s_{r_s}$ for the projected system (50), one follows exactly the same procedures given in § 5.2.4. The only difference is that in the $Tp$ simulations of the periodic system (37) described in § 5.2.1, the states should be projected onto $E^s\left(\tilde{A}_j\right)$ by $\mathbb{P}_j$ at time $j - m + T$. The simulations then resume with these states as new initial conditions. Similarly, in the adjoint simulations, the adjoint states should be left-multiplied by $\mathbb{P}_j^*$ at time $j + T$ before the simulations resume.

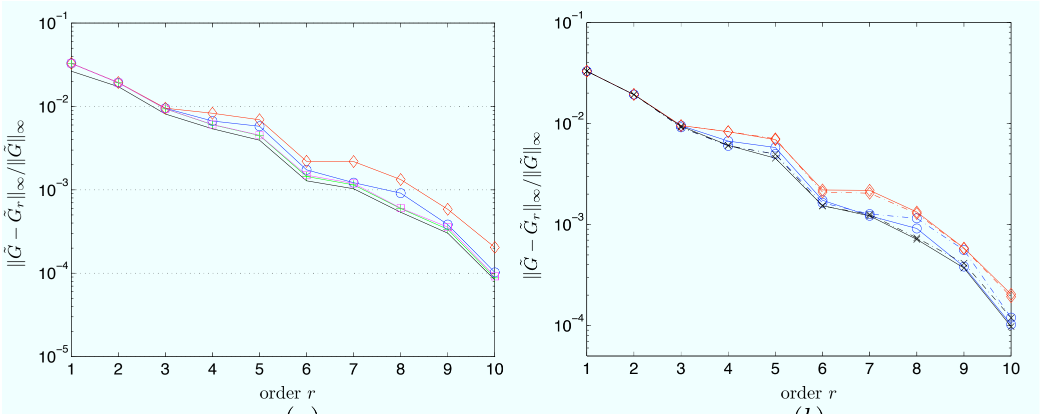
By construction, this method is applicable to other neutrally stable/unstable periodic systems, with small neutrally stable/unstable dimensions. For unstable systems, in impulse-response simulations one can repeatedly project the states once each period, using $\mathbb{P}_j$, to numerically confine the dynamics to the stable invariant subspace.

### 5.2.6 *Numerical example*

To illustrate the balanced POD algorithm, consider an exponentially stable example (similar to that in [14]): a linear periodic system (37) with period $T = 5$, state dimension $n = 30$, output dimension $q = 30$, control input dimension $p = 1$, and $\{A(k)\}_{k=1}^5$ are randomly generated diagonal matrices with diagonal entries bounded in $[0.16, 0.96]$, guaranteeing asymptotic stability. The matrices $B(k)$ and $C(k)$ are also randomly generated, with entries bounded in $[0, 1]$.

Here we pick the "lifting time" $j = 1$. Choose $m_c = m_o = 3T = 15$. Figure 27(a) shows the error plots of the infinity norm, $||\tilde{G} - \tilde{G}_r||_\infty/||\tilde{G}||_\infty$ versus $r$, the order of the reduced lifted system. Here $\tilde{G}_r$ is the impulse-response matrix of the reduced lifted system of order $r$. We see that the snapshot-based balanced truncation gives a good approximation of exact balanced truncation. Further, the balanced POD, even with low orders of output projection $r_{op}$, generates satisfying results. Recall that, for the lifted system, the order of output projection is $\tilde{r}_{op} = r_{op}T$.

Figure 27(b) shows comparisons between balanced POD results with the same order of output projection, one set based on $T$-periodic projection matrices along one period, and the other using single time-invariant projection matrix (see § 5.2.3). For the cases where $r_{op}$ are low, these two approaches give almost identical results, or even the latter one gives better results. However, when the order of output projection $r_{op}$ increases, the results based on $T$-periodic projection matrices are better than those by a single projection matrix, as we expect.



$(a)$ $(b)$

Figure 27: Error $||\tilde{G} - \tilde{G}_{\tilde{r}}||_\infty/||\tilde{G}||_\infty$, for lifted balanced POD approach: (a) For exact balanced truncation(+), balanced truncation by the method of snapshots but without output projection(□), balanced POD with $r_{op} = 1$ (◇), balanced POD with $r_{op} = 3$ (∘), and the lower bound for any model reduction scheme (−). All output projections are $T$-periodic. (b) Time varying $T$-periodic output projections versus time-invariant output projections: balanced POD with $r_{op} = 1$ (◇), balanced POD with $r_{op} = 3$ (∘) and balanced POD with $r_{op} = 5$ (+). Solid lines correspond to cases using $T$-periodic projection matrices, and dashed lines using one single projection matrix.

This algorithm has also been applied to a neutrally stable, time-periodic system obtained by linearizing the Ginzburg-Landau partial differential equation about its exponentially stable time-periodic solution; see [40].

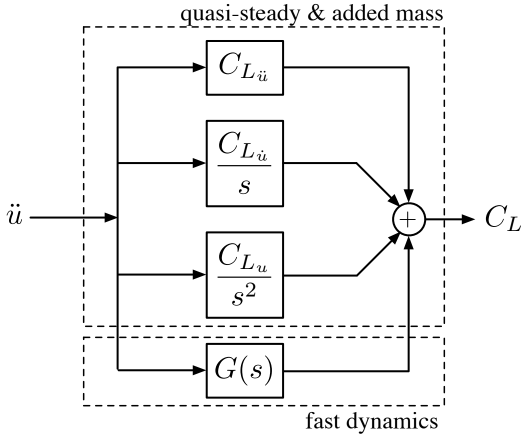


Figure 28: Schematic for reduced order model of Wagner's indicial response.

# 6 Control-oriented extensions of classical unsteady models

The classical models described in Section 2 are of limited use for control design: Theodorsen's models (Sec. 2.1) are valid only for sinusoidal maneuvers, and Wagner's models (Sec. 2.2) require a cumbersome convolution integral.

The main result of this section is a low-dimensional, state-space representation of unsteady aerodynamic forces, patterned after Wagner's indicial response model. Step-response simulations in either angle of attack $\alpha$ or vertical position $h$ are performed using an immersed boundary projection method. Based on the lift coefficient history from these simulations, the model reduction methods discussed in Section 5.1 (specifically, the Eigensystem Realization Algorithm) is used to construct a state-space model in the following form:

$$\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_{k+1} = \begin{bmatrix} A_r & 0 & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_{k} + \begin{bmatrix} B_r \\ 0 \\ \Delta t \end{bmatrix} \ddot{\mathbf{u}}_k$$

$$C_L(k\Delta t) = \begin{bmatrix} C_r & C_{L_{\mathbf{u}}} & C_{L_{\dot{\mathbf{u}}}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_{k} + C_{L_{\ddot{\mathbf{u}}}}\, \ddot{\mathbf{u}}_k \tag{51}$$

where $\mathbf{u} = \alpha$ for pitching, $\mathbf{u} = h$ for plunging, and $\mathbf{u} = \begin{bmatrix} \alpha & h \end{bmatrix}^T$ for pitching and plunging.

For the system to be proper, we choose $\ddot{\mathbf{u}}$ as the input[*]. Furthermore, $\mathbf{u}$ and $\dot{\mathbf{u}}$ appear explicitly in the state, since there is a quasi-steady dependence on these variables. The additional fast dynamics, modeled by $G(s)$, are obtained using the eigensystem realization algorithm (ERA), resulting in a low-dimensional state-space representation, $(A_r, B_r, C_r)$, of order $r$. This is shown in Figure 28.

The output of the model is the lift coefficient, $C_L$, and the model structure is conveniently expressed in terms of the stability derivatives, $C_{L_{u_i}} \cdot u_i$, and the additional fast dynamics,

---

[*] There are added mass forces proportional to $\ddot{\mathbf{u}}$, except in the degenerate case of pitching about the middle-chord.

$C_r \cdot \mathbf{x}$:

$$C_L = C_{L_{\mathbf{u}}} \cdot \mathbf{u} + C_{L_{\dot{\mathbf{u}}}} \cdot \dot{\mathbf{u}} + C_{L_{\ddot{\mathbf{u}}}} \cdot \ddot{\mathbf{u}} + C_r \cdot \mathbf{x} \tag{52}$$

The reduced-order model in Eq. (51) is shown to agree well with Wagner's indicial response over a range of frequencies and maneuvers. The goal of this section is not to demonstrate the relevance or accuracy of Wagner's model, but rather to demonstrate that it may be cast in the convenient state-space form, either for use itself, or as the foundation for a more sophisticated model. However, once it is observed that the reduced order model faithfully reconstructs Wagner's indicial response, comparisons with DNS for large-amplitude maneuvers provide interesting context.

