International Journal of Bifurcation and Chaos, Vol. 15, No. 3 (2005) 997–1013

MODEL REDUCTION FOR FLUIDS, USING BALANCED PROPER ORTHOGONAL DECOMPOSITION

C. W. ROWLEY
Department of Mechanical and Aerospace Engineering,
Princeton University, Princeton, NJ 08544, USA

Received May 15, 2004; Revised June 7, 2004

Many of the tools of dynamical systems and control theory have gone largely unused for fluids, because the governing equations are so dynamically complex, both high-dimensional and nonlinear. Model reduction involves finding low-dimensional models that approximate the full high-dimensional dynamics. This paper compares three different methods of model reduction: proper orthogonal decomposition (POD), balanced truncation, and a method called balanced POD. Balanced truncation produces better reduced-order models than POD, but is not computationally tractable for very large systems. Balanced POD is a tractable method for computing approximate balanced truncations, that has computational cost similar to that of POD. The method presented here is a variation of existing methods using empirical Gramians, and the main contributions of the present paper are a version of the method of snapshots that allows one to compute balancing transformations directly, without separate reduction of the Gramians; and an output projection method, which allows tractable computation even when the number of outputs is large. The output projection method requires minimal additional computation, and has *a priori* error bounds that can guide the choice of rank of the projection. Connections between POD and balanced truncation are also illuminated: in particular, balanced truncation may be viewed as POD of a particular dataset, using the observability Gramian as an inner product. The three methods are illustrated on a numerical example, the linearized flow in a plane channel.

Keywords: Model reduction; proper orthogonal decomposition; balanced truncation; snapshots.

1. Introduction

The past several decades have produced major advances in techniques for analyzing dynamical systems, both analytically and numerically. However, despite continuing improvements in computing power, many systems of interest remain out of reach of these tools, because of their high dimension. For instance, the mechanisms by which a fluid flow transitions from laminar to turbulent are still not fully understood: at this point, it is not even clear whether the mechanisms are fundamentally nonlinear [Holmes *et al.*, 1996] or linear [Farrell & Ioannou, 1993; Bamieh & Daleh, 2001]. The full nonlinear partial differential equations that describe a fluid flow are too complex to be analyzed directly, so in order to answer questions such as these, lower-dimensional models that approximate the full system are desirable.

The problem of obtaining a lower-dimensional approximation to a high-dimensional dynamical system is known as *model reduction*. This paper reviews two well-known approaches to model reduction, and presents a method which compares favorably with both of these. The method of *proper orthogonal decomposition* (POD) and Galerkin projection is popular in the fluids community, and in this method, one obtains a lower-dimensional approximation by projecting the full nonlinear system onto a set of basis functions

determined from empirical data. However, the POD/Galerkin method can yield unpredictable results, and is sensitive to details such as the empirical data used [Rathinam & Petzold, 2003], and the choice of inner product [Colonius & Freund, 2002]. POD/Galerkin models near stable equilibrium points can even be unstable [Smith, 2003].

A related method known as *balanced truncation* was developed in the control theory community for stable, linear, input–output systems, and does not suffer the same limitations as the POD method. Most notably, balanced truncation has error bounds that are close to the lowest error possible from any reduced-order model. In addition, this method has recently been extended to nonlinear systems using two distinct approaches [Lall *et al.*, 2002; Scherpen, 1993]. Balanced truncation has been used on some fluid problems [Cortelezzi & Speyer, 1998], but becomes computationally intractable for systems of very large dimension (e.g. 10 000 states or more), and so is not practical for many fluids systems.

This paper presents a method we refer to as *balanced* proper orthogonal decomposition, which combines ideas from POD and balanced truncation. The goal is to compute balanced truncations, or approximations to these, with computational cost similar to POD. Several previous methods have combined ideas from POD and balanced truncation, including the original work of Moore [1981]. The method presented here relies heavily on the work of Lall *et al.* [1999, 2002], who used empirical Gramians to generalize balanced truncation to nonlinear systems. Our goal is to use empirical Gramians to compute balancing transformations for very large systems. Previous works have addressed this problem as well, notably the work of Willcox and Peraire [2002], which used POD to compute low-rank approximations to the Gramians, from which the balancing transformation was computed using an efficient solver to find the eigenvectors of their product. However, this method has several drawbacks. In particular, it becomes intractable when the number of outputs is large, as a separate adjoint simulation is required for each output. Furthermore, in reducing the rank of the controllability and observability of Gramians before the balancing is performed, one risks prematurely truncating states that are poorly observable yet very strongly controllable, which can lead to less accurate models, as we shall see in the numerical example shown in Figs. 7 and 8.

The present method overcomes this latter drawback using a different method of snapshots, described in Sec. 3.1, in which one computes the balancing transformation directly from the snapshots, without individual reduction of the Gramians, and without a separate eigenvector solve. Furthermore, we describe an output projection method in Sec. 3.2, which allows the empirical observability Gramian to be computed even when the number of outputs is large, using many fewer adjoint simulations. This output projection is optimal in an L_2 sense, involves very little extra computation, and comes with an *a priori* error bound which can guide the rank of the output projection used [Eq. (27)]. Like balanced truncation, the present method is limited to stable, linear systems. However, because our method uses many of the same ideas as Lall *et al.* [1999] (in particular, empirical Gramians constructed from impulse responses), it is likely that similar computational techniques may be applied to nonlinear systems as well.

The paper is outlined as follows: in Sec. 2, we review the methods of POD/Galerkin projection and balanced truncation; we present our method in Sec. 3; and in Sec. 4, we compare the three methods on a example, the linearized flow in a plane channel.

2. Background on Model Reduction

The model reduction methods discussed in this paper fall in the category of *projection methods*, in that they involve projecting the equations of motion onto a subspace of the original phase space. The methods of POD/Galerkin and balanced truncation are briefly reviewed here, both for comparison with balanced POD, and also because our method uses ideas from both POD and balanced truncation. There are many other methods available for reducing both linear and nonlinear systems, and several of these are reviewed in [Antoulas *et al.*, 2001].

2.1. *Proper orthogonal decomposition*

Proper orthogonal decomposition, also known as principal component analysis, or the Karhunen–Loéve expansion, has been used for some time in developing low-dimensional models of fluids [Lumley, 1970; Sirovich, 1987; Holmes *et al.*, 1996]. The idea is, given a set of data that lies in a vector space $\mathcal{V}$, to find a subspace $\mathcal{V}_r$ of fixed dimension r

such that the error in the projection onto the subspace is minimized. Here, for simplicity, we will consider the case where $\mathcal{V} = \mathbb{R}^n$. For a fluid, $\mathcal{V}$ will be infinite-dimensional, consisting of functions on some spatial domain (for instance, velocity and pressure everywhere), but we will assume that the equations have already been discretized in space, for instance by a finite-difference or spectral method, so that $\mathcal{V}$ has finite dimension n (e.g. for a finite-difference simulation, n is the number of grid-point times the number of flow variables). For the infinite-dimensional case, see [Holmes *et al.*, 1996; Rowley *et al.*, 2004].

Suppose we have a set of data given by $x(t) \in \mathbb{R}^n$, with $0 \le t \le T$. We seek a projection $P_r : \mathbb{R}^n \to \mathbb{R}^n$ of fixed rank r, that minimizes the total error

$$\int_0^T \|x(t) - P_r x(t)\|^2 \, dt. \tag{1}$$

To solve this problem, introduce the $n \times n$ matrix

$$R = \int_0^T x(t)x(t)^* \, dt, \tag{2}$$

where * denotes the transpose, and find the eigenvalues and eigenvectors of R, given by

$$R\varphi_k = \lambda_k \varphi_k, \quad \lambda_1 \ge \cdots \ge \lambda_n \ge 0. \tag{3}$$

Since R is symmetric, positive-semidefinite, all the eigenvalues λ_k are real and non-negative, and the eigenvectors φ_k may be chosen to be orthonormal. The main result of POD is that the optimal subspace of dimension r is spanned by $\{\varphi_1, \ldots, \varphi_r\}$, and the optimal projection P_r is then given by

$$P_r = \sum_{k=1}^{r} \varphi_k \varphi_k^*.$$

The vectors φ_k are called *POD modes*.

2.1.1. *Galerkin projection*

One can then form reduced order models using *Galerkin projection* onto this subspace. Suppose the dynamics of a system are described by

$$\dot{x}(t) = f(x(t)). \tag{4}$$

Galerkin projection specifies dynamics of a variable $x_r(t) \in \text{span}\{\varphi_1, \ldots, \varphi_r\}$ by $\dot{x}_r(t) = P_r f(x_r(t))$, that is, simply projecting the original vector field f onto the r-dimensional subspace. Writing

$$x_r(t) = \sum_{j=1}^{r} a_j(t)\varphi_j, \tag{5}$$

substituting into the equations, and multiplying by φ_k^*, one obtains

$$\dot{a}_k(t) = \varphi_k^* f(x_r), \quad k = 1, \ldots, r, \tag{6}$$

a set of r ODEs that describe the evolution of $x_r(t)$.

