# Fast Approximated POD for a Flat Plate Benchmark with a Time Varying Angle of Attack

Gilead Tadmor[*] and Daniel Bissex[†]

*Electrical & Computer Engineering, 440 DA, Northeastern University*

*Boston, MA 02115, USA*

Bernd R. Noack[‡]

*Inst. of Fluid Mechanics and Technical Acoustics*

*Berlin University of Technology,*

*Straße des 17. Juni, 10623 Berlin, Germany*

Marek Morzyński[§]

*Institute of Combustion Engines and Transportation,*

*Poznan University of Technology,*

*Piotrowo 3, 60-965 Poznan, Poland*

Tim Colonius[¶] and Kunihiko Taira[‖]

*Engineering and Applied Science,*

*California Institute of Technology,*

*1200 East California Boulevard, MS 104-44 Pasadena, CA 91125, USA*

**An approximate POD algorithm provides an empirical Galerkin approximation with guaranteed a priori lower bound on the required resolution. The snapshot ensemble is partitioned into several sub-ensembles. Cross correlations between these sub-ensembles are approximated in terms of a far smaller correlation matrix. Computational speedup is nearly linear in the number of partitions, up to a saturation that can be estimated a priori. The algorithm is particularly suitable for analyzing long transient trajectories of high dimensional simulations, but can be applied also for spatial partitioning and parallel processing of very high spatial dimension data. The algorithm is demonstrated using transient data from two simulations. First, a two dimensional simulation of the flow over a flat plate, as it transitions from $AOA = 30^o$ to a horizontal position and back. Second, a three dimensional simulation of a flat plate with aspect ratio two as it transitions from a horizontal position to $AOA = 30^o$.**

## I. Introduction

The Proper Orthogonal Decomposition (POD) approximation algorithm[1–3] is known under a variety of conceptually equivalent titles, including the Karhunen-Loève Decomposition,[4–6] Empirical Orthogonal Functions[7–9] and Principal Component Aanalysis,[10] all tracing back, at least, to the early 20th century.[11] POD provides a statistically optimal low dimensional subspace approximation of an ensemble variation, in the sense that the mean variance resolution is maximized, given the allowed subspace dimension. Technically, the

---

[*]Professor, ECE and Mathematics, Northeastern University, Boston, MA 02115, Senior Member, AIAA.

[†]Graduate Student, ECE, 440 DA, Northeastern University, Boston, MA 02115

[‡]Ass. Professor, Institute of Fluid Mechanics and Technical Acoustics, Berlin University of Technology, Berlin, Germany.

[§]Professor, Inst. of Combustion Engines and Transportation, Poznan University of Technology, Poland

[¶]Professor, Mechanical Engineering, Caltech, Pasadena, CA 91125, AIAA Member

[‖]Graduate Student, Mechanical Engineering, Caltech, Pasadena, CA 91125

algorithm reduces to a singular value decomposition (SVD) of a discrete or continuous integral operator,[2, 3, 9] defined in terms of the ensemble.

In distributed nonlinear dynamic systems, the algorithm aims to extract an optimal low dimensional subspace for a Galerkin projection of the original system. Examples of direct interest to the current authors include shear flows,[12] wake flows[13–15] and cavity flows.[16–18] The computational complexity of the algorithm scales as $O(\min\{M \cdot N^2, M^2 \cdot N\})$, where $N$ is the number of snapshots and $M$ is the spatial dimension. In typical fluid flow applications, $N \ll M$; with $N \propto O(10^1 - 10^2)$, the method of snapshots[1] alleviates some of the computational burden, since it is focused on a correlation matrix whose size is the length of the data trajectory or trajectories.

Model based feedback control design for fluid flows requires models covering both attractors and transient trajectories over a wide range of operating conditions and system parameters.[19–22] This requires longer data trajectories leading to a growing computational challenge and motivates a search for faster suboptimal algorithms. Approximate SVD algorithms abound, in a variety of domains,[23–30] including flow control.[21, 22, 31] Here we discuss a variant of the domain decomposition method, with options for partitioning both the domain and the data ensemble (i.e., along the time axis). Like the original POD algorithm, the algorithm provides a provable error bound and does not involve an iterative convergence. The key idea is to partition the data ensemble into a number of sub-ensembles, and to approximate the cross-sub-ensemble correlations in term of a synthesized, much smaller global ensemble. The computational complexity thus reduces at a nearly linear ratio with the number of partitions, up to a saturation limit.

