23 Jun 2007

Balanced Proper Orthogonal Decomposition for Model Reduction of Infinite Dimensional Linear Systems

John R. Singler
Missouri University of Science and Technology, singlerj@mst.edu

Belinda A. Batten

Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2007 Chicago, 20–23 June 2007.

Balanced Proper Orthogonal Decomposition for Model Reduction of Infinite Dimensional Linear Systems

John R. Singler[1] **and Belinda A. Batten**[1]

[1] *Department of Mechanical Engineering, Oregon State University*

emails: John.Singler@oregonstate.edu, bbatten@engr.orst.edu

Abstract

In this paper, we extend a method for reduced order model derivation for finite dimensional systems developed by Rowley to infinite dimensional systems. The method is related to standard balanced truncation, but includes aspects of the proper orthogonal decomposition in its computational approach. The method is also applicable to nonlinear systems. The method is applied to a convection diffusion equation.

Key words: balanced truncation, proper orthogonal decomposition, infinite dimensional systems

1 Introduction and Overview

In this work, we formally extend Rowley's balanced POD algorithm [8] to the infinite dimensional case. The resulting algorithm is a POD-type procedure to design an approximate balanced transformation of an infinite dimensional linear system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \quad x(0) = x_0, \\ y(t) &= Cx(t), \end{aligned} \tag{1}$$

over a Hilbert space X with inner product $(\cdot, \cdot)$. We assume the linear operator $A : D(A) \subset X \to X$ generates an exponentially stable C_0-semigroup e^{At}, and the operators $B : U \to X$ and $C : X \to Y$ are bounded and finite rank. We also assume the input and output spaces are finite dimensional; specifically $U = \mathbb{R}^m$ and $Y = \mathbb{R}^p$.

Model reduction via balanced truncation is performed by first determining a balanced realization in which the controllable and observable states of (1) coincide. Then, the balanced model is truncated based on the eigenvalues of the product of the observability and controllability Gramians by eliminating the states corresponding to modes

that are difficult to control and observe. Specifically, define the controllability and observability operators $\mathcal{B} : L^2(0,\infty;U) \to X$ and $\mathcal{C} : X \to L^2(0,\infty;Y)$ by

$$\mathcal{B}u = \int_0^\infty e^{At}Bu(t)\,dt, \quad [\mathcal{C}x](t) = Ce^{At}x.$$

The adjoint operators $\mathcal{B}^* : X \to L^2(0,\infty;U)$ and $\mathcal{C}^* : L^2(0,\infty;Y) \to X$ are given by

$$[\mathcal{B}^*x](t) = B^*e^{A^*t}, \quad \mathcal{C}^*y = \int_0^\infty e^{A^*t}C^*y(t)\,dt.$$

The controllability and observability Gramians, $L_B \in \mathcal{L}(X)$ and $L_C \in \mathcal{L}(X)$, are defined by

$$L_Bx = \mathcal{B}\mathcal{B}^*x = \int_0^\infty e^{At}BB^*e^{A^*t}x\,dt, \quad L_Cx = \mathcal{C}^*\mathcal{C}x = \int_0^\infty e^{A^*t}C^*Ce^{At}x\,dt.$$

The eigenvalues of $L_CL_B \in \mathcal{L}(X)$ are equal to the squares of the singular values of the Hankel operator $\mathcal{H} : L^2(0,\infty;U) \to L^2(0,\infty;Y)$ defined by

$$[\mathcal{H}u](t) = [\mathcal{C}\mathcal{B}u](t) = \int_0^\infty Ce^{A(t+s)}Bu(s)\,ds.$$

An important fact is that that the Hankel singular values are independent of the chosen coordinate system, or system realization.

The coordinate change that balances the system—the balancing transformation—produces observability and controllability Gramians that are equal and diagonal. In the infinite dimensional setting, the Gramians are equal to a diagonal operator on ℓ^2, the space of square summable sequences; see [5, 7] and the review in [4]. The Hankel singular values are then ordered from greatest to least, and the states corresponding to the "small" singular values are truncated to produce a low order model. This method is rather standard and known in the literature (see, e.g., [9]).