2.1.2. *Method of snapshots*

To compute the POD modes, one must solve an $n \times n$ eigenvalue problem (3). For a discretization of a fluid problem, the dimension n often exceeds 10^6, so direct solution of this eigenvalue problem is not often feasible. If the data is given as "snapshots" $x(t_j)$ at discrete times $t_1, \ldots, t_m$, then one can transform the $n \times n$ eigenvalue problem (3) into an $m \times m$ eigenvalue problem [Sirovich, 1987]. In this case, the integral in (3) becomes a sum

$$R = \sum_{j=1}^{m} x(t_j)x(t_j)^* \delta_j \tag{7}$$

where δ_j are quadrature coefficients. Assembling the data into an $n \times m$ matrix

$$X = \begin{bmatrix} x(t_1)\sqrt{\delta_1} & \cdots & x(t_m)\sqrt{\delta_m} \end{bmatrix} \tag{8}$$

the sum (7) may be written $R = XX^*$. In the method of snapshots, one then solves the $m \times m$ eigenvalue problem

$$X^* X u_k = \lambda_k u_k, \quad u_k \in \mathbb{R}^m, \tag{9}$$

where the eigenvalues λ_k are the same as in (3). The eigenvectors u_k may be chosen to be orthonormal, and the POD modes are then given by $\varphi_k = Xu_k/\sqrt{\lambda_k}$. In matrix form, with $\Phi = [\varphi_1 \ \cdots \ \varphi_m]$, and $U = [u_1 \ \cdots \ u_m]$, this becomes

$$\Phi = XU\Lambda^{-1/2}. \tag{10}$$

The $m \times m$ eigenvalue problem (9) is more efficient than the $n \times n$ eigenvalue problem (3) when the number of snapshots m is smaller than the number of states n.

2.1.3. *Remarks and limitations*

A physical explanation of POD modes is that they maximize the average energy in the projection of the data onto the subspace spanned by the modes. This is equivalent to minimizing the error (1), since

$$\arg\min_{\{\varphi_k\}} \langle \|x - P_r x\|^2 \rangle = \arg\max_{\{\varphi_k\}} \langle \|P_r x\|^2 \rangle$$

where $\langle\cdot\rangle$ is the average over the data ensemble (this follows from the Pythagorean theorem, since P_r is an orthogonal projection). In particular, the energy in the projection is given by

$$\int_0^T \|P_r x(t)\|^2 \, dt = \sum_{k=1}^{r} \lambda_k. \tag{11}$$

Though POD modes are very effective (indeed optimal) at approximating a given dataset, they are not necessarily the best modes for describing the *dynamics* that generate a particular dataset, since low-energy features may be critically important to the dynamics. For instance, in a fluid flow where acoustic resonances occur, acoustic waves play a crucial role, even though they have much smaller energy than hydrodynamic pressure fluctuations. In practice, one sometimes neglects some of the higher-energy POD modes in forming reduced-order models [Smith, 2003], in favor of lower-energy modes that are more dynamically important. In fact, adding more POD modes can even make dynamical models worse [Rowley *et al.*, 2004]. These are undesirable characteristics of a model reduction procedure, and part of the motivation behind balanced POD is to improve on these limitations.

2.2. *Balanced truncation*

Balanced truncation is a method of model reduction for stable, linear input–output systems, introduced by [Moore, 1981]. Consider a stable linear input–output system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{12}$$

where $u(t) \in \mathbb{R}^p$ is a vector of inputs, $y(t) \in \mathbb{R}^q$ is a vector of outputs, and $x(t) \in \mathbb{R}^n$ is the state vector.

One begins by defining *controllability* and *observability Gramians*, which are symmetric, positive-semidefinite matrices defined by

$$\begin{aligned} W_c &= \int_0^\infty e^{At} B B^* e^{A^* t} \, dt \\ W_o &= \int_0^\infty e^{A^* t} C^* C e^{At} \, dt, \end{aligned} \tag{13}$$

usually computed by solving the Lyapunov equations

$$\begin{aligned} AW_c + W_c A^* + BB^* &= 0 \\ A^* W_o + W_o A + C^* C &= 0. \end{aligned} \tag{14}$$

The controllability Gramian W_c measures to what degree each state is excited by an input. For two states x_1 and x_2 with $\|x_1\| = \|x_2\|$, if $x_1^* W_c x_1 > x_2^* W_c x_2$, then state x_1 is "more controllable" than x_2 (i.e. it takes a smaller input to drive the system from rest to x_1 than to x_2). The Gramian W_c is positive-definite if and only if all states are reachable with some input $u(t)$.

Conversely, the observability Gramian W_o measures to what degree each state excites future outputs. For an initial state x_0, and with zero input, one has $\|y\|_2^2 = x_0^* W_o x_0$, where $\|\cdot\|_2$ denotes the $L_2[0,\infty)$ norm. States which excite larger output signals are called "more observable," and in this sense are more dynamically important than states that are less observable.

The Gramians depend on the coordinates, and under a change of coordinates $x = Tz$, they transform as

$$W_c \mapsto T^{-1} W_c (T^{-1})^*, \quad W_o \mapsto T^* W_o T.$$

Balancing refers to changing to coordinates in which the controllability and observability properties are balanced — more precisely, the transformed Gramians are equal and diagonal:

$$T^{-1} W_c (T^{-1})^* = T^* W_o T = \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n). \tag{15}$$

The diagonal elements $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ are called the *Hankel singular values* of the system, and are independent of the coordinate system. A basic result is that a balancing transformation T exists as long as the system is both controllable and observable (i.e. $W_c, W_o > 0$). The transformation is found by computing appropriately scaled eigenvectors of the product $W_c W_o$ (in particular, $W_c W_o T = T\Sigma^2$). In the balanced coordinates, the states that are least influenced by the input also have the least influence on the output. Balanced truncation involves first changing to these coordinates, and then truncating the least controllable/observable states, which have little effect on the input–output behavior.

2.2.1. *Error bounds*

A useful property of balanced truncation is that one has *a priori* error bounds that are close to the lower bound achievable by any reduced-order model. To understand these error bounds, consider the transfer function

$$\hat{G}(s) = C(sI - A)^{-1} B,$$

which relates the Laplace transform of the input to the Laplace transform of the output ($\hat{y}(s) = \hat{G}(s)\hat{u}(s)$). The L_2-induced operator norm of $\hat{G}$ is defined by

$$\max_u \frac{\|Gu\|_2}{\|u\|_2} = \|G\|_\infty \equiv \max_\omega \sigma_1(\hat{G}(i\omega)), \qquad (16)$$

where $\sigma_1(M)$ denotes the maximum singular value of the matrix M. The following error bounds are standard results [Dullerud & Paganini, 1999]: first, any reduced order model G_r with r states must satisfy

$$\|G - G_r\|_\infty > \sigma_{r+1}, \qquad (17)$$

where σ_{r+1} is the first neglected Hankel singular value of G. This is a fundamental limitation for any reduced order model. Balanced truncation also guarantees an upper bound of the error:

$$\|G - G_r\|_\infty < 2 \sum_{j=r+1}^{n} \sigma_j, \qquad (18)$$

which is usually close to the lower bound (17), if the Hankel singular values drop off quickly. Balanced truncation is not optimal, in the sense that there may be other reduced-order models with smaller error norms, but *a priori* guarantees and strong heuristic justification make it a popular and effective technique.

2.2.2. *Empirical Gramians*

Instead of computing the Gramians by solving Lyapunov equations (14), one may compute them from data from numerical simulations. This was the original approach used by [Moore, 1981], and was used in [Lall *et al.*, 1999, 2002] to extend balanced truncation to nonlinear systems.

2.2.3. *Controllability Gramian*

To compute the controllability Gramian for a system with p inputs, writing $B = [b_1, \ldots, b_p]$, one forms the state responses to unit impulses

$$\begin{aligned} x_1(t) &= e^{At} b_1 \\ &= \text{response to impulsive input } u_1(t) = \delta(t) \\ &\vdots \\ x_p(t) &= e^{At} b_p \\ &= \text{response to impulsive input } u_p(t) = \delta(t) \end{aligned}$$

Then the controllability Gramian is given by

$$W_c = \int_0^\infty (x_1(t)x_1(t)^* + \cdots + x_p(t)x_p(t)^*)dt. \qquad (19)$$

Note the similarity between the expression above and the operator in (2) that arises in POD of the dataset $\{x_1(t), \ldots, x_p(t)\}$. In fact, the POD modes for this dataset of impulse responses are just the largest eigenvectors of W_c, or, in other words, the most controllable modes of the realization. Note that since the Gramian matrices depend on the coordinate system, so do the POD modes of this dataset.