The need to analyze long data trajectories is associated with the issue of variations in temporally local dominant coherent structures. As noted, that issue has been addressed by several authors, e.g., in.[19–22] In particular,[19] we note that such changes may often entail continuous deformation of a fixed (and low) number of modes, rather than a change in the topological structure of the modes, or the number of modes used. Thus, the number of modes extracted by POD - or approximate POD - analysis of a long transient trajectory is expected to far exceed the number of modes needed for a similar resolution, over each short subinterval. Following the interpolatory approach of,[32] we suggest to construct the few temporally local modes as an operating point dependent linear combination of the global mode set, obtained by applying the fast POD algorithm described in this paper. A computationally inexpensive way to achieve that goal is to apply a moving window POD to the compressed dynamics of the time coefficients of a high resolution, low dimensional Galerkin approximation.

To illustrate the advantages and computational savings provided by the algorithm, the method is applied to transient data trajectories of two and three dimensional flat plate simulations. The simulations capture transients through the bifurcation point, as the angle of attack transitions between a horizontal position, with a fully attached flow, and a high angle of attack, with separated flow and vortex shedding.

## II.   POD: A Brief Overview

Here we set the language and nomenclature used in the paper. An ensemble of *snapshots* takes values in a Hilbert space $\mathcal{H}$:

$$\mathbf{\Phi} = \{\Phi_t : t \in T\} \subset \mathcal{H}, \quad \Phi_0 = mean(\mathbf{\Phi}), \qquad \tilde{\mathbf{\Phi}} = \left\{ \tilde{\Phi}_t \doteq \Phi_t - \Phi_0, : t \in T \right\}$$

where the index set $T$ is interpreted as a time interval (or union of intervals). An example would be the velocity vector fields with $\mathcal{H} = L_2(\Omega)$, where $\Omega \subset \mathbb{R}^2$ or $\subset \mathbb{R}^3$ is the computational domain. The inner product and norm are then

$$(\Phi, \Psi) \doteq \int_{\Omega} dV \, \Phi \cdot \Psi, \qquad \|\Phi\| \doteq \sqrt{(\Phi, \Psi)}$$

We denote $\mathcal{T} \doteq L_2(T)$, and by a dual use of notations, identify ensembles, such as $\tilde{\mathbf{\Phi}}$, with integral operators

$$\tilde{\mathbf{\Phi}} : \mathcal{T} \mapsto \mathcal{H}, \quad \tilde{\mathbf{\Phi}}\xi = \int_T dt \, \xi_t \tilde{\Phi}_t$$

In a computational context we shall refer to the discrete version where both $\Omega$ $T$ are assumed discrete domains, and $\tilde{\mathbf{\Phi}}$ is identified with (appropriately weighted) matrix multiplication.

American Institute of Aeronautics and Astronautics

A resolution level $\lambda \in (0,1)$ is prescribed and the POD approximation identifies a lowest dimension subspace $\mathcal{H}_r \subset \mathcal{H}$ for which

$$\frac{\langle \|\pi_r \tilde{\Phi}_t\|^2 \rangle}{\langle \|\tilde{\Phi}_t\|^2 \rangle} \geq \lambda \tag{1}$$

where $\pi_r$ is the orthogonal projection over $\mathcal{H}_r$ and where $\langle \ \rangle$ stands for the ensemble average. Using matrix notations, this can be written as

$$\frac{tr\, \pi_r \tilde{\boldsymbol{\Phi}} \tilde{\boldsymbol{\Phi}}' \pi_r}{tr\, \tilde{\boldsymbol{\Phi}} \tilde{\boldsymbol{\Phi}}'} \geq \lambda \tag{2}$$

where "$tr$" indicates a matrix trace and a prime indicates the adjoint operator.