## 6.1 Reduced-order approximations of Wagner's indicial response

This section details the process by which we obtain the reduced order model (ROM) in (51) for the lift force on an airfoil as a function of motion input variables, angle of attack $\alpha$ and vertical position $h$. In particular, given step-response experiments, we first subtract off quasi-steady and added-mass forces in Section 6.1.1, and then model the remainder of the fast dynamics using the eigensystem realization algorithm (ERA) in Section 6.1.2. A step-by-step summary of the process is presented in Section 6.1.4.

Before proceeding with the details of the method, it is important to highlight two main characteristics of aerodynamic step-response experiments that motivate the form of the reduced order model. First, step responses in certain variables, such as angle of attack and vertical velocity, result in nonzero steady-state lift. Second, the aerodynamic response to smoothed step inputs, discussed in Section 6.1.3, are dominated by forces proportional to the second derivatives of the configuration variables, $\ddot{\alpha}$ and $\ddot{h}$, for the duration of the step maneuver.

The case of pitching about the middle chord is an exception to characteristic 2 above, and a step in vertical position violates characteristic 1 above, and Section 6.1.5 addresses this special case. Finally, the theory is presented for the case of single-input single-output (SISO) systems in one configuration variable, either $\alpha$ or $h$; however, the theory generalizes to multiple-input multiple-output (MIMO) systems, which is briefly discussed in Section 6.1.6.

### *6.1.1 Aerodynamic impulse response and stability derivatives*

The following discussion provides a theoretical foundation relating the various transfer functions, state space realizations and impulse-response simulations. In particular, a step response in a given input variable $\mathbf{u}$ may be viewed as an impulse-response in the time derivative of that variable $\dot{\mathbf{u}}$. Moreover, in the instance when the output $y$ depends on $\ddot{\mathbf{u}}$, it is necessary for the input to be $\ddot{\mathbf{u}}$, since there is no way to represent a derivative in state-space form.

In the discussion that follows, it may be helpful to think of $\mathbf{u}$ as angle of attack $\alpha$ and the output $y$ as the lift coefficient $C_L$. However, the theory is presented in a general framework for completeness.

**Impulse response in u** According to linear system theory, the impulse-response $y^{\delta}_{\mathbf{u}}(t)$ is the output corresponding to a delta function input, $\mathbf{u}(t) = \delta(t)$.[†] By linear superposition, it

---

[†] The output is the lift coefficient, $y = C_L$ and the input is either the angle-of-attack, $\mathbf{u} = \alpha$, or the vertical position, $\mathbf{u} = h$.

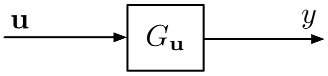


Figure 29: Transfer function $G_{\mathbf{u}}(s)$ between $\mathbf{u}$ and $y$.

is possible to construct the output response to an arbitrary input $\mathbf{u}(t)$ by convolution with the impulse-response $y^{\delta}_{\mathbf{u}}(t)$:

$$y(t) = \int_0^t y^{\delta}_{\mathbf{u}}(t-\tau)\mathbf{u}(\tau)d\tau = (y^{\delta}_{\mathbf{u}} * \mathbf{u})(t) \tag{53}$$

Taking the Laplace transform of both sides of Eq. (53) yields the convenient form

$$Y(s) = G_{\mathbf{u}}(s)U(s) \tag{54}$$

where $G_{\mathbf{u}}(s) = \mathcal{L}\left[y^{\delta}_{\mathbf{u}}(t)\right]$ is the transfer function relating input signals $U(s)$ to output signals $Y(s)$. This is shown schematically in Figure 29.

**Step response in $\mathbf{u}$ (impulse response in $\dot{\mathbf{u}}$)** For aerodynamic systems, an impulse response in $\mathbf{u}$ (either angle of attack or vertical position) might be nonphysical. Historically, it was practical to consider the step response $y^{S}_{\mathbf{u}}(t)$ (indicial response) to step inputs $\mathbf{u} = H(t)$, where $H(t)$ is the Heaviside step function. The step response in $\mathbf{u}$ is the same as the impulse response in $\dot{\mathbf{u}}$, so that $y^{S}_{\mathbf{u}} = y^{\delta}_{\dot{\mathbf{u}}}$. The superposition integral in Eq. (53) may be rewritten[‡] in terms of the step response $y^{S}(t)$:

$$y(t) = y^{S}(t)u(0) + \int_0^t y^{S}(t-\tau)\dot{u}(\tau)d\tau \tag{55}$$

For aerodynamic systems, there may be a nonzero steady-state value for the step-response in $\mathbf{u}$. The steady-state value, given by $G_{\mathbf{u}}(0) = y^{\delta}_{\dot{\mathbf{u}}}(\infty)$ corresponds to the quasi-steady lift slope $C_{L_u}$. It is convenient to split the step-response $y^{S}_{\mathbf{u}}(t) = y^{\delta}_{\dot{\mathbf{u}}}(t)$ into the steady-state and transient components:

$$y^{\delta}_{\dot{\mathbf{u}}}(t) = y^{\delta}_{\dot{\mathbf{u}}}(\infty) + {y^{\delta}_{\dot{\mathbf{u}}}}'(t) \tag{56}$$

so that Eq. (55) becomes

$$\begin{aligned} y(t) &= y^{\delta}_{\dot{\mathbf{u}}}(\infty)u(0) + {y^{\delta}_{\dot{\mathbf{u}}}}'(t)\mathbf{u}(0) + y^{\delta}_{\dot{\mathbf{u}}}(\infty)\int_0^t \dot{\mathbf{u}}(\tau)d\tau + \int_0^t {y^{\delta}_{\dot{\mathbf{u}}}}'(t-\tau)\dot{\mathbf{u}}(\tau)d\tau && (57) \\ &= {y^{\delta}_{\dot{\mathbf{u}}}}'(t)\mathbf{u}(0) + y^{\delta}_{\dot{\mathbf{u}}}(\infty)\mathbf{u}(t) + \int_0^t {y^{\delta}_{\dot{\mathbf{u}}}}'(t-\tau)\dot{\mathbf{u}}(\tau)d\tau && (58) \end{aligned}$$

Again, taking the Laplace transform of both sides of Eq. (58) yields

---

[‡] This is seen by substituting $y^{\delta}_{\dot{\mathbf{u}}}(t) = \int_0^t y^{\delta}_{\mathbf{u}}(t-s)ds$ into the following integral $\int_0^t y^{\delta}_{\dot{\mathbf{u}}}(t-\tau)\dot{u}(\tau)d\tau$ and integrating by parts.

Figure 30: Transfer function $G_{\dot{\mathbf{u}}}(s)$ between $\dot{\mathbf{u}}$ and $y$.

$$
\begin{aligned}
Y(s) &= G_{\dot{\mathbf{u}}}{}'(s)\mathbf{u}(0) + y^{\delta}_{\dot{\mathbf{u}}}(\infty)U(s) + G_{\dot{\mathbf{u}}}{}'(s)\mathcal{L}\left[\dot{\mathbf{u}}(t)\right] && (59)\\
&= \left[y^{\delta}_{\dot{\mathbf{u}}}(\infty) + sG_{\dot{\mathbf{u}}}{}'(s)\right]U(s) && (60)
\end{aligned}
$$

where $G_{\dot{\mathbf{u}}}(s) = \mathcal{L}\left[y^{\delta}_{\dot{\mathbf{u}}}(t)\right]$ is the transfer function from $\dot{\mathbf{u}}$ to $y$, as shown in Figure 30. Rearranging Eq. (60) yields the following:

$$
Y(s) = \underbrace{\left[\frac{G_{\mathbf{u}}(0)}{s} + G_{\dot{\mathbf{u}}}{}'(s)\right]}_{G_{\dot{\mathbf{u}}}(s)} sU(s) \tag{61}
$$

where $G_{\mathbf{u}}(0) = y^{\delta}_{\dot{\mathbf{u}}}(\infty)$. This transfer function corresponds to the following state-space representation:

$$
\begin{aligned}
\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix}_{k+1} &= \begin{bmatrix}A & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix}_{k} + \begin{bmatrix}B\\ \Delta t\end{bmatrix}\dot{\mathbf{u}}_k\\
y_k &= \begin{bmatrix}C & G_{\mathbf{u}}(0)\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \mathbf{u}\end{bmatrix}_{k} + D\dot{\mathbf{u}}_k
\end{aligned} \tag{62}
$$

where $(A, B, C, D)$ is a state-space realization of $G_{\dot{\mathbf{u}}}{}'(s)$.