If data from simulations is used to find the impulse responses, then it is usually given at discrete times $t_1, \ldots, t_m$, and the integral above becomes a quadrature sum, as in (7), and we may stack the snapshots as columns of a matrix

$$X = [x_1(t_1)\sqrt{\delta_1} \quad \cdots \quad x_1(t_m)\sqrt{\delta_m} \quad \cdots \quad x_p(t_1)\sqrt{\delta_1} \cdots x_p(t_m)\sqrt{\delta_m}], \qquad (20)$$

where again δ_j are quadrature coefficients. The quadrature approximation to (19) is then

$$W_c = XX^*. \qquad (21)$$

2.2.4. *Observability Gramian*

The procedure for computing the empirical observability Gramian proceeds analogously: we compute impulse responses of the adjoint system

$$\dot{z} = A^* z + C^* v.$$

If q is the number of outputs and $C^* = (c_1, \ldots, c_q)$, then let

$$\begin{aligned} z_1(t) &= e^{A^* t} c_1 \\ &= \text{response to impulsive input } v_1(t) = \delta(t) \\ &\vdots \\ z_q(t) &= e^{A^* t} c_q \\ &= \text{response to impulsive input } v_q(t) = \delta(t), \end{aligned}$$

from which the observability Gramian is given by

$$W_o = \int_0^\infty (z_1(t)z_1(t)^* + \cdots + z_q(t)z_q(t)^*)dt.$$

One then forms the data matrix Y, as in (20), and writes the Gramian as

$$W_o = YY^*.$$

Note that this method requires q integrations of the adjoint system, where q is the number of outputs. Thus, this method is not feasible when the number of outputs is large, for instance, if the output is the full state. The empirical Gramian may also be computed from n simulations of the primal system $\dot{x} = Ax$, where n is the number of states (as is done in [Lall *et al.*, 2002]), but clearly this is also not feasible when the number of states is large. This difficulty is the motivation behind the output projection method to be discussed in Sec. 3.2.

3. Balanced POD

The main idea of balanced POD is to obtain an approximation to balanced truncation that is computationally tractable for large systems. The present method involves two components: computing the balancing transformation directly from snapshots of empirical Gramians, without needing to compute the Gramians themselves; and an output projection method to enable tractable computation even when the number of outputs is large. The method has deep connections with POD: it may be viewed as POD with respect to a particular inner product, or as a biorthogonal decomposition, as discussed in Sec. 3.4.

3.1. *Balanced truncation using the method of snapshots*

Suppose the controllability and observability Gramians may be factored as

$$W_c = XX^*, \quad W_o = YY^*, \tag{22}$$

where W_c and W_o are $n \times n$ square matrices, but X and Y may be rectangular, with differing dimensions. For instance, X and Y may be data matrices used to form empirical Gramians, as described in the previous section. In the method of snapshots used here, the balancing modes are computed by forming the singular value decomposition (SVD) of the matrix Y^*X:

$$\begin{aligned} Y^*X &= U\Sigma V^* \\ &= \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} \\ &= U_1 \Sigma_1 V_1^* \end{aligned} \tag{23}$$

where $\Sigma_1 \in \mathbb{R}^{r\times r}$ is invertible, r is the rank of Y^*X, and $U_1^*U_1 = V_1^*V_1 = I_r$. Define the matrices $T_1 \in \mathbb{R}^{n\times r}$ and $S_1 \in \mathbb{R}^{r\times n}$ by

$$T_1 = XV_1\Sigma_1^{-1/2}, \quad S_1 = \Sigma_1^{-1/2}U_1^*Y^*. \tag{24}$$

A proposition proved in the appendix establishes that if $r = n$ (that is, the Gramians are full rank), then the matrix Σ_1 contains the Hankel singular values, T_1 determines the balancing transformation, and S_1 is its inverse. Furthermore, if $r < n$, then the columns of T_1 form the first r columns of the balancing transformation, and the rows of S_1 form the first r rows of the inverse transformation.

Remark. The major advantage of the above method for computing the balancing transformation is that the Gramians themselves never need to be computed. Only one SVD is needed, of a matrix with dimension $N_p \times N_d$, where N_p is the number of primal snapshots (columns of X), and N_d is the number of dual snapshots (columns of Y). If the number of snapshots is much smaller than the number of states n, as is typical for a problem in fluids, then this represents considerable savings. In particular, the size of the SVD is independent of n, and once the snapshots are computed, the entire method scales linearly with n. Thus, the overall computation time is similar to POD (compare (23)–(24) with (9)–(10)), except that here one also needs to compute adjoint snapshots, which do not arise in POD.

The method above is also similar to a well-known method for computing balancing transformations from the Cholesky factorization of the Gramians [Laub *et al.*, 1987]. The present method differs in that the factorization (22) need not be the Cholesky factorization, and neither of the Gramians needs to be full-rank. (In particular, the system does not need to be controllable or observable.) The present method does share the same desirable numerical characteristics as the method in [Laub *et al.*, 1987], in particular that the Gramians never need to be "squared up," and thus the method is less sensitive to numerical round-off than methods that

involve computing the full Gramians W_c and W_o, rather than a factorization.[1]

3.2. *Output projection*

Recall from Sec. 2.2 that in order to compute data for the observability Gramian, one requires q simulations of the adjoint system, where q is the number of outputs. This procedure is clearly not feasible if the number of outputs is large. The idea of this section is to alleviate this problem by projecting the output onto an appropriate subspace, in such a way that the input–output behavior is almost unchanged. Instead of the system (12), consider the related system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= P_r C x \end{aligned} \tag{25}$$

where P_r is an orthogonal projection with rank r. Such a projection allows us to compute the empirical observability Gramian using only r simulations of the adjoint system, rather than q simulations. To see this, write the projection P_r as the product $P_r = \Phi_r \Phi_r^*$, where Φ_r is a $q \times r$ matrix, with $\Phi_r^* \Phi_r = I_r$ (this can always be done for any orthogonal projection). The observability Gramian (13) then becomes

$$W_o = \int_0^\infty e^{A^* t} C^* \Phi_r \Phi_r^* C e^{At}\, dt$$

and so may be computed from r simulations of the adjoint system

$$\dot{z}(t) = A^* z + C^* \Phi_r v$$

where $v \in \mathbb{R}^r$. When the number of outputs q is large, the reduction in computational cost is substantial.

We would like to choose P_r such that the input–output behavior of (25) is as close as possible to the input–output behavior of (12). We can measure this input–output behavior by considering the *impulse response matrix* $G(t)$, whose element $G_{ij}(t)$ is the output component $y_i(t)$ corresponding to an impulsive input $u_j(t) = \delta(t)$. The impulse response completely determines the input–output behavior of a linear system. If $G(t)$ is the impulse response of (12), then the impulse response of (25) is $P_r G(t)$, and we seek a projection P_r that minimizes the error

$$\int_0^\infty \|G(t) - P_r G(t)\|^2\, dt \tag{26}$$

with respect to some norm on matrices. If we use a norm induced by an inner product, for instance the Frobenius norm $\|A\|_F^2 = \mathrm{Tr}(A^* A)$, which is induced by the inner product $\langle A, B \rangle = \mathrm{Tr}(A^* B)$, then *the projection P_r that minimizes the error (26) is the projection onto the first r POD modes of the dataset $G(t)$*. For instance, if $\Phi_r = [\varphi_1 \quad \cdots \quad \varphi_r]$ is a matrix containing the first r POD modes of $G(t)$, then $P_r = \Phi_r \Phi_r^*$ is the projection that minimizes (26).

A convenient numerical feature of this method for computing P_r is that the necessary snapshots for computing the POD modes of $G(t)$ have already been computed, for the empirical controllability Gramian. To compute the snapshots for W_c, as in Sec. 2.2, we compute impulse responses $x_1(t), \ldots, x_p(t)$, for each of the p inputs. The dataset required for computing P_r is simply $Cx_1(t), \ldots, Cx_p(t)$, so we need only to multiply each of our snapshots by the output matrix C.

3.2.1. *Error bounds*

One can also quantify the error for the projected system. In particular, if $\lambda_1, \ldots, \lambda_m$ denote the POD eigenvalues of the dataset $\{Cx_1(t), \ldots, Cx_p(t)\}$, then

$$\|G - P_r G\|_2^2 = \sum_{j=r+1}^{m} \lambda_j, \tag{27}$$

where m is the number of outputs, and the 2-norm is given by

$$\|G\|_2^2 = \int_0^\infty \mathrm{Tr}(G(t)^* G(t))\, dt. \tag{28}$$

The proof follows immediately from a variant of (11). This result gives us guidance in choosing the number of modes to keep in the projection, based on the desired accuracy of the reduced-order model, and the POD eigenvalues computed from the impulse response data.