The solution is formally given in terms of the singular value decomposition (SVD)

$$\tilde{\boldsymbol{\Phi}} = \mathbf{USV}' = \sum_i \sigma_i U_i V_i'$$

whereby $\mathcal{H}_r$ is

$$\mathcal{H}_r \doteq span\{U_i\}_{i=1}^{L}, \quad L \doteq \min\left\{ \tilde{L} \ : \ \frac{\sum_{i=1}^{\tilde{L}} \sigma_i^2}{\sum_{i=1}^{N} \sigma_i^2} \geq \lambda \right\}, \quad \pi_r = \sum_{i=1}^{L} U_i U_i'$$

The indirect *method of snapshots*[1] addresses the generic situation where $N \ll M$. Then the right singular vectors $V_i$ are computed first as eigenvectors of the $N \times N$ correlation matrix $\tilde{\boldsymbol{\Phi}}' \tilde{\boldsymbol{\Phi}}$.

As noted earlier, the computational complexity of the POD algorithm is $O(\min\{M \cdot N^2, M^2 \cdot N\})$. The algorithms discussed in this note alleviates this complexity through a two step procedure, whereby POD is first applied to sub-ensembles, through either temporal or spatial partitioning of the original ensemble.

## III.   Time Partition: A Shared Mean Algorithm

The technical part of the paper will be presented for computational acceleration by a partition of the ensemble index, $T = \cup_{k=1}^{K} T_k$, and a preliminary POD dimension reduction in each of the subsets

$$\boldsymbol{\Phi}_k = \{\Phi_t \ : \ t \in T_k\} \subset \mathcal{H}, \quad k = 1:K$$

For simplicity, the sub-sets will be of the same size: $N_0 \doteq size(T_k) = N/K$. The method presented here readily extend to spatial partitioning. The full paper will provide the detail. An example of a recent discussion of an iterative method based on spatial partitioning is[31] .

The procedure outlined here is practical when available RAM suffices to store the entire ensemble, $\Phi$, whereby the global average can be computed without multiple data loading from a hard disk. A lower bound $0 < \lambda < 1$ is specified for the requisite resolution, as defined in (2).
*Step 0:* Compute the global mean, $\Phi_0$ and the perturbation samples, $\tilde{\Phi}_t$, $t \in T$.
*Step 1:* Compute the SVD for each of the operators corresponding to the sub-ensembles $\tilde{\boldsymbol{\Phi}}_k$

$$\tilde{\boldsymbol{\Phi}}_k = \mathbf{U}_k \mathbf{S}_k \mathbf{V}_k' = \sum_{i=1}^{N_0} \sigma_{k,i} U_{k,i} V_{k,i}' \tag{3}$$

*Step 2:* Truncate the SVD at $N_k \leq N_0$, substituting $\tilde{\boldsymbol{\Phi}}_k$ by

$$\bar{\boldsymbol{\Phi}}_k \doteq \bar{\mathbf{U}}_k \bar{\mathbf{S}}_k \bar{\mathbf{V}}_k' \doteq \sum_{i=1}^{N_k} \sigma_{k,i} U_{k,i} V_{k,i}'$$

where

$$N_k \doteq \min\left\{ \tilde{N} \ : \ \frac{\sum_{i=1}^{\tilde{N}} \sigma_{k,i}^2}{\sum_{i=1}^{N_0} \sigma_{k,i}^2} \geq \sqrt{\lambda} \right\}, \quad \pi_{r,\,k} \doteq \sum_{i=1}^{N_k} U_{k,i} U_{k,i}'$$

Note that $\bar{\boldsymbol{\Phi}}_k = \pi_{r,\,k} \tilde{\boldsymbol{\Phi}}_k$.

American Institute of Aeronautics and Astronautics

*Step 3:* Define the alternative sub-ensembles and associated operators

$$\boldsymbol{\Psi}_k \doteq \{\sigma_{k,i} U_{k,i}\}_{i=1}^{N_k} \qquad \Leftrightarrow \qquad \boldsymbol{\Psi}_k \xi = \bar{\mathbf{U}}_k \bar{\mathbf{S}}_k \xi = \sum_{i=1}^{N_k} \sigma_{k,i} \xi_i U_{k,i}$$