**Impulse response in $\ddot{\mathbf{u}}$** It is observed that there are added-mass forces proportional to $\ddot{\mathbf{u}}$. These added-mass terms appear in $G_{\dot{\mathbf{u}}}{}'(s)$ as nonproper derivative terms, so that there does not exist a state-space realization $(A, B, C, D)$ for $G_{\dot{\mathbf{u}}}{}'(s)$. Therefore, we must choose $\ddot{\mathbf{u}}$ as the input variable for the system to be proper and representable in state-space form.

After subtracting off the steady-state lift corresponding to $G_{\mathbf{u}}(0) = C_{L_{\mathbf{u}}}$, it is possible to integrate the impulse response in $\dot{\mathbf{u}}$ to obtain the impulse response in $\ddot{\mathbf{u}}$:

$$
y^{\delta}_{\ddot{\mathbf{u}}}(t) = \int_0^t y^{\delta}_{\dot{\mathbf{u}}}(\tau)d\tau \tag{63}
$$

This amounts to multiplying the input by $s$ and the output by $1/s$. It is now possible to repeat the process above, subtracting off the steady-state lift $G_{\dot{\mathbf{u}}}{}'(0) = y^{\delta}_{\ddot{\mathbf{u}}}(\infty)$, now corresponding to $C_{L_{\dot{\mathbf{u}}}}$. The result is a transfer function $G_{\ddot{\mathbf{u}}}(s)$ from $\ddot{\mathbf{u}}$ to $y$ as shown in Figure 31:

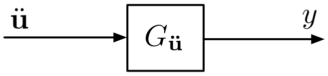


Figure 31: Transfer function $G_{\ddot{\mathbf{u}}}(s)$ between $\ddot{\mathbf{u}}$ and $y$.

$$G_{\ddot{\mathbf{u}}}(s) = \frac{G_{\mathbf{u}}(0)}{s^2} + \frac{G_{\dot{\mathbf{u}}}'(0)}{s} + G_{\ddot{\mathbf{u}}}'(s) \tag{64}$$

Let $(A, B, C, D)$ be a state-space realization of the transfer function $G_{\ddot{\mathbf{u}}}'(s)$. It is then possible to rewrite the time-domain response in the more convenient linear system framework

$$\begin{aligned} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_{k+1} &= \begin{bmatrix} A & 0 & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_{k} + \begin{bmatrix} B \\ 0 \\ \Delta t \end{bmatrix} \ddot{\mathbf{u}}_k \\ y_k &= \begin{bmatrix} C & G_{\mathbf{u}}(0) & G_{\dot{\mathbf{u}}}'(0) \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_{k} + D\,\ddot{\mathbf{u}}_k \end{aligned} \tag{65}$$

with initial condition $\begin{bmatrix} \mathbf{0} & \mathbf{u}(0) \end{bmatrix}^T$. It is worth noting that the $\mathbf{x}$ dynamics are decoupled from the last two states $\begin{bmatrix} \mathbf{u} & \dot{\mathbf{u}} \end{bmatrix}^T$. The $\mathbf{x}$ dynamics model the transient dynamics from the step-response, while the additional terms $G_{\mathbf{u}}(0)\mathbf{u}$ and $G_{\dot{\mathbf{u}}}'(0)\dot{\mathbf{u}}$ account for the quasi-steady lift associated with constant $\mathbf{u}$ and $\dot{\mathbf{u}}$. The next section details an approach to obtain an $r$-dimensional realization $(A_r, B_r, C_r, D_r)$ for $G_{\ddot{\mathbf{u}}}'(s)$ using the eigensystem realization algorithm.

### *6.1.2 Modeling fast dynamics using the Eigensystem Realization Algorithm*

One method of obtaining a state-space realization for the transfer function $G_{\ddot{\mathbf{u}}}'(s)$ is the eigensystem realization algorithm (ERA). As described in Section 5.1, this method involves taking snapshots $y_{\ddot{\mathbf{u}}}^{\delta\,\prime}(k\Delta t)$ from an impulse-response function $y_{\ddot{\mathbf{u}}}^{\delta\,\prime}(t) = \mathcal{L}^{-1}[G_{\ddot{\mathbf{u}}}'(s)]$ and it returns a reduced order, discrete-time, state-space model with time-step $\Delta t$.

Recall that we begin with a full-order model, and wish to construct a reduced-order model of the form

$$\left.\begin{aligned} \mathbf{x}(k+1) &= A\mathbf{x}(k) + B\mathbf{u}(k) \\ \mathbf{y}(k) &= C\mathbf{x}(k) + D\mathbf{u}(k) \end{aligned}\right\} \xrightarrow{\text{Reduction}} \begin{aligned} \mathbf{x}_r(k+1) &= A_r\mathbf{x}_r(k) + B_r\mathbf{u}(k) \\ \mathbf{y}(k) &= C_r\mathbf{x}_r(k) + D_r\mathbf{u}(k) \end{aligned} \tag{66}$$

Note that it is not necessary to know the full-order model explicitly: we need not know the matrices $(A, B, C, D)$, but we do need to be able to obtain impulse response data from the system. This data may be obtained either from simulations or from experiments.

The steps involved are described in more detail in Section 5.1, but here we give a brief summary:

1. The first term in the impulse response is the $D$ matrix, or feed-through term. If the impulse is in $\mathbf{\ddot{u}}$, then the $D$ matrix corresponds to the term $C_{L_{\mathbf{\ddot{u}}}}$ which accounts for added mass proportional to $\mathbf{\ddot{u}}$.

2. Gather the next $(m_c + m_o) + 2$ outputs from the impulse-response simulation, where $m_c$ and $m_o$ are integers representing the number of snapshots for controllability and observability. The outputs $\mathbf{y}(k) = CA^{k-1}B$ are called Markov parameters, and are synthesized into a generalized Hankel matrices:

$$H = \begin{bmatrix} CB & CAB & \cdots & CA^{m_c}B \\ CAB & CA^2B & \cdots & CA^{m_c+1}B \\ \vdots & \vdots & \ddots & \vdots \\ CA^{m_o}B & CA^{m_o+1}B & \cdots & CA^{m_c+m_o}B \end{bmatrix} \tag{67}$$

$$H' = \begin{bmatrix} CAB & CA^2B & \cdots & CA^{m_c+1}B \\ CA^2B & CA^3B & \cdots & CA^{m_c+2}B \\ \vdots & \vdots & \ddots & \vdots \\ CA^{m_o+1}B & CA^{m_o+2}B & \cdots & CA^{m_c+m_o+1}B \end{bmatrix} \tag{68}$$

3. Compute the singular value decomposition of $H$:

$$H = U\Sigma V^* = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} = U_1 \Sigma_1 V_1^* \tag{69}$$

4. Finally, let $\Sigma_r$ be the first $r \times r$ block of $\Sigma_1$, $U_r, V_r$ the first $r$ columns of $U_1, V_1$, and the reduced order $A_r, B_r, C_r$ are given as follows:

$$A_r = \Sigma_r^{-1/2} U_r^* H' V_r \Sigma_r^{-1/2} \tag{70}$$

$$B_r = \text{ first } p \text{ columns of } \Sigma_r^{1/2} V_1^* \tag{71}$$

$$C_r = \text{ first } q \text{ rows of } U_r \Sigma_r^{1/2} \tag{72}$$

### *6.1.3 Choice of step function*

For a number of reasons, an actual step response is non-physical. First, it is impossible to command in experiments or simulations, because it would correspond to a body instantaneously dematerializing and then rematerializing it in another location. An alternative is to use a smoothed step maneuver and approach the limit as the maneuver becomes very rapid. As the maneuver becomes increasingly rapid, the added-mass forces begin to dominate; in fact, a good rule of thumb is to choose a maneuver rapid enough that the lift response for the duration of the maneuver is dominated by added-mass.

In our simulations, the duration of the maneuver is $T = .01$ convective time units, and the amplitude is either $A = 1^\circ \approx .01745$ rad in the case of pitching or $A = .01745$ chord lengths in the case of vertical position. This is sufficiently rapid for the added-mass forces to dominate during the maneuver.