The balanced POD algorithm determines a truncated approximate balancing transformation $T_r : \mathbb{R}^r \to X$ and its left inverse $S_r : X \to \mathbb{R}^r$ (i.e., $S_rT_r = I_r$). To obtain a low order model, approximate the solution $x(t)$ of the linear system (1) by Galerkin projection as

$$x(t) \approx x_r(t) = T_rS_rx(t) = T_ra(t), \quad \text{where} \quad a(t) = S_rx(t). \tag{2}$$

Substituting this approximate solution into the linear system yields the reduced order model

$$\begin{aligned} \dot{a}(t) &= A_ra(t) + B_ru(t), \quad a(0) = a_0, \\ y(t) &= C_ra(t), \end{aligned} \tag{3}$$

where $A_r = S_rAT_r$, $B_r = S_rB$, $C_r = CT_r$, and $a_0 = S_rx_0$.

We may apply this Galerkin projection to obtain low order models of more general, in fact nonlinear, systems. For example, suppose the model takes the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + F(x(t)) + Bu(t) + Dw(t), \quad x(0) = x_0, \\ y(t) &= Cx(t) + Ew(t), \end{aligned} \tag{4}$$

where F is a nonlinear operator and w is a disturbance. Design the approximate balancing transformation about the linearized system and use the approximation for the solution (2) to obtain the model

$$\begin{aligned} \dot{a}(t) &= A_r a(t) + F_r(a(t)) + B_r u(t) + D_r w(t), \quad a(0) = a_0, \\ y(t) &= C_r a(t) + E_r w(t), \end{aligned} \tag{5}$$

where A_r, B_r, C_r, and a_0 are as above, $D_r = S_r D$, $E_r = E$, and $F_r(a) = S_r F(T_r a)$.

2 Formal Derivation of the Algorithm

We now give a formal derivation of the balanced POD algorithm for the infinite dimensional setting described above. We do not attempt to rigorously justify the derivation; in some cases we simply proceed by analogy with the finite dimensional case. Convergence analysis of the algorithm is left for future work.

The complete algorithm is presented in Section 3 below. One possible numerical implementation of the algorithm is given in Section 3.1.

2.1 Special Forms of the Gramians

One of the main components of the balanced POD algorithm is to compute approximate factors of the Gramians using simulation data. This is possible because of the special form of the Gramians.

Given the specific assumptions regarding the input and output operators, B and C, in Section 1, we can write them in the form

$$Bu = \sum_{j=1}^{m} b_j u_j, \quad Cx = [\,(c_1, x),\, \ldots\, , (c_p, x)\,]^T,$$

where $u = [\,u_1,\, \ldots\, , u_m\,]^T \in U$, and each b_j and c_j are in X.

This allows us to rewrite the Gramians. First, define the functions $w_j(t) = e^{At} b_j$, for $j = 1, \ldots, m$. Then w_j is the solution of the evolution equation

$$\dot{w}_j(t) = A w_j(t), \quad w_j(0) = b_j.$$

The controllability operator $\mathcal{B} : L^2(0, \infty; U) \to X$ defined above takes the form

$$\mathcal{B}u = \int_0^\infty e^{At} B u(t)\, dt = \int_0^\infty \sum_{j=1}^{m} w_j(t) u_j(t)\, dt,$$

and its adjoint operator $\mathcal{B}^* : X \to L^2(0, \infty; U)$ is easily computed to be

$$[\mathcal{B}^* x](t) = [\,(w_1(t), x),\, \ldots\, , (w_m(t), x)\,]^T.$$

Therefore, the controllability Gramian $L_B = \mathcal{B}\mathcal{B}^* \in \mathcal{L}(X)$ is given by

$$L_B x = \int_0^\infty \sum_{j=1}^{m} w_j(t)(w_j(t), x)\, dt.$$

To treat the observability Gramian, we need the adjoint operator $C^* \in \mathcal{L}(Y, X)$ given by

$$C^* y = \sum_{j=1}^{p} c_j y_j,$$

where $y = [\, y_1, \ldots, y_p \,]^T \in Y$. We follow a similar procedure as used for B and define $z_j(t) = e^{A^* t} c_j$, for $j = 1, \ldots, p$. Then z_j is the solution of the adjoint equation