[1] As one reviewer remarked, POD modes may also be computed by a SVD of the snapshot matrix X from (8). This approach also has better roundoff properties than computing the eigenvalue decomposition of $X^* X$ as in (9), although it requires more computation.

3.3. *Summary*

To summarize, the steps in the balanced POD method are as follows:

1. Integrate solutions $x_1(t), \ldots, x_p(t)$ of the system $\dot{x} = Ax$, with initial conditions $x_k(0) = b_k$, where b_k denotes the kth column of the B matrix in (12).
2. Compute POD modes φ_k of the dataset $\{Cx_1(t), \ldots, Cx_p(t)\}$, and choose a projection rank r such that the error (27) is acceptable.
3. Integrate solutions $z_1(t), \ldots, z_r(t)$ of the adjoint system $\dot{z} = A^*z$, with initial conditions $z_k(0) = C^*\varphi_k$.
4. Form the data matrices X and Y for the primal and dual solutions, as in (20).
5. Compute the SVD of Y^*X, and the balanced POD modes are given by (24).

If the number of outputs is small, then one may skip step 2 and in step 3 use initial conditions $z_k(0) = c_k^*$, where c_k is the kth row of C.

Reduced-order models may then be formed by transforming to balanced coordinates and projecting. Note that there is no need to transform all of the states: if we write

$$\begin{aligned} x(t) = Tz(t) &= \begin{bmatrix} T_1 & T_2 \end{bmatrix} \begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} \\ &= T_1 z_1(t) + T_2 z_2(t), \end{aligned}$$

where $z_1(t)$ are states to be retained and $z_2(t)$ are states to be truncated, then the transformed equations are

$$\begin{aligned} \dot{z}_1 &= S_1 A T_1 z_1 + S_1 A T_2 z_2 + S_1 B u \\ \dot{z}_2 &= S_2 A T_1 z_1 + S_2 A T_2 z_2 + S_2 B u \\ y &= C T_1 z_1 + C T_2 z_2, \end{aligned}$$

where $S = T^{-1}$. Setting $z_2 = 0$ gives the truncated model

$$\begin{aligned} \dot{z}_1 &= S_1 A T_1 z_1 + S_1 B u \\ y &= C T_1 z_1 \end{aligned}$$

Thus, to compute a reduced-order model of order r, all we need is the first r columns of T and the first r rows of S, given by (24). Note, however, that this is not the same as orthogonal projection onto the subspace spanned by the first r columns of T, since the columns of T are not orthogonal.

3.4. *Relation to POD*

There are deep connections between the POD/Galerkin method and balanced truncation, which are elucidated by the balanced POD procedure. For instance, balanced truncation may be viewed as a biorthogonal decomposition, instead of the orthogonal decomposition given by POD. Alternatively, balanced truncation may be viewed as a special case of POD, using a particular dataset (impulse responses), and using the observability Gramian as an inner product. The former point of view is useful for numerics, and the latter is useful for analysis, as it yields a guarantee that if balanced POD is used, then Galerkin projections of stable nonlinear systems are guaranteed to be stable as well.

3.4.1. *Biorthogonal decomposition*

In the POD/Galerkin procedure, one finds a sequence of orthogonal basis functions $\{\varphi_j\}$, for projection of the dynamics. Balanced truncation can be viewed in the same way, but using a sequence of biorthogonal functions $\{\varphi_j\}, \{\psi_j\}$. Let the matrices T_1 and S_1 from (24) be written

$$T_1 = \begin{bmatrix} \varphi_1 & \cdots & \varphi_r \end{bmatrix}, \quad S_1 = \begin{bmatrix} \psi_1^* \\ \vdots \\ \psi_r^* \end{bmatrix},$$

with $\varphi_j, \psi_j \in \mathbb{R}^n$. Then since $S_1 T_1 = I_r$, we have $\psi_i^* \varphi_j = \delta_{ij}$, so the sequences are biorthogonal. Now, approximate $x(t)$ as in (5), as

$$x_r(t) = \sum_{j=1}^{r} a_j(t)\varphi_j, \quad a_j(t) = \psi_j^* x(t).$$

Substituting into the equation $\dot{x} = f(x)$, multiplying by ψ_k^* and using biorthogonality now gives

$$\dot{a}_k = \psi_k^* f(x),$$

which is identical to (6), but using the *adjoint modes* ψ_k for the projection. Of course, one needs a linear system to define Gramians or adjoint equations, but the idea is that even for a nonlinear system, one may compute balancing modes $\{\varphi_j\}, \{\psi_j\}$ using a linearization, or a method similar to that in [Lall *et al.*, 2002], and then project the nonlinear system $\dot{x} = f(x)$ without having to transform the entire state before truncating.

3.4.2. *Observability Gramian as an inner product*

One of the difficulties with the POD/Galerkin method is that the inner product used for computing POD modes and projecting the dynamics is arbitrary. Sometimes, an appropriate inner product is obvious, as for incompressible flow [Holmes *et al.*, 1996], but other times, as for compressible flow, a suitable inner product is not obvious [Rowley *et al.*, 2004], and different choices can give dramatically different results [Colonius & Freund, 2002]. Perhaps the deepest connection between POD/Galerkin and balanced truncation is that for a stable linear system, balanced truncation may be viewed as a *special case* of POD, using impulse responses for a dataset (i.e. the matrix X in (20)), and using the observability Gramian as an inner product.

To see this, first define an inner product on $\mathbb{R}^n$ by

$$\langle a, b\rangle_{W_o} = a^* W_o b \tag{29}$$

where W_o is the observability Gramian (which is positive definite as long as the system is observable). As mentioned in Sec. 2.2, W_o measures states of large "dynamical importance," so this inner product weights dynamically important states more heavily. The POD modes of the dataset X with respect to this inner product are eigenvectors of $R = XX^*W_o$ (see [Rowley *et al.*, 2004] for an explanation of POD with respect to an arbitrary inner product). These eigenvectors will be orthogonal with respect to the inner product (29), though not with respect to the standard inner product.

POD modes are normalized balancing modes. Since the dataset X was produced such that $XX^* = W_c$, the POD modes are just the eigenvectors of $R = W_c W_o$: in other words, they are the balancing modes, normalized differently. Furthermore, the eigenvalues of R are the squares of the Hankel singular values. If we compute the POD modes using the method of snapshots as in (9), we form the SVD $X^*W_oX = V_1\Sigma_1^2 V_1^*$, and the POD modes are columns of

$$\Phi = [\tilde{\varphi}_1 \quad \cdots \quad \tilde{\varphi}_r] = XV_1\Sigma^{-1}.$$

Note that these modes are the same as columns of T_1 in (24), with a different scaling. If we define "adjoint modes" $\tilde{\psi}_j = W_o\tilde{\varphi}_j$, then

$$\langle \tilde{\varphi}_i, \tilde{\varphi}_j\rangle_{W_o} = \tilde{\varphi}_i^* W_o \tilde{\varphi}_j = \tilde{\psi}_i^* \tilde{\varphi}_j = \delta_{ij}$$

so these adjoint modes may be viewed as a biorthogonal decomposition with respect to the standard inner product $\langle \psi, \varphi\rangle = \psi^*\varphi$, as in the previous section. These adjoint modes are also rescaled versions of the rows of S_1 in (24), since one easily checks that, with $W_o = YY^*$, and $X^*Y = U_1\Sigma_1 V_1^*$,

$$\tilde{S}_1 := \begin{bmatrix} \tilde{\psi}_1^* \\ \vdots \\ \tilde{\psi}_r^* \end{bmatrix} = \Phi^* W_o = \Sigma_1^{-1} V_1^* X^* Y Y^* = U_1^* Y^*,$$

a rescaling of S_1 in (24).

3.4.3. *Guaranteed stability*

A useful consequence of using the observability Gramian as an inner product for Galerkin projection is that in this case, the reduced-order model preserves the stability of an equilibrium point at the origin, even if the full model is nonlinear. It is well-known that balanced truncations of stable linear systems are stable, but POD/Galerkin models of nonlinear systems may be unstable even if the nonlinear system is linearly stable at the origin [Smith, 2003].