The linear ramp maneuver was first introduced as a canonical pitching maneuver to compare and study various experiments, simulations and models[46]. The equations for $u$ and $\dot{u}$ are given in Eqs. (73–75), and the maneuver is shown in Figure 32:

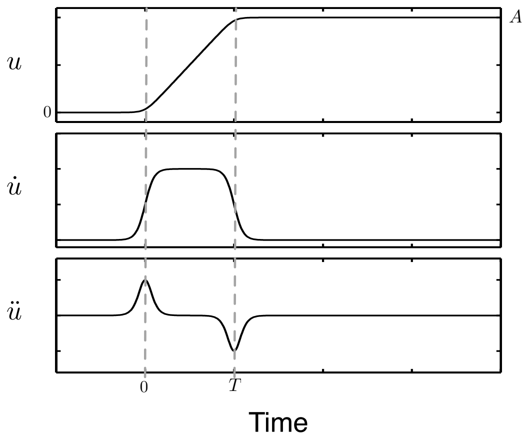


Figure 32: Linear ramp maneuver in $u$.

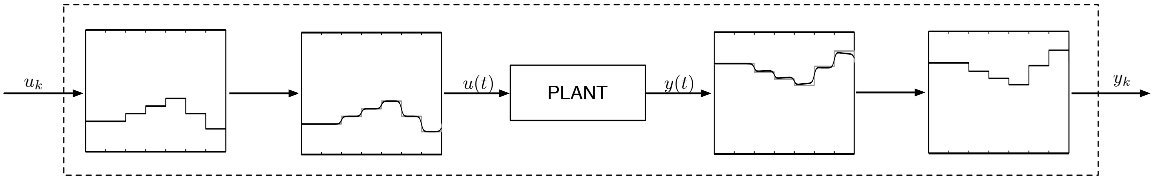


Figure 33: Diagram illustrating the discrete-time system arising from pitch-ramp step.

$$G(t) = \log \left[ \frac{\cosh(at)}{\cosh(a(t-T))} \right] \tag{73}$$

$$u(t) = A \frac{G(t)}{\max G(t)} \tag{74}$$

$$\dot{u}(t) = A \frac{\tanh(at) - \tanh(a(t-T))}{\max G(t)} \tag{75}$$

The start of the maneuver is $t = 0$ and the duration of the ramp-up is $T$. The parameter $a$ effects how gradual or abrupt the ramp acceleration is. By choosing $a$ large, such as $a = 1000$, it is possible to obtain a maneuver where the $\ddot{u}$ acceleration effects are localized near time $t = 0$ and $t = T$, and the velocity $\dot{u}$ is constant throughout much of the maneuver. This results in an approximately piecewise linear ramp-up, where the transition at $t = 0$ and $t = T$ are smoothed.

The linear ramp maneuver has a number of benefits that make it a natural choice for our smoothed step maneuver. First, the boxy profile of the velocity $\dot{u}$ resembles the shape of a discrete-time impulse in $\dot{u}$ with time step $T$. This makes it possible to run simulations that fully resolve the maneuver in time, and then sample starting at the middle of the maneuver with sampling time $T$ and obtain a good approximation of the discrete time system. This is shown schematically in Figure 33.

### *6.1.4 Method summary*

The overall method is summarized as follows:

1. Run an impulse-response simulation in $\dot{\mathbf{u}}$, which is a step response in $\mathbf{u}$. This response exhibits dominant derivative ($\ddot{\mathbf{u}}$) term during the smoothed step maneuver. This indicates that the input should be $\ddot{\mathbf{u}}$ in order to have a proper system, which will be addressed in step 3.
2. Subtract off $C_{L_{\mathbf{u}}}$ term, corresponding to steady state lift after step in $\mathbf{u}$:
$$C_L' = C_L - C_{L_\alpha} \tag{76}$$
3. Integrate to get impulse in $\ddot{u}$ (less $C_{L_{\mathbf{u}}}$ contribution)
$$\tilde{C}_L = \int C_L' d\tau \tag{77}$$
4. Subtract off $C_{L_{\dot{\mathbf{u}}}}$ term, corresponding to steady state lift due to constant $\dot{\mathbf{u}}$,
$$\tilde{C}_L' = \tilde{C}_L - C_{L_{\dot{\mathbf{u}}}} \tag{78}$$
5. Sample smoothed step input and output functions starting at the middle of the smoothed impulse with sampling time $\Delta t$.
6. Obtain Markov parameters
```
H = markov(Y,U,M)
```
7. First term, $H_1$ is the $D$ matrix corresponding to $\ddot{u}$ feed-through term, $C_{L_{\ddot{\mathbf{u}}}}$.
8. The rest of the Markov parameters $H_j$ go into ERA code
```
[Ar,Br,Cr,HSVs] = ERA(H,m,n,r)
```
9. Synthesize into final form of ROM:

$$\begin{aligned}
\begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_{k+1} &= \begin{bmatrix} A_r & 0 & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_k + \begin{bmatrix} B_r \\ 0 \\ \Delta t \end{bmatrix} \ddot{\mathbf{u}}_k \\
C_L(k\Delta t) &= \begin{bmatrix} C_r & C_{L_{\mathbf{u}}} & C_{L_{\dot{\mathbf{u}}}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \\ \dot{\mathbf{u}} \end{bmatrix}_k + C_{L_{\ddot{\mathbf{u}}}} \ddot{\mathbf{u}}_k
\end{aligned} \tag{79}$$

The preceding algorithm is sufficiently general to produce reduced order models for an airfoil pitching about various points along the chord, plunging vertically, or pitching and plunging. However, because of the exceptions mentioned in the introduction to this section, the case of middle-chord pitching and vertical plunging require minor special attention.

- **Exception 1. Mid-chord pitching:** Because there are no added-mass forces proportional to $\ddot{\alpha}$ in the degenerate case of pitching about the middle chord, we must manually set $C_{L_{\ddot{\alpha}}} = 0$. Although the value obtained in the algorithm is quite small, any non-zero value will result in incorrect behavior at high frequencies due to the relative degree.
- **Exception 2. Vertical plunging:** Given a step-up in vertical position, there is no corresponding steady-state lift, and so we manually set $C_{L_h} = 0$. This guarantees the correct behavior at low frequencies.

### *6.1.5 Possible variations for plunging and pitching about mid-chord*

As mentioned above, middle-chord pitching and vertical plunging are exceptions and require special attention. The reduced order model, Eq. (51) is sufficiently general to handle these exceptions. In the case of pitching about the middle chord, we manually set $C_{L_{\ddot{\alpha}}} = 0$ and in the case of vertical plunging we manually set $C_{L_h} = 0$.

Alternatively, it is possible to eliminate steps 3–4 above for mid-chord pitching, resulting in the model:

$$\begin{aligned} \begin{bmatrix} \mathbf{x} \\ \alpha \end{bmatrix}_{k+1} &= \begin{bmatrix} A_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \alpha \end{bmatrix}_k + \begin{bmatrix} B_r \\ \Delta t \end{bmatrix} \dot{\alpha}_k \\ C_L(k\Delta t) &= \begin{bmatrix} C_r & C_{L_\alpha} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \alpha \end{bmatrix}_k + C_{L_{\dot{\alpha}}}\ \dot{\alpha}_k \end{aligned} \tag{80}$$

For plunging, one may use a step in vertical velocity, modifying steps 1–4 and results in the model:

$$\begin{aligned} \begin{bmatrix} \mathbf{x} \\ \dot{h} \end{bmatrix}_{k+1} &= \begin{bmatrix} A_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{h} \end{bmatrix}_k + \begin{bmatrix} B_r \\ \Delta t \end{bmatrix} \ddot{h}_k \\ C_L(k\Delta t) &= \begin{bmatrix} C_r & C_{L_{\dot{h}}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \dot{h} \end{bmatrix}_k + C_{L_{\ddot{h}}}\ \ddot{h}_k \end{aligned} \tag{81}$$

It is interesting to note that in this form, with the input for pitching about the mid-chord now $\dot{\alpha}$, these models should be equivalent by symmetry. In other words, a plunging motion $\dot{h}(t)$ translates directly into an angle-of-attack motion $\alpha_{\text{eff}}(t)$ by a change of variables.

### *6.1.6 Multiple Input Multiple Output (MIMO) generalization*

An important result of this paper is that the above algorithm generalizes to MIMO representations. Lift coefficient dynamics in angle of attack and vertical position are easily representable in a combined model.