$$\dot{z}_j(t) = A^* z_j(t), \quad z_j(0) = c_j.$$

The adjoint of the observability operator $\mathcal{C}^* : L^2(0, \infty; Y) \to X$ takes the form

$$\mathcal{C}^* y = \int_0^\infty e^{A^* t} C^* y(t)\, dt = \int_0^\infty \sum_{j=1}^{p} z_j(t) y_j(t)\, dt$$

and the operator $\mathcal{C} : X \to L^2(0, \infty; Y)$ is given by $[\mathcal{C}x](t) = [\, (z_1(t), x), \ldots, (z_p(t), x) \,]^T$. Therefore, the observability Gramian $L_C = \mathcal{C}^* \mathcal{C} \in \mathcal{L}(X)$ is

$$L_C x = \int_0^\infty \sum_{j=1}^{p} z_j(t)(z_j(t), x)\, dt.$$

2.2 The Empirical Gramians

The Gramians can be approximated using time snapshots of the states $w_i(t)$ and $z_i(t)$. Specifically, we approximate the time integrals with the quadratures

$$\begin{aligned} L_B x = \int_0^\infty \sum_{i=1}^{m} w_i(t)(w_i(t), x)\, dt \qquad &\approx \qquad L_B^{n_1} x = \sum_{i=1}^{m} \sum_{j=1}^{n_1} \alpha_j^2 w_i(t_j)(w_i(t_j), x), \\ L_C x = \int_0^\infty \sum_{i=1}^{p} z_i(t)(z_i(t), x)\, dt \qquad &\approx \qquad L_C^{n_2} x = \sum_{i=1}^{p} \sum_{k=1}^{n_2} \beta_k^2 z_i(t_k)(z_i(t_k), x). \end{aligned}$$

Here, $\{\alpha_j^2\}$ and $\{\beta_k^2\}$ are quadrature weights corresponding to the sets of quadrature points $\{t_j\}$ and $\{t_k\}$; different quadrature points and weights can be used for each w_i and z_i if desired. Since w_i are z_i are solutions to linear evolution equations, they are continuous in time and therefore have a well defined value at the quadrature points. The approximate Gramians $L_B^{n_1} \in \mathcal{L}(X)$ and $L_C^{n_2} \in \mathcal{L}(X)$ are called *empirical Gramians*.

Following Rowley in the finite dimensional case, we factor the empirical Gramians. Define "vectors" of weighted snapshots

$$\tilde{w} = [\, \alpha_1 w_1(t_1), \ldots, \alpha_{n_1} w_1(t_{n_1}), \ldots, \alpha_1 w_m(t_1), \ldots, \alpha_{n_1} w_m(t_{n_1}) \,]^T \in X^{N_1}, \quad (6)$$

$$\tilde{z} = [\, \beta_1 z_1(t_1), \ldots, \beta_{n_2} z_1(t_{n_2}), \ldots, \beta_1 z_p(t_1), \ldots, \beta_{n_2} z_p(t_{n_2}) \,]^T \in X^{N_2}, \quad (7)$$

where $N_1 = mn_1$, $N_2 = pn_2$, and $X^q = X \times \cdots \times X$ (q times). These vectors allow the empirical Gramians to be written as $L_B^{n_1} = PP^*$ and $L_C^{n_2} = Q^* Q$, where the operators

$P : \mathbb{R}^{N_1} \to X$ and $Q : X \to \mathbb{R}^{N_2}$ are defined by

$$Pa = \sum_{i=1}^{N_1} a_i \tilde{w}_i, \quad Qx = [\,(\tilde{z}_1, x), \ldots, (\tilde{z}_{N_2}, x)\,]^T,$$

and their adjoint operators $P^* : X \to \mathbb{R}^{N_1}$ and $Q^* : \mathbb{R}^{N_2} \to X$ are given by

$$P^*x = [\,(\tilde{w}_1, x), \ldots, (\tilde{w}_{N_1}, x)\,]^T, \quad Q^*a = \sum_{i=1}^{N_2} a_i \tilde{z}_i.$$

Note that P and Q and their adjoints depend on the quadrature points and weights; however, we suppress this dependence for notational simplicity.