The stability result follows from a result in [Rowley *et al.*, 2004]: if the norm induced by an inner product is a Lyapunov function for a nonlinear system with a stable equilibrium point at the origin, then orthogonal projection of the dynamics onto *any* subspace will also be stable at the origin. One sees from (14) that $V(x) = \langle x, x\rangle_{W_o}$ is a Lyapunov function of the linearized system $\dot{x} = Ax$, with $\dot{V}(x) = -C^*C \leq 0$. If the nonlinear system $\dot{x} = f(x)$ has a linearly stable equilibrium point at the origin, with $Df(0) = A$, then $V(x)$ is also a Lyapunov function for the nonlinear system, and so Galerkin projections using $\langle \cdot, \cdot\rangle_{W_o}$ will also be stable.

4. Example: Linearized Channel Flow

In order to compare the effectiveness of the three model reduction methods considered in this paper, we consider the problem of fluid flow in a plane channel. In particular, we use linearized equations with a coarse enough discretization that conventional balanced truncation is still computationally tractable. Since balanced POD is meant to approximate balanced truncation, we may evaluate how close the approximation is, and compare the resulting models to those formed with the standard POD/Galerkin method. Focusing on linearized

equations allows us to use operator norms to objectively compare the errors in the reduced order models.

4.1. *Equations of motion*

Consider the problem of a fluid flowing in a plane channel, as depicted in Fig. 1. We focus on the linearized case, considering small perturbations about a steady, laminar flow. The flow is assumed periodic in the x- and z-directions, with no-slip boundary conditions at the walls $y = \pm 1$. We force the flow with a body force given by $B(y, z)f(t)$, acting in the wall-normal direction (here $B(y, z)$ specifies the spatial distribution of the force, and $f(t)$ is regarded as an input). We restrict ourselves to streamwise-constant perturbations (no variations in the x-direction), and for this case the equations are given by

$$\frac{\partial v}{\partial t} = \frac{1}{R}\nabla^2 v + Bf$$
$$\frac{\partial \eta}{\partial t} = \frac{1}{R}\nabla^2 \eta - U'\frac{\partial v}{\partial z}$$

where v is the wall-normal velocity and $\eta = u_z - w_x$ is the perturbation in wall-normal vorticity. Numerical investigations indicate that the laminar velocity profile $\mathbf{u} = (U(y), 0, 0)$, with $U(y) = 1 - y^2$, is linearly stable for Reynolds numbers $R < 5772$ [Drazin & Reid, 1981], so the infinite-time Gramians will be well defined.

For the numerical examples considered here, we consider $R = 100$, on the domain $z \in [0, 2\pi]$, and discretize the problem using 16 Chebyshev modes in the y-direction, and 16 Fourier modes in the z-direction. The forcing $B(y, z)$ is zero everywhere except in a small region at the center of the domain ($y = 0$, $z = \pi$). We take the output to be the entire state, that is, the values of (v, η) everywhere in space. The total number of states is $2 \cdot 16 \cdot 15 = 480$, which is small enough that we may compute the full Gramians exactly, for comparison with our approximate methods.

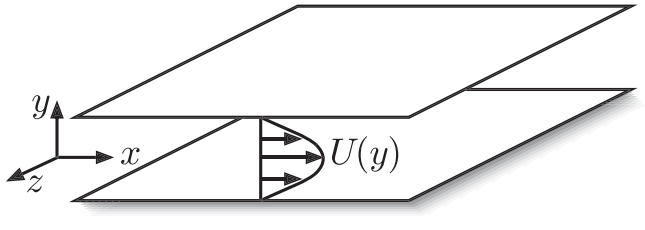

Fig. 1. Schematic of channel flow example.

4.2. *Results*

4.2.1. *Hankel singular values*

We begin by comparing the Hankel singular values σ_j, shown in Fig. 2. Here, the exact values for balanced truncation are compared to the approximate values for balanced POD, for both five-mode and ten-mode output projections P_r. Also shown are the POD eigenvalues λ_j, computed from (9), and observe that the eigenvalues fall off quite rapidly. The first five POD modes capture 95.6% of the energy, while the first ten modes capture 99.8% of the energy. Thus, one expects that five-mode and ten-mode output projections should closely match the full input–output system.

In Fig. 2, the exact Hankel singular values are computed using the algorithm in [Laub *et al.*, 1987], while the approximate versions are computed from (23). Both the primal and dual solutions were computed using 1000 snapshots equally spaced within time $0 \leq t \leq 200$, by which time transients have decayed to a maximum value of 0.0002, from a maximum value of 1 at the initial time.

For the five-mode output projection, the first five singular values match closely, while for the ten-mode output projection, the first ten singular values match. Though there is no guarantee that for an output projection of rank r, the first r singular values will be approximated well, empirically this seems to be the case, at least for the channel flow problem.

4.2.2. *Modes*

The first three modes are plotted in Figs. 3–5, which compare modes from exact balanced truncation, balanced POD with a five-mode output projection, and conventional POD. As explained in Sec. 3.4, for exact balanced truncation and balanced POD, the kth mode is the kth column of the transformation T, from (15) and (24), respectively. The POD modes are the eigenvectors from (3), also columns of the matrix Φ from (10).

The modes from balanced POD are nearly identical to those from exact balanced truncation, even for the five-mode output projection. For the ten-mode output projection, the modes also look

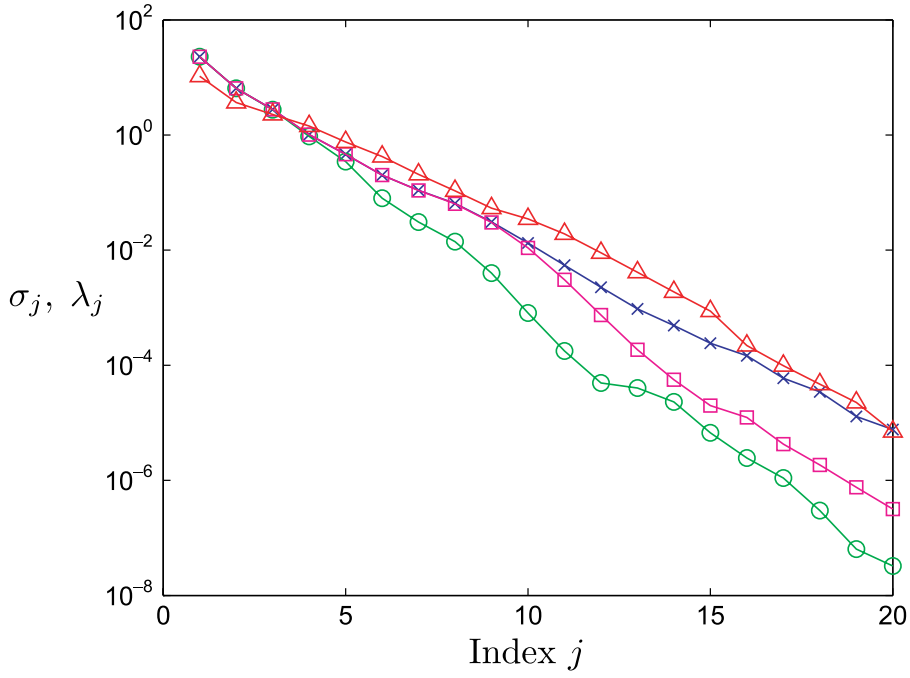

Fig. 2. Hankel singular values σ_j for linearized channel flow: balanced truncation (×), balanced POD with five-mode output projection (○), ten-mode output projection (□); and POD eigenvalues λ_j (△).

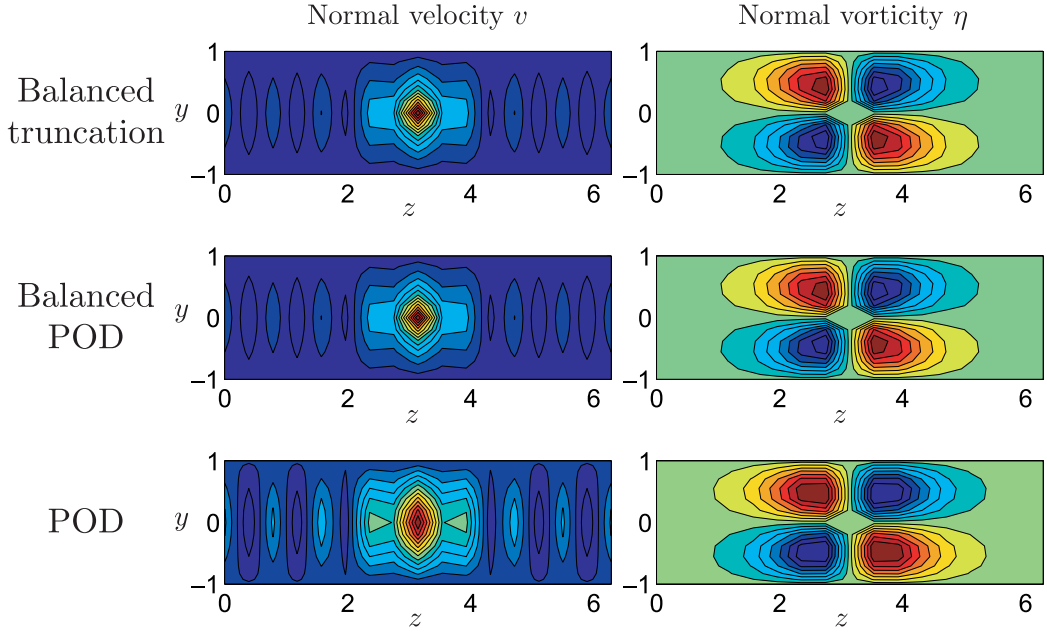

Fig. 3. Mode 1 for channel flow.

visually identical, so these are not shown. The conventional POD modes look similar in general structure, especially mode 1, but there are distinct differences in modes 2 and 3. Of course, we would not expect the POD modes to be the same as the balancing modes, unless the observability Gramian Y is the identity, so it is interesting that the POD modes look so similar.