First, treat each step response separately, first subtracting off $C_{L_u}$, then integrating, subtracting off $C_{L_{\dot{u}}}$, and finally passing through a Markov parameter identification algorithm to identify $C_{L_{\ddot{u}}}$ and the transient impulse response in $\ddot{u}$. Until now, we have exactly repeated steps 1–7 individually for each step response. Now, the ERA algorithm, step 8, requires a deviation. The Markov parameters $CA^kB$ are no longer coefficients, but matrices of size

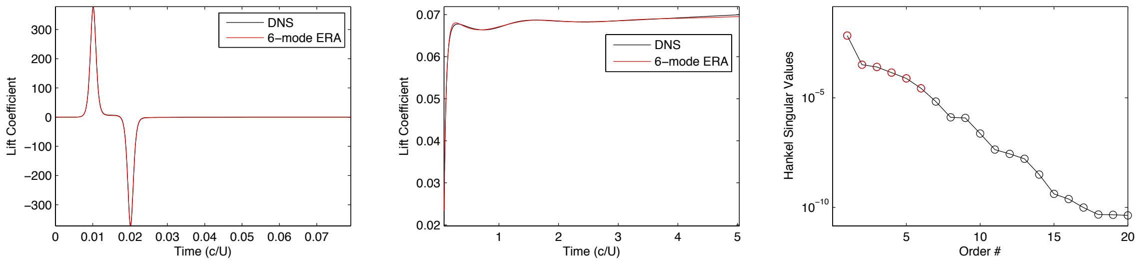


Figure 34: Step-responses (left, middle) and Hankel singular values (right) for 1° pitch-up about the quarter chord. DNS (black) is compared with a 6-mode ERA model (red).

input × output. In other words, we stack the two transient impulse responses to obtain a multidimensional impulse response. We pass this through the ERA algorithm to obtain a representation $(A_r, B_r, C_r)$ for the remainder of the dynamics. The full expression for the model is:

$$\begin{gathered}
\begin{bmatrix} \mathbf{x} \\ \alpha \\ h \\ \dot{\alpha} \\ \dot{h} \end{bmatrix}_{k+1} = \begin{bmatrix} A_r & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \alpha \\ h \\ \dot{\alpha} \\ \dot{h} \end{bmatrix}_k + \begin{bmatrix} B_{r\alpha} & B_{rh} \\ 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{h} \end{bmatrix}_k \\
C_L(k\Delta t) = \begin{bmatrix} C_r & C_{L_\alpha} & C_{L_h} & C_{L_{\dot{\alpha}}} & C_{L_{\dot{h}}} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \alpha \\ h \\ \dot{\alpha} \\ \dot{h} \end{bmatrix}_k + \begin{bmatrix} C_{L_{\ddot{\alpha}}} & C_{L_{\ddot{h}}} \end{bmatrix} \begin{bmatrix} \ddot{\alpha} \\ \ddot{h} \end{bmatrix}_k
\end{gathered} \tag{82}$$

## 6.2 Results: small-amplitude maneuvers

In this section, we present the results of the procedure described in section 6.1, applied to small-amplitude pitching and plunging maneuvers. As with previous examples in this report, we study a flat plate at Re = 100.

### *6.2.1 Pitching*

Here, we present results for a flat plate pitching about the quarter chord. The results are similar for airfoils pitched about the leading edge or the middle chord, with the exception that for middle-chord pitching, there are no added mass effects on the lift.

Figure 34 shows a comparison of a small-amplitude step for the full simulation (DNS), compared with a 6-mode model obtained by the Eigensystem Realization Algorithm (adjoint-free Balanced POD, from Section 5). In the leftmost plot, the responses due to $C_{L\ddot{a}lpha}$, $C_{L\dot{\alpha}}$, and $C_{L\alpha}$ have been subtracted off, as described in Section 6.1.4, so all that remains is the term $G(s)$ in Figure 28. The middle plot shows the full step response $C_L(t)$, with these terms included. Note that in both cases, the agreement of the 6-mode model is excellent.

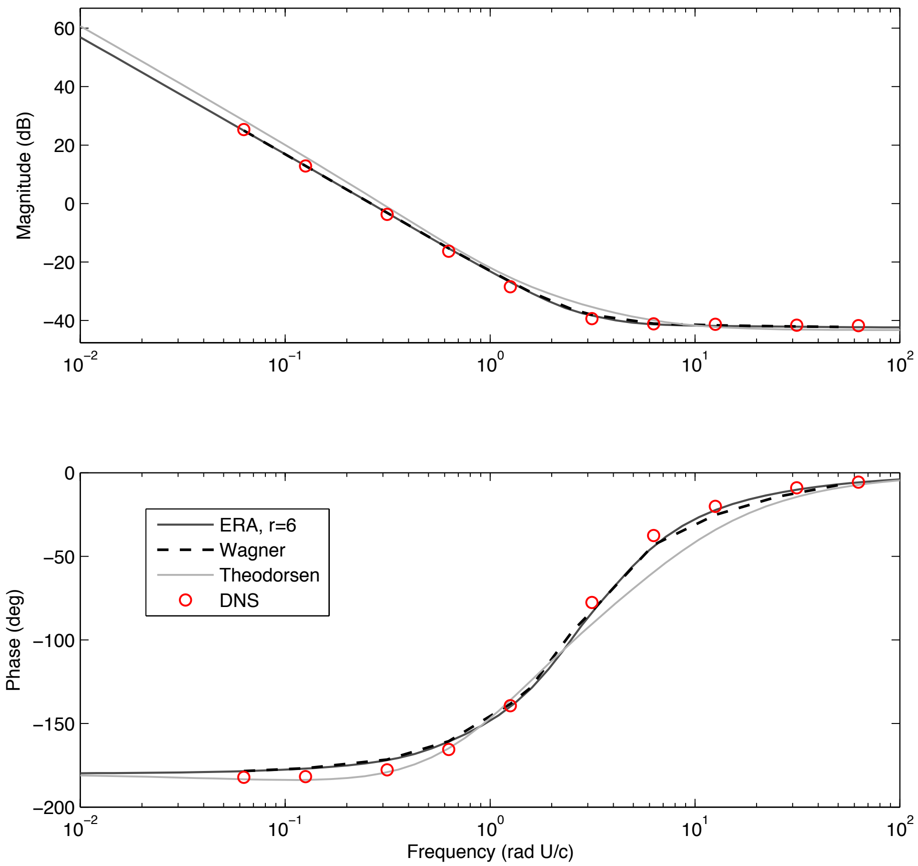


Figure 35: Frequency response of Wagner's model and ERA models for pitching at quarter chord.

The rapid decay of the Hankel singular values in the rightmost plot shows that the response may be captured effectively by a low-dimensional model.

Figure 35 shows a comparison of the Bode plots (frequency response) for the full simulation (DNS), compared with the full Wagner model, and the reduced-order ERA model. Theodorsen's model is also shown, for comparison. Note that in order to compute the Bode plot for Wagner's model, one must compute a convolution integral at each frequency, while for the reduced-order model, one has a simple 6th-order transfer function, suitable for control design. Agreement of this reduced-order model with Wagner is nearly perfect, and agreement with DNS is excellent as well.

### *6.2.2 Plunging*

Here, we consider a maneuver as in the previous section, but for a flat plate undergoing pure plunge.

Figure 36 shows the step responses for the plunge-up maneuver, comparing the full DNS with a 7th-order ERA model. As in Figure 34, the middle plot is the actual lift, while the leftmost plot shows the lift with the terms due to $C_{L\ddot{a}lpha}$, $C_{L\dot{\alpha}}$, and $C_{L\alpha}$ subtracted off, as

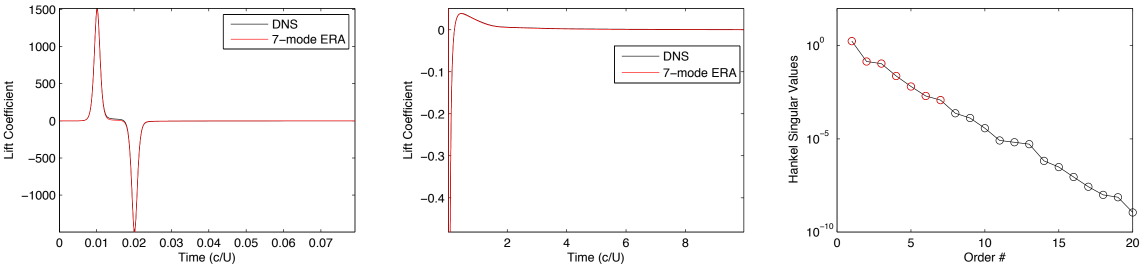


Figure 36: Step-responses (left, middle) and Hankel singular values (right) for plunge-up maneuver. DNS (black) is compared with a 7-mode ERA model (red).

described in Section 6.1.4. The step responses are visually identical.