2.3 The Approximate Balanced Transformation

Recall that the eigenvalues of the product of the Gramians can be used to compute a balancing transformation for the linear system. The balanced system is then truncated to form a reduced order model. We approximate the product of the Gramians $L = L_C L_B$ using the empirical Gramians, i.e., $L \approx L^n = L_C^{n_2} L_B^{n_1}$. Using the above factors, we have $L^n = Q^*QPP^*$. Following Curtain and Zwart ([6, Lemma 8.2.9, pages 401–402]), it is easy to show that L^n is compact and that the nonzero eigenvalues of L^n are equal to the squares of the nonzero singular values of QP.

The operator QP is a bounded linear mapping from $\mathbb{R}^{N_1}$ to $\mathbb{R}^{N_2}$; therefore, it can be represented as an $N_2 \times N_1$ matrix Γ with entries $\Gamma_{ij} = (\tilde{z}_i, \tilde{w}_j)$. Let the singular value decomposition of Γ be given by

$$\Gamma = U\Sigma V^* = [U_1\ U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} = U_1 \Sigma_1 V_1^*, \tag{8}$$

where $\Sigma_1 \in \mathbb{R}^{s \times s}$ is diagonal and invertible, $s = \operatorname{rank}(\Gamma)$, $U_1^* U_1 = I_s = V_1^* V_1$, and I_s is the identity matrix in $\mathbb{R}^{s \times s}$.

In the finite dimensional case, Rowley showed that an approximate balancing transformation is given by the operators $T_1 : \mathbb{R}^s \to X$ and $S_1 : X \to \mathbb{R}^s$ defined by

$$T_1 = PV_1\Sigma_1^{-1/2}, \quad S_1 = \Sigma_1^{-1/2} U_1^* Q.$$

In this paper, we assume the same is true for the infinite dimensional setting and leave theoretical analysis of the algorithm for future work due to size restrictions.

The operators $T_1 : \mathbb{R}^s \to X$ and $S_1 : X \to \mathbb{R}^s$ have the representations

$$T_1 a = \sum_{j=1}^{s} a_j \varphi_j, \quad S_1 x = [\,(\psi_1, x), \ldots, (\psi_s, x)\,]^T,$$

where the (primary) balanced POD modes $\{\varphi_i\}$ and the adjoint balanced POD modes $\{\psi_i\}$ are given by

$$[\,\varphi_1, \ldots, \varphi_s\,]^T = \Sigma_1^{-1/2} V_1^* \tilde{w}, \qquad [\,\psi_1, \ldots, \psi_s\,]^T = \Sigma_1^{-1/2} U_1^* \tilde{z}.$$

As in the finite dimensional case, the primary and adjoint balanced POD modes are biorthogonal, i.e., $(\psi_i, \varphi_j) = \delta_{ij}$. To see this, note $S_1 T_1 a = [\,(\psi_i, \varphi_j)\,]a$ for any $a \in \mathbb{R}^s$. Also, by definition,

$$S_1 T_1 a = \Sigma_1^{-1/2} U_1^* Q P V_1 \Sigma_1^{-1/2} a = I_s a.$$

Thus, $[\,(\psi_i, \varphi_j)\,] = I_s$, or $(\psi_i, \varphi_j) = \delta_{ij}$.

The approximate balancing transformations are truncated by picking $r < s$ and setting

$$T_r a = \sum_{j=1}^{r} a_j \varphi_j, \quad S_r x = [\,(\psi_1, x), \ldots, (\psi_r, x)\,]^T.$$

Thus, only the first r primary and adjoint balanced POD modes need to be computed. Also, we have $S_r T_r = I_r$, and the modes can be computed by

$$[\,\varphi_1, \ldots, \varphi_r\,]^T = \Sigma_r^{-1/2} V_r^* \tilde{w}, \qquad [\,\psi_1, \ldots, \psi_r\,]^T = \Sigma_r^{-1/2} U_r^* \tilde{z}, \tag{9}$$

where Σ_r, U_r, and V_r are appropriate truncations of Σ_1, U_1, and V_1.