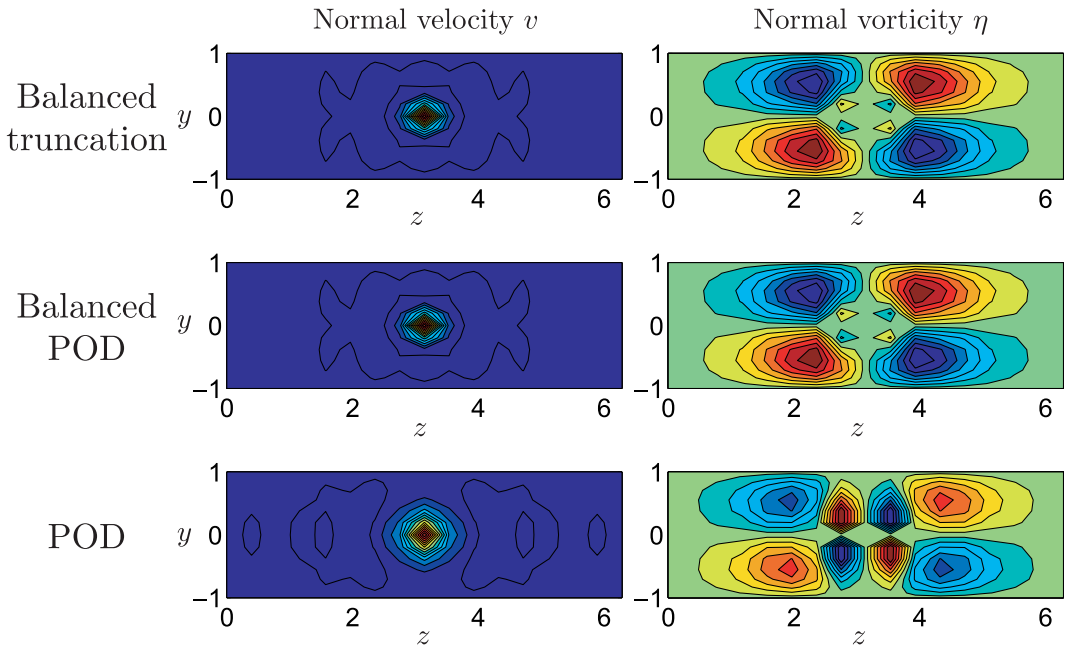

Fig. 4. Mode 2 for channel flow.

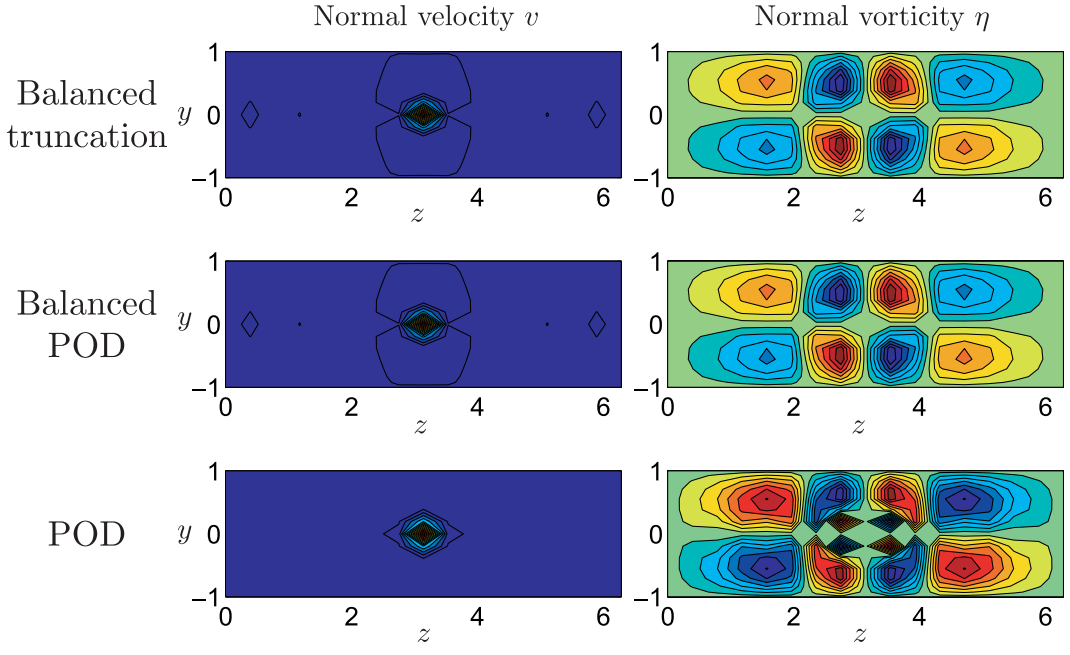

Fig. 5. Mode 3 for channel flow.

4.2.3. *Adjoint modes*

The corresponding adjoint modes for balanced POD are shown in Fig. 6. These look visually identical to the adjoint modes from balanced truncation (i.e. the first three rows of S_1 in (24)), so these are not shown. Recall that the POD modes are orthogonal, not biorthogonal, so the "adjoint modes" for POD are the same as the primal modes shown in Figs. 3–5. The adjoint modes in Fig. 6 look quite different from the primal modes or the POD modes,

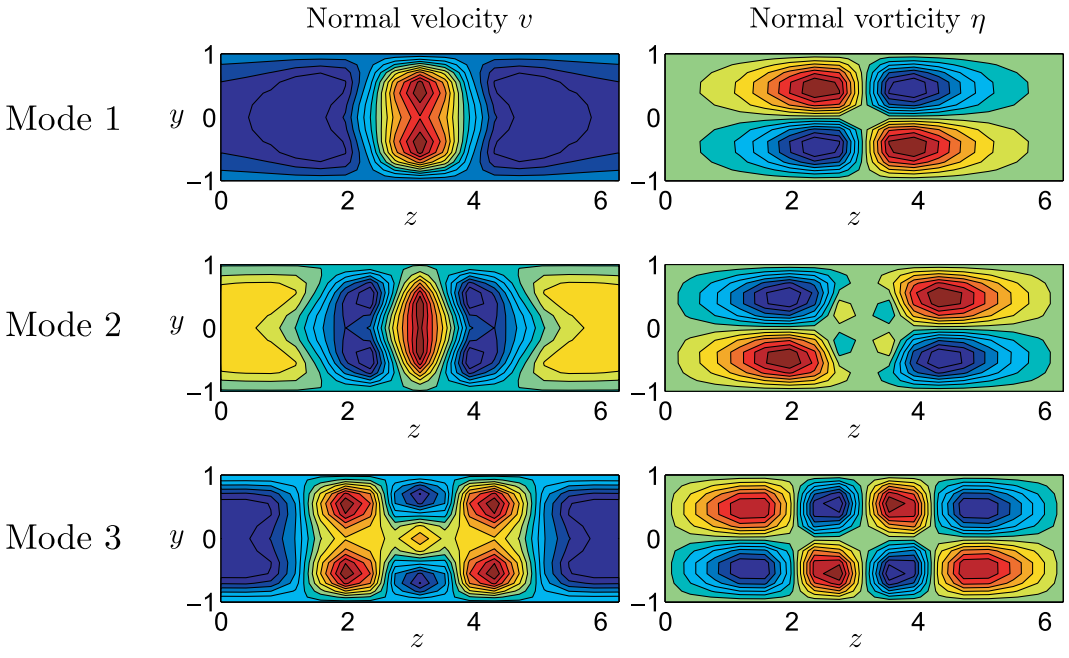

Fig. 6. Adjoint modes 1–3 for balanced POD. The adjoint modes for balanced truncation are nearly identical, and the adjoint modes for POD are the same as the primal modes.

so it is reasonable to say that, for this problem, the main difference between balanced POD and conventional POD is the choice of inner product used for the projection.