Figure 37 shows the corresponding Bode plots for these cases, including a comparison with Wagner and Theodorsen's models. All of the models match the DNS well, except for a 15–20° discrepancy in phase for very low frequencies. As before, the ERA model agrees nearly perfectly with Wagner's full indicial response, but is much more convenient to use for control design.

## 6.3 Results: large-amplitude maneuvers

Finally, we apply these models to a canonical unsteady maneuver considered in [46]. In this maneuver, the airfoil pitches up to a large angle of attack of 45°, holds for one convective time unit, and then pitches back down to zero angle of attack. Wagner's models, and the approximations to them given in the previous section, are designed for small angles of attack, and are not meant to capture such large excursions. However, we compare the model's predictions here to better understand the significance of nonlinearities on the unsteady aerodynamics.

Figure 39 shows the lift when the airfoil is pitched about the quarter chord, at Reynolds number 100. The lift from the full DNS is compared with the predictions of Wagner's model, and our 6-state low-order model from Section 6.2.1. The agreement between our model and Wagner is nearly perfect, as expected. The agreement with DNS is good up to about 2 convective time units, at which time the angle of attack exceeds 20°. The large peaks in lift are due to the added mass response, and the models are very effective at capturing these. The agreement is poor between about 2 and 4 convective time units, but then matches quite well as the airfoil pitches back down to zero angle of attack.

One of the reasons for the discrepancy for large angles of attack is that in the linearized models, the force is normal to the airfoil's surface, while the lift is defined as the force normal to the freestream velocity. Thus, a simple $\cos(\alpha)$ correction can be applied to correct for this difference. When this correction is applied (shown in the right plot in Figure 39, agreement with DNS is good for the entire pitch-up maneuver, but then begins to disgree during the hold and pitch-down portions.

Figure 40 shows the response analogous to that shown in Figure 39, but for the airfoil pitched about the middle chord. Here, there are no added mass forces from the unsteady pitching, so

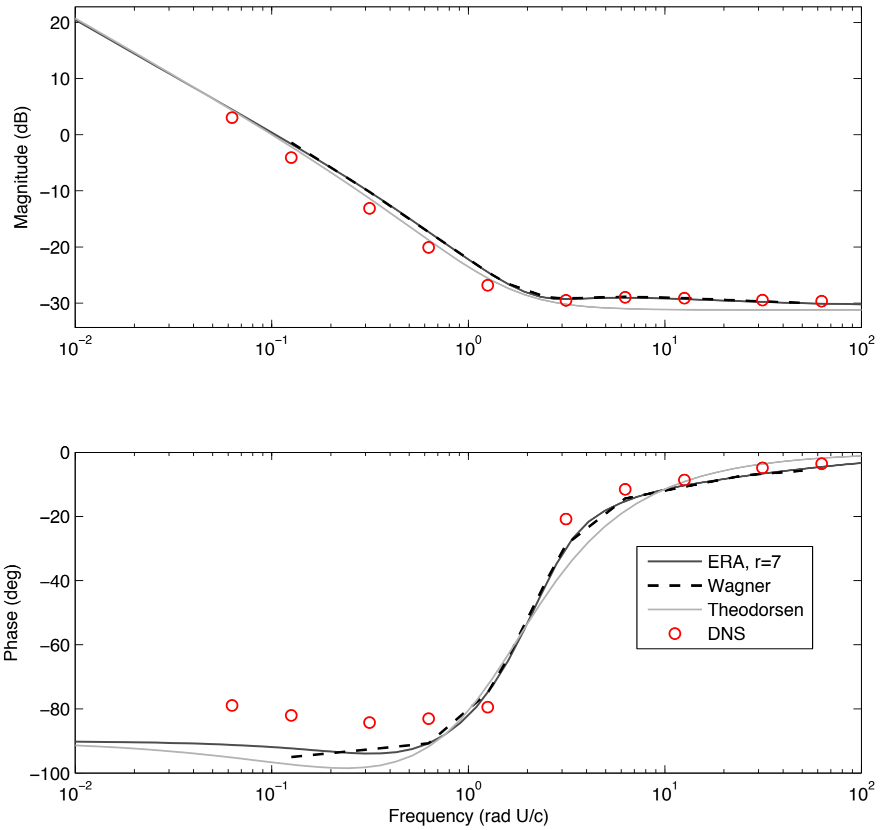


Figure 37: Frequency response of Wagner's model and ERA models for plunging.

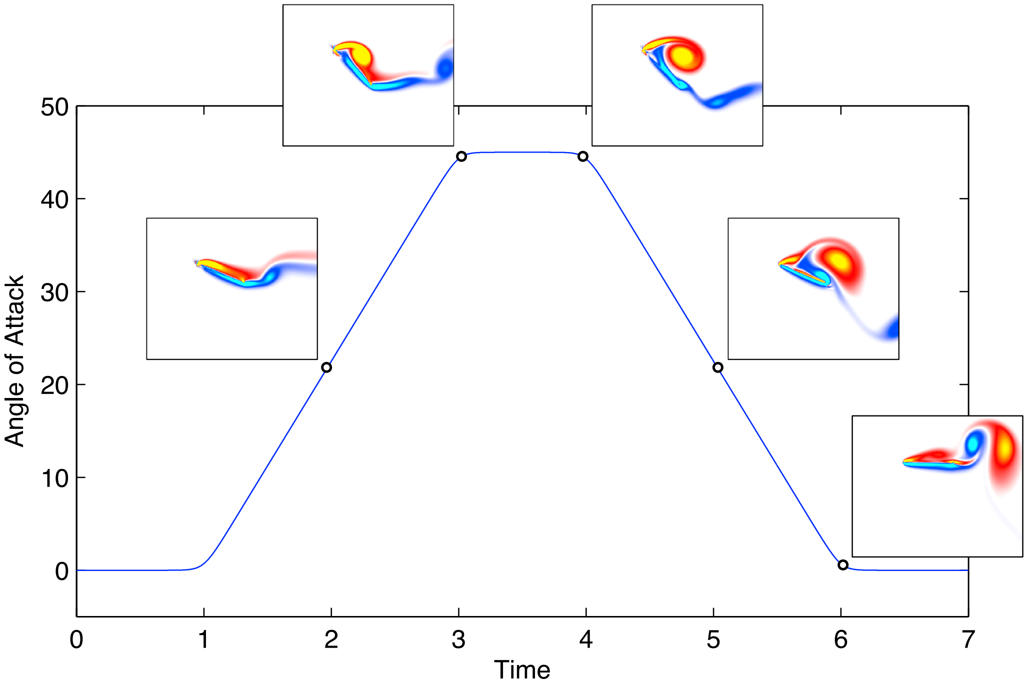


Figure 38: Canonical maneuver with pitch-up, hold, and pitch-down phases, showing the instantaneous vorticity field at various points along the maneuver, at Re = 100.

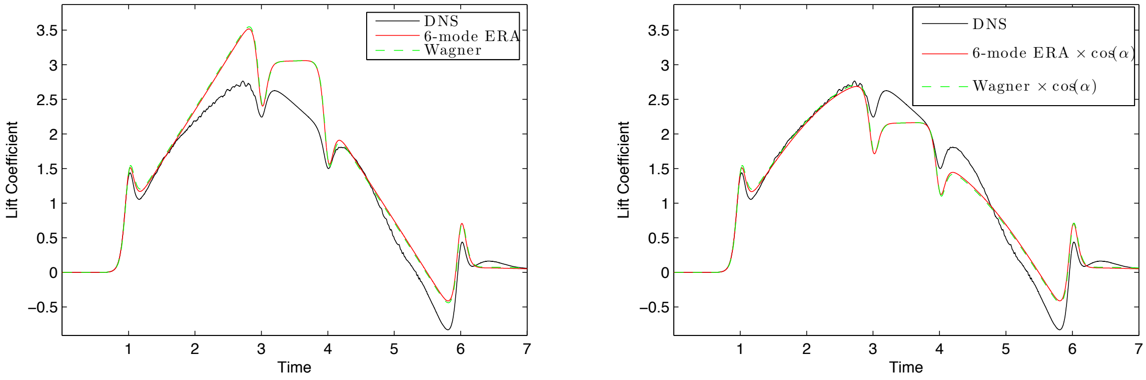


Figure 39: Comparison of lift model for canonical pitch-ramp-hold maneuver about *quarter chord*: *Left:* linearized model prediction; *Right*: model prediction with cos($\alpha$) correction.