3 The Balanced POD Algorithm

The construction of the operators T_r and S_r as shown above completes the balanced POD algorithm. As outlined in Section 1, we use these transformation to obtain the reduced order model (3). The complete procedure can be summarized as follows:

1. Approximate the solutions w_j of the differential equations

   $$\dot{w}_j(t) = A w_j(t), \quad w_j(0) = b_j, \tag{10}$$

   for $j = 1, \ldots, m$, where $Bu = \sum_{j=1}^{m} b_j u_j$.

2. Approximate the solutions z_j of the adjoint differential equations

   $$\dot{z}_j(t) = A^* z_j(t), \quad z_j(0) = c_j, \tag{11}$$

   for $j = 1, \ldots, p$, where $Cx = [\,(c_1, x), \ldots, (c_p, x)\,]^T$.

3. Form the matrix Γ, where $\Gamma_{ij} = (\tilde{z}_i, \tilde{w}_j)$, and the weighted snapshot vectors $\tilde{w}$ and $\tilde{z}$ defined in (6) and (7), respectively.

4. Compute the singular value decomposition of Γ as in (8), choose $r < \mathrm{rank}(\Gamma)$, and form the first r primary and adjoint balanced POD modes defined in (9):

   $$[\,\varphi_1, \ldots, \varphi_r\,]^T = \Sigma_r^{-1/2} V_r^* \tilde{w}, \qquad [\,\psi_1, \ldots, \psi_r\,]^T = \Sigma_r^{-1/2} U_r^* \tilde{z},$$

   where Σ_r, U_r, and V_r are appropriate truncations of Σ_1, U_1, and V_1.

5. Use the modes to form the matrices in the reduced order model (3):

   $$\begin{array}{rcccl} A_r & = & S_r A T_r & = & [\,(A\varphi_j, \psi_i)\,] \in \mathbb{R}^{r \times r}, \\ B_r & = & S_r B & = & [\,(b_j, \psi_i)\,] \in \mathbb{R}^{r \times m}, \\ C_r & = & C T_r & = & [\,(\varphi_j, c_i)\,] \in \mathbb{R}^{p \times r}, \\ a_0 & = & S_r x_0 & = & [\,(x_0, \psi_1), \ldots, (x_0, \psi_r)\,]^T \in \mathbb{R}^r. \end{array} \tag{12}$$

3.1 Finite Dimensional Galerkin Approximations

The algorithm presented above is flexible since we may use any procedure to approximate the solutions w_i and z_i of the linear differential equations (10) and (11). We describe the balanced POD algorithm with Galerkin approximations.

Let $W_1 = \text{span}\{\xi_j\}_{j=1}^k \subset D(A)$ and $W_2 = \text{span}\{\eta_j\}_{j=1}^\ell \subset D(A^*)$ be finite dimensional subsets of X. We compute the solutions of the primary and adjoint differential equations by the finite dimensional Galerkin approximations

$$w_\alpha(t) \approx \sum_{j=1}^k r_{j\alpha}(t)\xi_j, \quad z_\beta(t) \approx \sum_{j=1}^\ell s_{j\beta}(t)\eta_j,$$

for $\alpha = 1, \ldots, m$ and $\beta = 1, \ldots, p$. Here, k is the same for each α and ℓ is the same for each β; this is not necessary in general, but it does simplify the resulting algorithm. Using these Galerkin approximations, the balanced POD algorithm becomes:

1. Form the $k \times k$ matrices $\tilde{M}_k = [\,(\xi_j, \xi_i)\,]$ and $\tilde{A}_k = [\,(A\xi_j, \xi_i)\,]$. Approximate the Galerkin coefficient vectors $r_\alpha = [\,r_{1\alpha}, \ldots, r_{m\alpha}\,]^T$ by solving the equations

$$\tilde{M}_k \dot{r}_\alpha(t) = \tilde{A}_k r_\alpha(t), \quad \tilde{M}_k r_\alpha(0) = [\,(b_\alpha, \xi_i)\,], \qquad \alpha = 1, \ldots, m. \tag{13}$$