4.2.4. *Error norms*

The main reason for using a linear system to compare these model reduction procedures is to have an objective measure of how effective the various reduced-order models are at approximating the full-order system. For linear systems, we have norms which enable such an objective comparison. Perhaps the most intuitive norm is the H_2 norm, defined by (28). Since we have a single input, the impulse response matrix $G(t)$ is a column vector $g(t)$, and so

$$\|G\|_2^2 = \int_0^\infty \|g(t)\|^2 \, dt,$$

just the regular $L_2[0,\infty)$ norm of the impulse response vector. We can think of the error norm $\|G - G_r\|_2$ as being the RMS error between a simulation of the reduced-order model G_r and a simulation of the full model G, where the simulation begins with $v(x,z,0) = \eta(x,z,0) = 0$, and the forcing is $f(t) = \delta(t)$. This error is shown in Fig. 7, as the order r varies from 1 to 10. Notice that the error norms for balanced POD with both five-mode and ten-mode output projections are virtually the same as for balanced truncation, while POD is significantly worse for models of dimension six or smaller. For models of dimension greater than six, the error norms become smaller and all methods perform about the same.

Also shown is the error from an approximate balanced truncation in which the exact Gramians are computed, and then separately approximated by low-rank projections (to rank 30) using SVD. This separate reduction of Gramians is performed in the method of snapshots used in [Willcox & Peraire, 2002], although here their method of snapshots was not literally used, since it would require 480 adjoint simulations (the exact Gramians were computed by solving (14) instead). The balancing transformations are then found from the low-rank Gramians, and the L_2 errors of the resulting models are plotted in Fig. 7. One sees that the errors are significantly increased. It is interesting that if only the controllability Gramian is reduced to rank 30, while the exact observability Gramian is retained, then the results are similar to full balanced truncation or balanced POD (though these results are not shown in the figure). Thus, in truncating the observability Gramian, one is removing states that

are almost unobservable, but apparently strongly controllable, and this causes increased errors in the resulting models. This illustrates one of the advantages of our method of snapshots (Sec. 3.1), which does not require separate reduction of the Gramians.

The differences between balanced truncation and POD become even more apparent when one

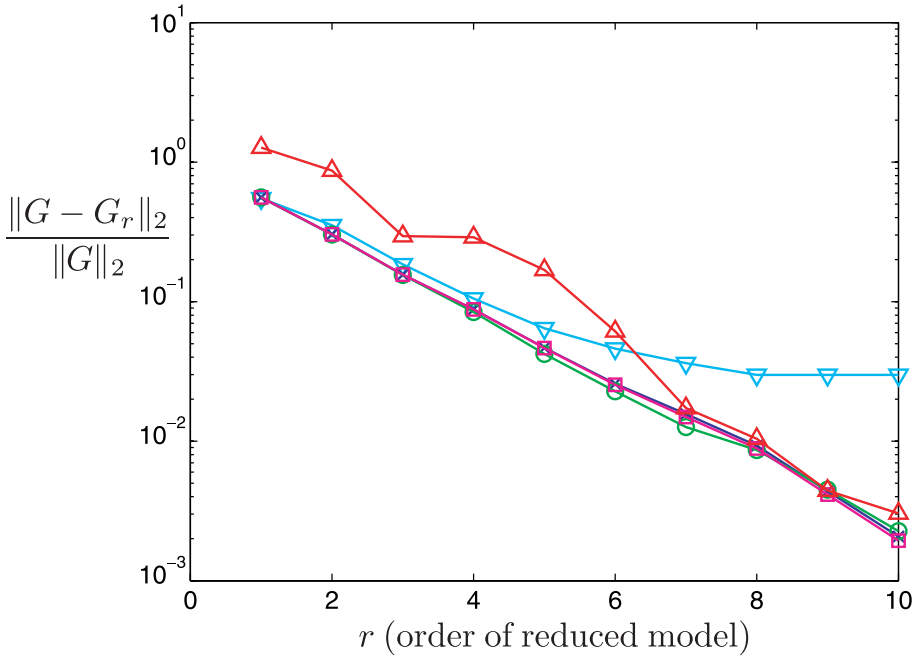

Fig. 7. Error $\|G - G_r\|_2/\|G\|_2$, for balanced truncation (×), balanced POD with five-mode and ten-mode output projection (○ and □), POD (△), and approximate balanced truncation with separate reduction of Gramians to rank 30 (▽).

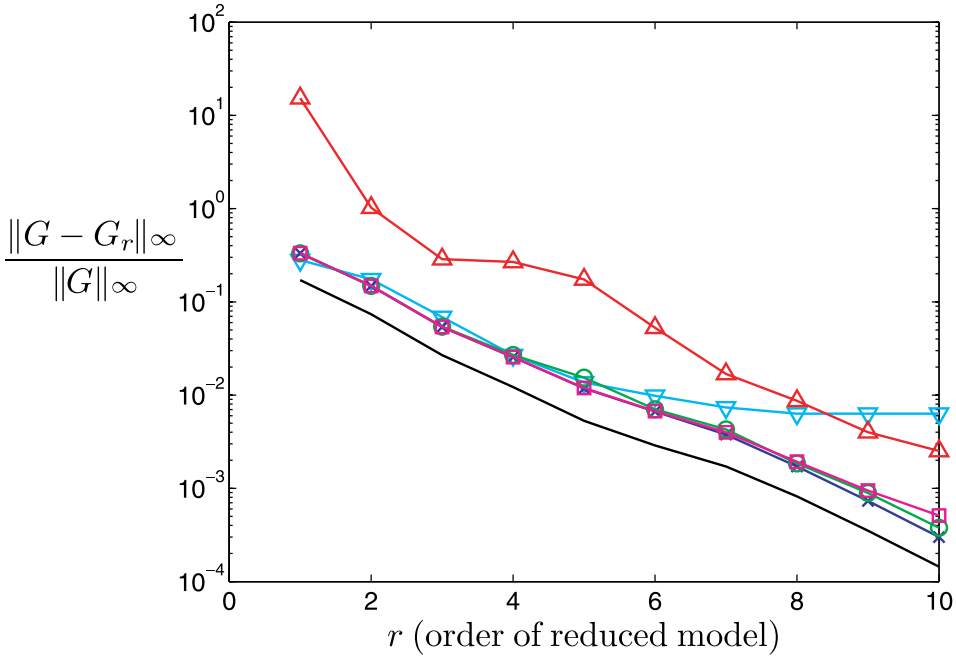

Fig. 8. Error $\|G-G_r\|_\infty/\|G\|_\infty$, for balanced truncation (×), balanced POD with five-mode and ten-mode output projection (○ and □), POD (△) approximate balanced truncation with separate reduction of Gramians to rank 30 (▽), and lower bound for any model reduction scheme (—).

considers the H_∞ norm $\|G-G_r\|_\infty$, defined by (16). This norm is perhaps the most useful, because it is an induced norm, and measures the maximum error over all possible inputs, not just an impulsive input. Figure 8 shows the error $\|G - G_r\|_\infty$ for the various reduced-order models G_r. Again, the norms for balanced POD are almost identical to the norms for exact balanced truncation, for both five-mode and ten-mode output projections. Here, the norms for POD are about an order of magnitude higher, for all models considered. The error from an approximate balanced truncation using a rank-30 reduction of the exact Gramians is also shown, and again results in larger errors for the more accurate models. Also shown in this figure is the lower bound (17) achievable by any reduced-order model of dimension r, and the balanced POD norms are indeed very close to this lower bound.

5. Conclusions

The balanced POD method described here is not the first to use empirical Gramians to compute approximate balanced truncations using simulation data. These empirical Gramians were used by Moore [1981] in his original development of balancing, and by others in extending balancing to nonlinear systems [Lall *et al.*, 1999, 2002], and computing balancing transformations for large systems [Willcox & Peraire, 2002].

This work addresses computing balancing transformations (or approximations of them) for very large systems with, e.g. millions of states, as arise in discretizations of problems in fluids. Standard methods for computing balanced truncations involve singular value decompositions of the empirical Gramians, which are full $n \times n$ matrices (where n is the number of states), which is not feasible when n is large. Previous computational methods for large systems [Willcox & Peraire, 2002] involve separate reduction of the Gramians, which can lead to less accurate models, as we have seen (Figs. 7 and 8). The method of snapshots described in Sec. 3.1 allows computing balanced truncations from SVDs of much smaller matrices, with dimension $N_p \times N_d$, where N_p and N_d are numbers of snapshots in a dataset of primal and dual solutions, respectively, without separate reduction of the Gramians.