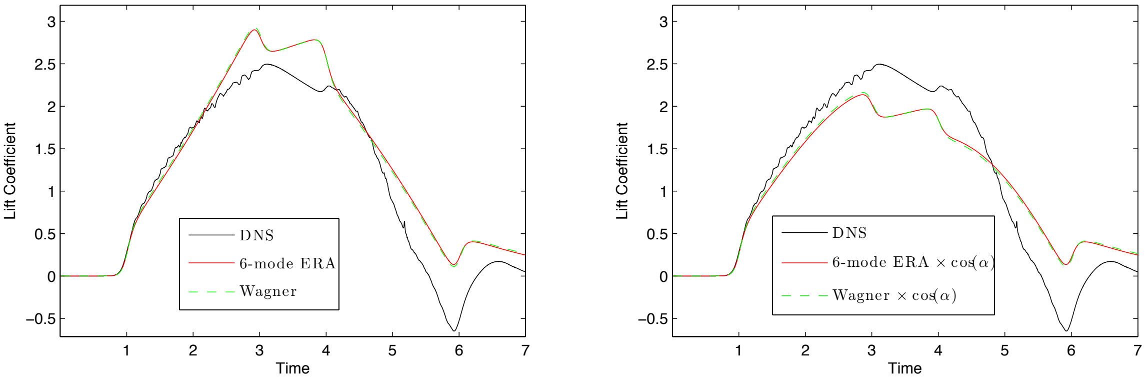


Figure 40: Comparison of lift model for canonical pitch-ramp-hold maneuver about *middle chord*: *Left:* linearized model prediction; *Right*: model prediction with $\cos(\alpha)$ correction.

the response looks significantly smoother. As before, the linearized model prediction is good when the angle of attack is small, but deviates when the angle becomes large. The $\cos(\alpha)$ correction improves the response during the pitch-up portion, but degrades the response during the pitch-down portion.

## 6.4 Conclusions

The goal of this section is to present a procedure for constructing unsteady aerodynamic models that are suitable for control design. The models are based on Wagner's indicial response [63], and agree with Wagner's models to an arbitrarily high degree of accuracy. While Wagner's models represent the lift as a convolution integral that is cumbersome to compute, and not suitable for control synthesis, our procedure produces state-space models that may be used directly for control design. Furthermore, our models are formulated in a way that directly builds upon standard approaches incorporating classical "stability derivatives" $C_{L\alpha}$, $C_{L\dot{\alpha}}$, $C_{L\ddot{\alpha}}$ (as in [57]), as shown in Figure 28.

Our models predict the unsteady response very well for small-amplitude maneuvers, as shown by the frequency responses in Figures 35 and 37. For large amplitude maneuvers, such as the canonical maneuver shown in Figure 38, agreement is poor when the angle of attack becomes larger than about 20°. More work is needed to extend these models to the nonlinear regime.

# Acknowledgment/Disclaimer

This work was sponsored by the Air Force Office of Scientific Research, USAF, under grant/contract number FA9550-07-1-0127. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

## References

[1] S. Ahuja and C. W. Rowley. Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators. *J. Fluid Mech.*, 645:447–478, Feb. 2010.

[2] S. Ahuja, C. W. Rowley, I. G. Kevrekidis, M. Wei, T. Colonius, and G. Tadmor. Low-dimensional models for control of leading-edge vortices: Equilibria and linearized models. AIAA Paper 2007-709, 45th AIAA Aerospace Sciences Meeting and Exhibit, Jan. 2007.

[3] J. C. Akers and D. S. Bernstein. ARMARKOV least-squares identification. In *Proceedings of the ACC*, pages 186–190, Albuquerque, NM, USA, 1997.

[4] D. A. Allwine, J. A. Strahler, D. A. Lawrence, J. E. Jenkins, and J. H. Myatt. Nonlinear modeling of unsteady aerodynamics at high angle of attack. AIAA Paper 2004-5275, 2004.

[5] S. Bagheri, L. Brandt, and D. S. Henningson. Input–output analysis, model reduction and control of the flat-plate boundary layer. *J. Fluid Mech.*, 620:263–298, 2009.

[6] B. Bamieh and J. B. Pearson. The H2 problem for sampled-data systems. *Sys. Control Lett.*, 19:1–12, 1992.

[7] S. Brunton and C. W. Rowley. Modeling the unsteady aerodynamic forces on small-scale wings. AIAA Paper 2009-1127, 47th Aerospace Sciences Meeting, Jan. 2009.

[8] S. L. Brunton and C. W. Rowley. Fast computation of finite-time Lyapunov exponent fields for unsteady flows. *Chaos*, 20:017510, 2010.

[9] S. L. Brunton, C. W. Rowley, K. Taira, T. Colonius, J. Collins, and D. R. Williams. Unsteady aerodynamic forces on small-scale wings: experiments, simulations and models. AIAA Paper 2008-520, 46th AIAA Aerospace Sciences Meeting and Exhibit, Jan. 2008.

[10] R. H. Cabell, M. A. Kegerise, D. E. Cox, and G. P. Gibbs. Experimental feedback control of flow-induced cavity tones. *AIAA J.*, 44(8):1807–1815, 2006.

[11] L. N. Cattafesta, III, S. Garg, M. Choudhari, and F. Li. Active control of flow-induced cavity resonance. AIAA Paper 97-1804, June 1997.

[12] Y. Chahlaoui and P. Van Dooren. Model reduction of time-varying systems. In *Dimension reduction of large-scale systems*, pages 131–148. Springer-Verlag, 2006.

[13] T. Colonius and K. Taira. A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Comp. Meth. Appl. Mech. Eng.*, 197(25-28):2131–46, 2008.

[14] M. Farhood, C. L. Beck, and G. E. Dullerud. Model reduction of periodic systems: a lifting approach. *Automatica*, 41:1085–1090, 2005.

[15] E. Franco, D. N. Pekarek, J. Peng, and J. O. Dabiri. Geometry of unsteady fluid transport during fluid-structure interactions. *J. Fluid Mech.*, 589:125–145, 2007.

[16] C. Garth, F. Gerhardt, X. Trichoche, and H. Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visulization and Computer Graphics*, 13(6):1464–1471, 2007.

[17] W. Gawronski. *Balanced control of flexible structures.* Springer, 1996.

[18] M. Goman and A. Khrabrov. State-space representation of aerodynamic characteristics of an aircraft at high angles of attack. *J. Aircraft*, 31(5):1109–1115, 1994.

[19] M. Green. *Analysis of bio-inspired propulsors.* PhD thesis, Princeton University, 2009.

[20] M. A. Green, C. W. Rowley, and G. Haller. Detection of Lagrangian coherent structures in 3D turbulence. *J. Fluid Mech.*, 572:111–120, Feb. 2007.

[21] J. Guckenheimer and P. J. Holmes. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 1983.

[22] G. Haller. Distinguished material surfaces and coherent structures in 3d fluid flows. *Physica D*, 149:248–277, 2001.

[23] G. Haller. Lagrangian coherent structures from approximate velocity data. *Phys. Fluids*, 14(6):1851–1861, 2002.

[24] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two dimensional turbulence. *Physica D*, 147:352–370, 2000.

[25] P. Hartman. *Ordoinary Differential Equations.* John Wiley and Sons, 1964.

[26] M. Ilak and C. W. Rowley. Modeling of transitional channel flow using balanced proper orthogonal decomposition. *Phys. Fluids*, 20(034103), 2008.

[27] M. Ilak and C. W. Rowley. Modeling of transitional channel flow using balanced proper orthogonal decomposition. *Phys. Fluids*, 20:034103, March 2008.

[28] J.-N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *J. Guid. Contr. Dyn.*, 8(5):620–627, 1985.

[29] J.-N. Juang and M. Q. Phan. *Identification and control of mechanical systems.* Cambridge University Press, 2001.

[30] M. M. Koochesfahani. Vortical patterns in the wake of an oscillating airfoil. *AIAA J.*, 27:1200–1205, 1989.

[31] H. Küsner. Zusammenfassender bericht über den instationären auftrieb von flügeln. *Luftfahrtforschung*, 13(12):410–424, 1935.

[32] S. Lall, J. E. Marsden, and S. Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *Int. J. Robust Nonlinear Control*, 12:519–535, 2002.

[33] J. G. Leishman. *Principles of Helicopter Aerodynamics.* Cambridge University Press, 2 edition, 2006.