Using operator notations, the selection of $N_k$ provides for

$$\begin{aligned} tr\, \boldsymbol{\Psi}_k \boldsymbol{\Psi}'_k &= tr\, \bar{\mathbf{U}}_k \bar{\mathbf{S}}_k^2 \bar{\mathbf{U}}'_k = tr\, \bar{\mathbf{U}}_k \bar{\mathbf{S}}_k \bar{\mathbf{V}}'_k \bar{\mathbf{V}}_k \bar{\mathbf{S}}_k \bar{\mathbf{U}}'_k \\ &= tr\, \bar{\bar{\boldsymbol{\Phi}}}_k \bar{\bar{\boldsymbol{\Phi}}}'_k = tr\, \pi_{r,\,k} \tilde{\boldsymbol{\Phi}}_k \tilde{\boldsymbol{\Phi}}'_k \pi_{r,\,k} \geq \sqrt{\lambda}\, tr\, \tilde{\boldsymbol{\Phi}}_k \tilde{\boldsymbol{\Phi}}'_k \end{aligned}$$

The advantage of $\boldsymbol{\Psi}_\mathbf{k}$ is that it holds only $N_k$ sample vectors, compared with $N_0 >> N_k$ vectors in $\bar{\bar{\boldsymbol{\Phi}}}_\mathbf{k}$.

*Step 4:* Define the truncated *global* set whose size is $\bar{N} \doteq \sum_{k=1}^K N_k \leq N$, $M$

$$\boldsymbol{\Psi} \doteq \cup_{k=1}^K \boldsymbol{\Psi}_\mathbf{k} = \cup_{k=1}^K \{\sigma_{k,i} U_{k,i}\}_{i=1}^{N_k}$$

and apply the POD procedure to $\boldsymbol{\Psi}$ with the resolution bound $\sqrt{\lambda}$

$$\boldsymbol{\Psi} = \bar{\mathbf{U}} \bar{\mathbf{S}} \bar{\mathbf{V}}' \doteq \sum_{i=1}^L \bar{\sigma}_i \bar{U}_i \bar{V}'_i$$

where

$$L \doteq \min \left\{ \tilde{L} : \frac{\sum_{i=1}^{\tilde{L}} \bar{\sigma}_i^2}{\sum_{i=1}^{\bar{N}} \bar{\sigma}_i^2} \geq \sqrt{\lambda} \right\}, \qquad \mathcal{H}_r \doteq span\{\bar{U}_i\}_{i=1}^L, \qquad \pi_r \doteq \sum_{i=1}^L \bar{U}_i \bar{U}'_i$$

Denote $\bar{\boldsymbol{\Psi}} \doteq \pi_r \boldsymbol{\Psi}$. The associated Galerkin approximation of the original ensemble is then

$$\bar{\boldsymbol{\Phi}} \doteq \pi_r \tilde{\boldsymbol{\Phi}} = \left\{ \bar{\Phi}_i \doteq \sum_{j=1}^N f_{i,j} \bar{U}_j \;:\; f_i \doteq \bar{U}'_i \tilde{\boldsymbol{\Phi}} \in \mathbb{R}^N \right\}_{i=1}^L$$

**Observation III.1** *The projected ensemble satisfies the resolution bound $\lambda$:*

$$\frac{tr\, \pi_r \tilde{\boldsymbol{\Phi}} \tilde{\boldsymbol{\Phi}}' \pi_r}{tr\, \tilde{\boldsymbol{\Phi}} \tilde{\boldsymbol{\Phi}}'} \geq \lambda$$

*Proof.* The following follows from the constructions in Steps 1 – 4s

$$\begin{aligned} tr\, \pi_r \tilde{\boldsymbol{\Phi}} \tilde{\boldsymbol{\Phi}}' \pi_r &= tr\, \pi_r \left( \sum_{k=1}^K \tilde{\boldsymbol{\Phi}}_k \tilde{\boldsymbol{\Phi}}'_k \right) \pi_r \geq tr\, \pi_r \left( \sum_{k=1}^K \pi_{r,\,k} \tilde{\boldsymbol{\Phi}}_k \tilde{\boldsymbol{\Phi}}'_k \pi_{r,\,k} \right) \pi_r \\ &= tr\, \pi_r \left( \sum_{k=1}^K \boldsymbol{\Psi}_k \boldsymbol{\Psi}'_k \right) \pi_r = tr\, \pi_r \boldsymbol{\Psi} \boldsymbol{\Psi}' \pi_r = tr\, \bar{\boldsymbol{\Psi}} \bar{\boldsymbol{\Psi}}' \geq \sqrt{\lambda}\, tr\, \boldsymbol{\Psi} \boldsymbol{\Psi}' \\ &= \sqrt{\lambda}\, tr\, \sum_{k=1}^K \boldsymbol{\Psi}_k \boldsymbol{\Psi}'_k = \sqrt{\lambda}\, tr\, \sum_{k=1}^K \bar{\bar{\boldsymbol{\Phi}}}_k \bar{\bar{\boldsymbol{\Phi}}}'_k \geq \lambda\, tr\, \sum_{k=1}^K \tilde{\boldsymbol{\Phi}}_k \tilde{\boldsymbol{\Phi}}'_k = \lambda\, tr\, \tilde{\boldsymbol{\Phi}} \tilde{\boldsymbol{\Phi}}' \qquad \square \end{aligned} \qquad (4)$$