2. Form the $\ell \times \ell$ matrices $\hat{M}_\ell = [\,(\eta_j, \eta_i)\,]$ and $\hat{A}_\ell = [\,(A^*\eta_j, \eta_i)\,]$. Approximate the Galerkin coefficient vectors $s_\beta = [\,s_{1\beta}, \ldots, s_{p\beta}\,]^T$ by solving the equations

$$\hat{M}_\ell \dot{s}_\beta(t) = \hat{A}_\ell s_\beta(t), \quad \hat{M}_\ell s_\beta(0) = [\,(c_\beta, \eta_i)\,], \qquad \beta = 1, \ldots, p.$$

3. Define the weighted snapshot coefficient matrices $R \in \mathbb{R}^{N_1 \times k}$ and $S \in \mathbb{R}^{N_2 \times \ell}$ by

$$\begin{aligned} R &= [\,\alpha_1 r_1(t_1), \ldots, \alpha_{n_1} r_1(t_{n_1}), \ldots, \alpha_1 r_m(t_1), \ldots, \alpha_{n_1} r_m(t_{n_1})\,]^T, \\ S &= [\,\beta_1 s_1(t_1), \ldots, \beta_{n_2} s_1(t_{n_2}), \ldots, \beta_1 s_p(t_1), \ldots, \beta_{n_2} s_p(t_{n_2})\,]^T. \end{aligned}$$

   Then the weighted snapshot vectors $\tilde{w}$ and $\tilde{z}$ defined in (6) and (7), respectively, are approximated by

$$\tilde{w} \approx R[\,\xi_1, \ldots, \xi_k\,]^T, \quad \tilde{z} \approx S[\,\eta_1, \ldots, \eta_\ell\,]^T.$$

   Also, the matrix Γ is approximated by $\hat{\Gamma} = SNR^T$, where the $\ell \times k$ matrix N is given by $N = [(\eta_i, \xi_j)]$.

4. Compute the singular value decomposition of $\hat{\Gamma}$ as in (8) and choose $r < \text{rank}(\hat{\Gamma})$. Then the first r primary and adjoint balanced POD modes are approximated by

$$\begin{aligned} \big[\,\varphi_1, \ldots, \varphi_r\,\big]^T &\approx \Sigma_r^{-1/2} V_r^* R[\,\xi_1, \ldots, \xi_k\,]^T, \\ \big[\,\psi_1, \ldots, \psi_r\,\big]^T &\approx \Sigma_r^{-1/2} U_r^* S[\,\eta_1, \ldots, \eta_\ell\,]^T, \end{aligned}$$

   where Σ_r, U_r, and V_r are appropriate truncations of Σ_1, U_1, and V_1. Let $\Phi = \Sigma_r^{-1/2} V_r^* R \in \mathbb{R}^{r \times k}$ and $\Psi = \Sigma_r^{-1/2} U_r^* S \in \mathbb{R}^{r \times \ell}$. Then for each i,

$$\varphi_i \approx \sum_{j=1}^k \Phi_{ij}\xi_j, \quad \psi_i \approx \sum_{j=1}^\ell \Psi_{ij}\eta_j.$$

5. Substitute the approximate modes into the reduced order model matrices (12):

$$\begin{aligned} A_r &= [\,(A\varphi_j, \psi_i)\,] \approx \Psi[\,(A\xi_j, \eta_i)\,]\Phi^T, \\ B_r &= [\,(b_j, \psi_i)\,] \approx \Psi[\,(b_j, \eta_i)\,], \\ C_r &= [\,(\varphi_j, c_i)\,] \approx [\,(\xi_j, c_i)\,]\Phi^T, \\ a_0 &= [\,(x_0, \psi_1), \ldots, (x_0, \psi_r)\,]^T \approx \Psi[\,(x_0, \eta_1), \ldots, (x_0, \eta_\ell)\,]^T. \end{aligned}$$

3.2 Comparison to the Finite Dimensional Algorithm

The Galerkin method presented above gives one way to compare the infinite dimensional balanced POD algorithm presented here which we term "balance POD then discretize" with the finite dimensional POD algorithm applied to a discretization of an infinite dimensional system which we call "discretize then balance POD".

In the "discretize then balance POD" approach, one applies the Galerkin method (or some other discretization scheme) to the linear system (1) to obtain the ordinary differential equation system (13) in step 1 above along with the finite dimensional output equation $y_k = \tilde{C}_k r_\alpha$, where $\tilde{C}_k = [\,(\xi_j, c_i)\,]$. Finite dimensional balanced POD is then performed on this system to obtain a reduced order model.