[34] F. Lekien. *Time-dependent dynamical systems and geophysical flows.* PhD thesis, California Institute of Technology, 2003.

[35] F. Lekien, C. Coulliette, A. J. Mariano, E. H. Ryan, L. K. Shay, G. Haller, and J. E. Marsden. Pollution release tied to invariant manifolds: a case study for the coast of florida. *Phys. D*, 210:1–20, 2005.

[36] F. Lekien, S. C. Shadden, and J. E. Marsden. Lagrangian coherent structures in $n$-dimensional systems. *Journal of Mathematical Physics*, 48:065404, 2007.

[37] R. K. Lim, M. Q. Phan, and R. W. Longman. State-space system identification with identified Hankel matrix. Mechanical and Aerospace Engineering Tech. Report 3045, Princeton University, 1998.

[38] D. Lipinski, B. Cardwell, and K. Mohseni. A lagrangian analysis of a two-dimensional airfoil with vortex shedding. *jphysa*, 41:344011, 2008.

[39] Z. Ma, S. Ahuja, and C. W. Rowley. Reduced order models for control of fluids using the eigensystem realization algorithm. *Theor. Comput. Fluid Dyn.*, (accepted), 2010.

[40] Z. Ma and C. W. Rowley. Low-dimensional linearized models for systems with periodic orbits, with application to the Ginzburg-Landau equation. AIAA Paper 2008-4196, 4th Flow Control Conference, June 2008.

[41] J. Magill, M. Bachmann, G. Rixon, and K. McManus. Dynamic stall control using a model-based observer. *J. Aircraft*, 40(2):355–362, 2003.

[42] R. A. Meyer and C. S. Burrus. A unified analysis of multirate and periodically time-varying digital filters. *IEEE Trans. Circuits Syst.*, 22:162–168, 1975.

[43] B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Contr.*, 26(1):17–32, Feb. 1981.

[44] J. H. Myatt and G. A. Addington. Rotary-rate effects on critical-state location for a 65-degree delta wing. AIAA Paper 99-4098, 1999.

[45] B. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *J. Fluid Mech.*, 497:335–363, 2003.

[46] M. V. OL, A. Altman, J. D. Eldredge, D. J. Garmann, and Y. Lian. Résumé of the aiaa fdtc low reynolds number discussion group's canaonical cases. AIAA Paper 2010-1085, 48th AIAA Aerospace Sciences Meeting, Jan. 2010.

[47] K. Padberg, T. Hauff, F. Jenko, and O. Junge. Lagrangian structures and transport in turbulent magnetized plasmas. *New J. Phys.*, 9:400, 2007.

[48] J. Peng and J. O. Dabiri. The 'upstream wake' of swimming and flying animals and its correlation with propulsive efficiency. *J. Exp. Biol.*, 211:2669–2677, 2008.

[49] C. W. Rowley. Model reduction for fluids using balanced proper orthogonal decomposition. *Int. J. Bifurcation Chaos*, 15(3):997–1013, Mar. 2005.

[50] F. Sadlo and R. Peikert. Efficient visualization of lagrangian coherent structures by filtered amr ridge extraction. *IEEE Transactions on Visulization and Computer Graphics*, 13(6), 2007.

[51] H. Salman, J. S. Hesthaven, T. Warburton, and G. Haller. Predicting transport by lagrangian coherent structures with a high-order method. *Theor. Comput. Fluid Dyn.*, 21:39–58, 2007.

[52] S. Shadden, K. Katija, M. Rosenfeld, J. Marsden, and J. O. Dabiri. Transport and stirring induced by vortex formation. *J. Fluid Mech.*, 593:315–331, 2007.

[53] S. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Phys. D*, 212:271–304, 2005.

[54] K. Shi, H.-P. Seidel, H. Theisel, T. Weinkauf, and H.-C. Hege. Visualizing transport structures of time-dependent flow fields. *IEEE Computer Graphics and Applications*, pages 24–36, September/October 2008.

[55] S. Shokoohi, L. M. Silverman, and P. Van Dooren. Linear time-variable systems: Balancing and model reduction. *IEEE Trans. Automat. Contr.*, 28:810–822, 1983.

[56] L. Sirovich. Turbulence and the dynamics of coherent structures, parts I–III. *Q. Appl. Math.*, XLV(3):561–590, Oct. 1987.

[57] R. F. Stengel. *Flight Dynamics.* Princeton University Press, 2004.

[58] T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. Technical Report 496, NACA, 1935.

[59] M. Tobak and L. B. Schiff. On the formulation of the aerodynamic characteristics in aircraft dynamics. Technical Report R-456, NASA, 1976.

[60] A. Varga. Balanced truncation model reduction of periodic systems. In *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 2379–2384, 2000.

[61] E. I. Verriest and T. Kailath. On generalized balanced realizations. *IEEE Trans. Automat. Contr.*, 28:833–844, 1983.

[62] T. von Karman and W. R. Sears. Airfoil theory for non-uniform motion. *J. Aeronaut. Sci.*, 5(10):379–390, 1938.

[63] H. Wagner. über die entstehung des dynamischen auftriebes von tragflügeln. *Zeitschrift für Angewandte Mathematic und Mechanik*, 5(1):17, 1925.

[64] Y. Wang, G. Haller, A. Banaszuk, and G. Tadmor. Closed-loop lagrangian separation control in a bluff body shear flow model. *Phys. Fluids*, 15(8):2251–66, 2003.

[65] M. M. Wilson, J. Peng, J. O. Dabiri, and J. D. Eldgredge. Lagrangian coherent structures in low Reynolds number swimming. *J. Phys.: Condens. Matter*, 21:204105, 2009.

[66] K. Zhou, G. Salomon, and E. Wu. Balanced realization and model reduction for unstable systems. *Int. J. Robust and Nonlin. Contr.*, 9(3):183–198, Mar. 1999.

## Personnel Supported During Duration of Grant

| | |
|---|---|
| Steven Brunton | Graduate student, Princeton University |
| Zhanhua Ma | Graduate Student, Princeton University |
| Milos Ilak | Graduate student, Princeton University |
| Juan Melli | Graduate student, Princeton University |
| Brandt Belson | Graduate student, Princeton University |
| Clarence W. Rowley | Associate Professor, Princeton University |

## Publications

- S. Brunton and C. W. Rowley. Modeling the unsteady aerodynamic forces on small-scale wings. AIAA Paper 2009-1127, 47th Aerospace Sciences Meeting, Jan. 2009.
- S. L. Brunton and C. W. Rowley. Fast computation of finite-time lyapunov exponent fields for unsteady flows. 20:017510, 2010.
- S. L. Brunton and C. W. Rowley. Unsteady aerodynamic models for agile flight at low reynolds numbers. AIAA Paper 2010-552, 48th AIAA Aerospace Sciences Meeting, Jan. 2010.
- S. L. Brunton, C. W. Rowley, K. Taira, T. Colonius, J. Collins, and D. R. Williams. Unsteady aerodynamic forces on small-scale wings: experiments, simulations and models. AIAA Paper 2008-520, 46th AIAA Aerospace Sciences Meeting and Exhibit, Jan. 2008.
- Z. Ma, S. Ahuja, and C. W. Rowley. Reduced order models for control of fluids using the eigensystem realization algorithm. *Theor. Comput. Fluid Dyn.*, (available online), doi:10.1007/s00162-010-0184-8, 2010.
- Z. Ma and C. W. Rowley. Low-dimensional linearized models for systems with periodic orbits, with application to the Ginzburg-Landau equation. AIAA Paper 2008-4196, 4th Flow Control Conference, June 2008.
- Z. Ma, C. W. Rowley, and G. Tadmor. Snapshot-based balanced truncation for linear time-periodic systems. *IEEE Transactions on Automatic Control*, 55(2):469–473, Feb. 2010.

## Honors & Awards Received

- Distinguished Teacher Award, Princeton School of Engineering and Applied Science, 2010.
- Institute for Defense Analysis, Defense Science Study Group member, 2008–2009.
- Princeton Engineering commendation list for outstanding teaching, for MAE 434, Fall 2009
- Princeton Engineering commendation list for outstanding teaching, for MAE 501, Fall 2007
- Princeton Engineering commendation list for outstanding teaching, for MAE 433, Spring 2007.

## Interactions/transitions

The results of this research were presented at several conferences: the AIAA Aerospace Sciences Meetings in 2008, 2009, 2010, and SIAM Dynamical Systems meeting in Snowbird, UT (May 2007). A number of interactions with AFRL perssonel were maintained, especially with Dr. Michael Ol and Dr. James Myatt.