*Concluding comments on computational savings.* The algorithm trades a single POD procedure of a set of $N$ samples, by $K$ POD computations for sub-ensembles of size $N_0 = N/K$, and a final POD computation for an ensemble of size $\bar{N} = \sum_{k=1}^K \bar{N}_k$. Assuming the generic case $N \ll M$, the computational complexity is then proportional to

$$K \cdot M \frac{N^2}{K^2} + M \cdot \bar{N}^2 = M \left( \frac{N^2}{K} + \bar{N}^2 \right)$$

An increase in K would thus reduce the computational complexity of POD of each of the sub-ensembles, but this advantage would eventually be offset by the growth in $\bar{N} = \sum_k N_k$. The values of $N_k$, hence of $\bar{N}$, can often be estimated a priori, from familiarity with the system. For example, in a periodically dominated system, a lower bound $N_*$ on $N_k$ can be estimated when each sub-ensemble reduces to a single or few periods. An estimated lower bound on $\bar{N}$ would then be $K \cdot N_*$. It is easy to see that an a priori estimate of the optimal value of $K$, to minimize the overall computational complexity, would then be

$$K_* \approx \frac{1}{(2R^2)^{\frac{1}{3}}} \qquad \text{where} \qquad R \doteq \frac{N_*}{N}$$

American Institute of Aeronautics and Astronautics

leading to an overall computational complexity of

$$\frac{3}{4^{\frac{1}{3}}} R^{\frac{4}{3}} M N^2$$

a plot of the a priori estimated of the ratio of computational savings, as a function of $R = N_*/N$, is given in Figure 1. As an example a reference length of 100 periods with 60 sampled per period and an a priori estimate of achieving the desired resolution by 10 modes over short (single or few periods), translates to $R = 0.017$ and a reduction of the computational burden to about 3.73% of that required by a standard POD algorithm.

**Relative computatational complexity as a function of R**
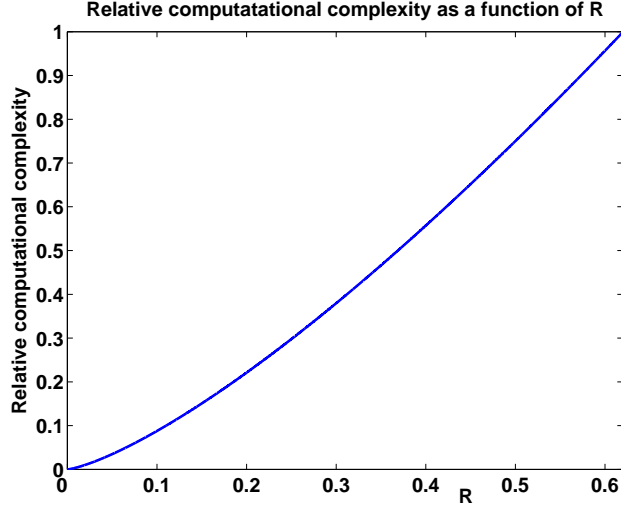


Figure 1. The a priori estimated relative computational complexity of the the algorithm in Section III, compared with the computational complexity of standard POD, as a function of $R \doteq N_*/N$.

## IV. Time Partition: A Local Mean Algorithm

This section concerns a variant of the algorithm where the global mean is not computed in a preliminary procedure. This remove the need to load the entire data set twice. Thus, the local mean is computed first for each sub-ensemble, and is used in the local POD procedure, instead of the global mean. The global mean is evaluated and used only in a counterpart of Step 4.