Furthermore, previous methods as in [Lall *et al.*, 1999] and [Willcox & Periare, 2002] are not tractable for systems with large numbers of outputs, as one must integrate an adjoint solution for each output. Section 3.2 describes an output projection method that approximates full balanced truncation with guaranteed error bounds, and dramatically reduces the number of adjoint solutions necessary. In the example shown, integration of five adjoint solutions produced models that were virtually indistinguishable in the H_∞ norm from full balanced truncations, which would have required 480 adjoint simulations using previous methods.

The formulation of balanced POD also clarifies some connections between balanced truncation and POD, most importantly that for a linear system, balanced truncation is a special case of POD. In particular, one uses a dataset consisting of responses to unit impulses (one for each input), and uses the observability Gramian for the inner product. This inner product weight states of large "dynamical importance," as opposed to POD, which retains only the most energetic modes. This suggests that even for a nonlinear system, the observability Gramian from a linearization might be a good choice of inner product for POD, if reduced-order models are desired. The balanced POD procedure not only removes subjectivity in the choice of inner product for POD, but also guarantees that a Galerkin projection of a nonlinear system with a stable equilibrium point at the origin will also have a stable equilibrium point at the origin.

Although many of the developments in this paper are restricted to stable, linear systems, Sec. 3.4 suggests how many of these ideas might be extended to large-scale nonlinear systems as well, following the approaches in [Lall *et al.*, 2002].

Acknowledgments

This work was partially supported by the NSF, grant CMS-0347239, under program manager M. Tomizuka; and by AFOSR, grant F49620-03-1-0081, under program managers B. King, S. Heise and J. Schmisseur.

References

Antoulas, A. C., Sorensen, D. C. & Gugercin, S. [2001] "A survey of model reduction methods for large-scale systems," *Contemp. Math.* **280**, 193–219.

Bamieh, B. & Daleh, M. [2001] "Energy amplification in channel flows with stochastic excitation," *Phys. Fluids* **13**, 3258–3269.

Colonius, T. & Freund, J. B. [2002] "POD analysis of sound generation by a turbulent jet," AIAA Paper 2002-0072.

Cortelezzi, L. & Speyer, J. L. [1998] "Robust reduced-order controller of laminar boundary layer transitions," *Phys. Rev.* **E58**, 1906–1910.
Drazin, P. G. & Reid, W. H. [1981] *Hydrodynamic Stability* (Cambridge University Press).
Dullerud, G. E. & Paganini, F. [1999] *A Course in Robust Control Theory: A Convex Approach*, Texts in Applied Mathematics, Vol. 36 (Springer-Verlag).
Farrell, B. F. & Ioannou, P. J. [1993] "Stochastic forcing of the linearized Navier–Stokes equations," *Phys. Fluids* **A5**, 2600–2609.
Holmes, P., Lumley, J. L. & Berkooz, G. [1996] *Turbulence, Coherent Structures, Dynamical Systems and Symmetry* (Cambridge University Press).
Lall, S., Marsden, J. E. & Glavaški, S. [1999] "Empirical model reduction of controlled nonlinear systems," in *Proc. IFAC World Congress*, Vol. F, pp. 473–478.
Lall, S., Marsden, J. E. & Glavaški, S. [2002] "A subspace approach to balanced truncation for model reduction of nonlinear control systems," *Int. J. Robust Nonlin. Contr.* **12**, 519–535.
Laub, A. J., Heath, M. T., Page, C. C. & Ward, R. C. [1987] "Computation of balancing transformations and other applications of simultaneous diagonalization algorithms," *IEEE Trans. Automat. Contr.* **32**, 115–122.
Lumley, J. L. [1970] *Stochastic Tools in Turbulence* (Academic Press).
Moore, B. C. [1981] "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Automat. Contr.* **26**, 17–32.
Rathinam, M. & Petzold, L. R. [2003] "A new look at proper orthogonal decomposition," *SIAM J. Numer. Anal.* **41**, 1893–1925.
Rowley, C. W., Colonius, T. & Murray, R. M. [2004] "Model reduction for compressible flow using POD and Galerkin projection," *Physica* **D189**, 115–129.
Scherpen, J. M. A. [1993] "Balancing for nonlinear systems," *Syst. Contr. Lett.* **21**, 143–153.
Sirovich, L. [1987] "Turbulence and the dynamics of coherent structures, parts I–III," *Q. Appl. Math.* **XLV**, 561–590.
Smith, T. R. [2003] "Low-dimensional models of plane Couette flow using the proper orthogonal decomposition," PhD thesis, Princeton University.
Willcox, K. & Peraire, J. [2002] "Balanced model reduction via the proper orthogonal decomposition," *AIAA J.* **40**, 2323–2330.

Appendix A
Theorems on Computing Balancing Transformations

Here, we consider empirical Gramians defined by (22), with balancing transformations T_1 and S_1 defined by (23)–(24). The following theorem establishes that if one takes enough snapshots that the empirical Gramians W_c and W_o have full rank n (clearly, at least n snapshots are required, and the system must be both controllable and observable), then Σ_1 contains the Hankel singular values (square roots of the eigenvalues of the product W_cW_o), and T_1 is the balancing transformation that simultaneously diagonalizes W_c and W_o.

Proposition 1. *Let W_c and W_o be empirical Gramians defined by (22), and suppose Y^*X has rank $r = n$. Then the matrix T_1 is square and invertible, with inverse S_1, and*

$$S_1 W_c S_1^* = T_1^* W_o T_1 = \Sigma_1.$$

Proof. To show $S_1 = T_1^{-1}$, we have

$$\begin{aligned} S_1 T_1 &= \Sigma_1^{-1/2} U_1^* Y^* X V_1 \Sigma_1^{-1/2} \\ &= \Sigma_1^{-1/2} \Sigma_1 \Sigma_1^{-1/2} = I_n. \end{aligned}$$

Also,

$$\begin{aligned} S_1 W_c S_1^* &= \Sigma_1^{-1/2} U_1^* Y^* X X^* Y U_1 \Sigma^{-1/2} \\ &= \Sigma_1^{-1/2} (\Sigma_1 V_1^*)(V_1 \Sigma_1) \Sigma_1^{-1/2} = \Sigma_1, \end{aligned}$$

and a similar calculation shows $T_1^* W_o T_1 = \Sigma_1$. ■

Of course, our main interest is in large systems for which the number of snapshots, and hence the rank of W_c, W_o is much smaller than n. The following theorem establishes that in this case, Σ_1 also contains all nonzero Hankel singular values, and T_1 contains the first r columns of the balancing transformation.

Proposition 2. *Suppose Y^*X has rank $r < n$. Then there exist matrices $S_2, T_2 \in \mathbb{R}^{n \times (n-r)}$ such that for*

$$T = [T_1 \quad T_2], \quad S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix},$$

T is invertible with $T^{-1} = S$, and

$$S W_c W_o T = \begin{bmatrix} \Sigma_1^2 & 0 \\ 0 & 0 \end{bmatrix}, \tag{A.1}$$

and furthermore,

$$SW_cS^* = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & M_1 \end{bmatrix}, \qquad T^*W_oT = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & M_2 \end{bmatrix}, \tag{A.2}$$

where M_1 and M_2 are matrices in $\mathbb{R}^{(n-r)\times(n-r)}$.

Proof. As in the proof of Theorem 1, $S_1T_1 = I_r$. Choose T_2 such that its columns form a basis for the nullspace of S_1 (an $(n-r)$-dimensional subspace of $\mathbb{R}^n$). Then $S_1T_2 = 0$, and T is invertible, since its columns are linearly independent. Define S_2 as the last $n-r$ rows of T^{-1}, and it follows that $S_2T_1 = 0$.

First, we show

$$T^*W_oT = \begin{bmatrix} T_1^*W_oT_1 & T_1^*W_oT_2 \\ T_2^*W_oT_1 & T_2^*W_oT_2 \end{bmatrix} = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & M_2 \end{bmatrix}.$$

As in the proof of Theorem 1, $T_1^*W_oT_1 = \Sigma_1$. Next,

$$\begin{aligned} T_1^*W_oT_2 &= \Sigma_1^{-1/2}V_1^*X^*YY^*T_2 \\ &= \Sigma_1^{-1/2}(\Sigma_1U_1^*)Y^*T_2 = \Sigma_1S_1T_2 = 0_{r\times(n-r)}, \end{aligned}$$

and thus $T_2^*W_oT_1 = (T_1^*W_oT_2)^* = 0_{(n-r)\times r}$. The results for SW_cS^* and SW_cW_oT follow similarly, using $S_2T_1 = 0$. ■