If certain conditions are satisfied, the "balance POD then discretize" approach presented here produces the same reduced order model as the "discretize then balance POD" approach outlined above. It can be checked that the following conditions are sufficient:

- The Galerkin subspaces W_1 and W_2 must be equal (therefore, $k = \ell$).
- The Galerkin scheme must satisfy $\tilde{A}_k^* = \hat{A}_k$.
- The same quadrature points and weights are used.
- The inner product for the finite dimensional balanced POD must be weighted by the matrix $\tilde{M}_k$, i.e., $(a, b) = a^T \tilde{M}_k b$.

In this case, the matrix Φ^T is produced by the finite dimensional balanced POD algorithm, and the same reduced order model results from both approaches.

We note that certain problems and numerical schemes may not satisfy the first two conditions above. For example, if the domain of A does not equal the domain of A^*, the first condition may be difficult or impossible to satisfy. Also, certain Galerkin schemes may not satisfy the duality property required in the second condition; for an example with a delay equation, see [3]. In these cases, the "discretize then balance" approach may not produce an actual approximate balancing transformation.

4 Numerical Results

All numerical results in this section are for the convection diffusion equation

$$\begin{aligned} w_t(t,x) &= \mu w_{xx}(t,x) - \kappa w_x(t,x) + b(x)u(t), \\ y(t) &= \int_0^1 c(x)w(t,x)\,dx, \\ w(t,0) &= 0, \quad w(t,1) = 0, \quad w(0,x) = w_0(x). \end{aligned}$$

with $\mu = 0.1$ and $\kappa = 1$. The functions $b(x)$ and $c(x)$ are piecewise constant with $b(x) = 1$ when $0.1 < x < 0.3$, $c(x) = 1$ when $0.6 < x < 0.7$, and both are zero otherwise. The linear operators are defined as

$$Aw = \mu w_{xx} - \kappa w_x, \;\; D(A) = H^2 \cap H_0^1, \quad A^* w = \mu w_{xx} + \kappa w_x, \;\; D(A^*) = D(A).$$

The solutions of the primary and dual linear systems were approximated with standard piecewise linear finite elements using equally spaced nodes. The solutions were integrated over $0 \leq t \leq 2$ using Matlab's `ode15s` solver with default error tolerances. The quadrature points were chosen as the time points returned from `ode15s` and the trapezoid rule was used for the quadrature weights. Time refinement was performed by decreasing the error tolerances of the ODE solver.

We compare the results of the balanced POD algorithm with standard balancing computations. We focus on the Hankel singular values and the balancing modes since these are used to construct the reduced order model. For this example problem, the two approaches give identical results when refined until convergence. In the balanced POD computations, spatial refinement was more important for convergence than time refinement. This is not surprising since the solutions of the primary and dual linear systems are not highly variable in time.

In Figure 1, we show the first 20 approximate Hankel singular values for standard balancing and for balanced POD. The methods produce identical results. For each computation, we used 256 equally spaced finite element nodes. The singular values are converged — further refinement in space (and in time for balanced POD) produces little change. The remaining singular values are below machine precision. The first 5 singular values contain over 99.99% of the "energy" or information in the dataset.

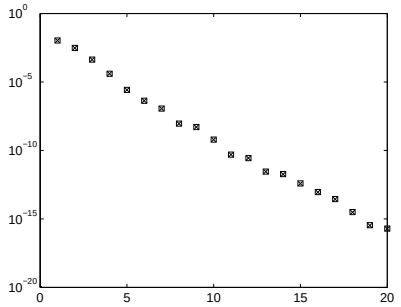

Figure 1: Approximate Hankel singular values for standard balancing (squares) and balanced POD (x).

In Figures 2 and 3, we show primary and adjoint balanced POD modes. All modes are converged and standard balancing produces identical results, as it should for this example. In general, the higher numbered modes are slower to converge under refinement with both standard balancing and balanced POD. For these computations, 128 equally spaced finite element nodes were used.

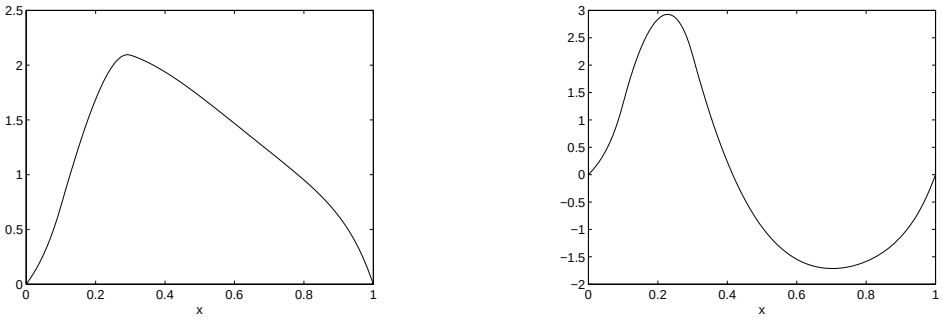

Figure 2: Balanced POD mode 1 (left) and mode 2 (right).

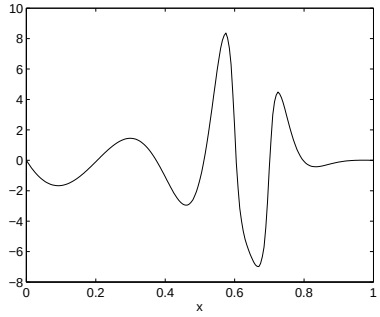

Figure 3: Fifth adjoint balanced POD mode.

5 Conclusions and Future Work

In this paper, we extended Rowley's balanced POD algorithm to infinite dimensional systems. In addition, we compared finite and infinite dimensional algorithms and gave conditions when balanced POD "commutes" with discretization. Preliminary numerical results for the convection diffusion equation indicate convergence of the algorithm by comparing the balanced POD with standard balancing computations.

This method shows promise for reduced order model design. In particular, it is computationally tractable for infinite dimensional systems, even if approximating finite dimensional systems have very high dimensions. Additionally, it is applicable even if matrices from approximating systems are not available. One only needs to be able to approximate solutions of standard and dual linear systems. Moreover, there is potential to use error estimators for the solutions of the linear equations to show where to refine to improve accuracy.

We point out, however, that balanced POD may not be feasible for: 1) systems with solutions that decay slowly to zero or are highly oscillatory in time because they may need a large number of time quadrature points, or 2) systems that have a large

number of inputs.

In a future paper, we will complete the convergence analysis of this method. In addition, we will compare this approach with balanced truncation methods using large scale matrix Lyapunov solvers (see [1, 2] and the references therein). Even in the case that matrix solvers perform better, balanced POD may still be preferable due to the advantages listed above. Future work includes extending this approach to systems with unbounded input and output operators.

References

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, PA, 2005.

[2] P. Benner, V. Mehrmann, and D. C. Sorensen, eds., *Dimension Reduction of Large-Scale Systems*, Springer-Verlag, Berlin, 2005.

[3] J. Borggaard, J. A. Burns, E. Vugrin, and L. Zietsman, *On strong convergence of feedback operators for non-normal distributed parameter systems*, in Proceedings of the IEEE Conference on Decision and Control, vol. 2, 2004, pp. 1526 – 1531.

[4] R. F. Curtain, *Model reduction for control design for distributed parameter systems*, in Research Directions in Distributed Parameter Systems, SIAM, Philadelphia, PA, 2003, pp. 95–121.

[5] R. F. Curtain and K. Glover, *Balanced realisations for infinite-dimensional systems*, in Operator Theory and Systems (Amsterdam, 1985), Birkhäuser, Basel, 1986, pp. 87–104.

[6] R. F. Curtain and H. J. Zwart, *An Introduction to Infinite-Dimensional Linear System Theory*, Springer-Verlag, New York, 1995.

[7] K. Glover, R. F. Curtain, and J. R. Partington, *Realization and approximation of linear infinite-dimensional systems with error bounds*, SIAM Journal on Control and Optimization, 26 (1988), pp. 863–898.

[8] C. W. Rowley, *Model reduction for fluids, using balanced proper orthogonal decomposition*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 15 (2005), pp. 997–1013.

[9] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, 1996.