As above, the effect of this added component on the efficiency of the algorithm depends on the quality of local estimates of the global mean. Pertinent a priori knowledge can thus be exploited in an effective partitioning of the global ensemble.

*Step 1* For each $k = 1, \ldots, K$, compute the local sub-ensemble mean $\Phi_{k,0} = mean\,(\mathbf{\Phi_k})$, the perturbation sub-ensemble

$$\widehat{\mathbf{\Phi}}_k = \left\{ \widehat{\Phi}_{k,i} \doteq \Phi_i - \Phi_{k,0} \,:\, i \in \mathcal{T}_k \right\}$$

and an SVD of the corresponding perturbation operator

$$\widehat{\mathbf{\Phi}}_k = \mathbf{U}_k \mathbf{S}_k \mathbf{V}'_k = \sum_{i=1}^{N_0} \sigma_{k,i} U_{k,i} V'_{k,i}.$$

*Step 2:* Truncate the SVD of Step 1 at $N_k \leq N_0$, substituting the original sample sub-ensemble $\widehat{\mathbf{\Phi}}_k$ by the columns of the matrix

$$\check{\mathbf{\Phi}}_k \doteq \sum_{i=1}^{N_k} \sigma_{k,i} U_{k,i} V'_{k,i}$$

where

$$N_k \doteq \min \left\{ \tilde{N} \; : \; \frac{\sum_{i=1}^{\tilde{N}} \sigma_{k,i}^2}{\sum_{i=1}^{N_0} \sigma_{k,i}^2} \geq \sqrt{\lambda} \right\}, \qquad \pi_{r,k} \doteq \sum_{i=1}^{N_k} U_{k,i} U_{k,i}'$$

Let $\Phi_0 = mean(\mathbf{\Phi})$ be the yet unknown global mean, let $\tilde{\Phi}_{k,0} \doteq \Phi_{k,0} - \Phi_0$, let $\mathcal{H}_{rk} \doteq span\{U_{k,i}\}_{i=1}^{N_k}$, and denote $\eta = [1, 1, \ldots, 1]' \in \mathbb{R}^{N_0}$. Given that $0 = mean(\widehat{\mathbf{\Phi}_k}) = \frac{1}{\mathbf{N_0}} \widehat{\mathbf{\Phi}_k} \eta$, the following is a routine probabilistic equality

$$\tilde{\mathbf{\Phi}}_k \tilde{\mathbf{\Phi}}_k' = (\widehat{\mathbf{\Phi}}_k + \tilde{\Phi}_{k,0}\eta')(\widehat{\mathbf{\Phi}}_k + \tilde{\Phi}_{k,0}\eta')' = \widehat{\mathbf{\Phi}}_k \widehat{\mathbf{\Phi}}_k' + N_0 \tilde{\Phi}_{k,0} \tilde{\Phi}_{k,0}'$$

The truncated version of $\tilde{\mathbf{\Phi}}_k$ is denoted

$$\bar{\mathbf{\Phi}}_k \doteq \check{\mathbf{\Phi}}_k + \tilde{\Phi}_{k,0}\eta'$$

Since $mean(\check{\mathbf{\Phi}}_k) = mean(\pi_{r,k}\widehat{\mathbf{\Phi}_k}) = \mathbf{0}$, hence $\check{\mathbf{\Phi}}_k \eta = 0$, a repetition of the equalities right above, leads to

$$\begin{aligned} tr\, \bar{\mathbf{\Phi}}_k \bar{\mathbf{\Phi}}_k' = tr\, \left( \check{\mathbf{\Phi}}_k + \tilde{\Phi}_{k,0}\eta' \right) \left( \check{\mathbf{\Phi}}_k + \tilde{\Phi}_{k,0}\eta' \right)' &= tr\, \left( \check{\mathbf{\Phi}}_k \check{\mathbf{\Phi}}_k' + N_0 \tilde{\Phi}_{k,0} \tilde{\Phi}_{k,0}' \right) \\ &\geq \sqrt{\lambda}\, tr\, \left( \widehat{\mathbf{\Phi}}_k \widehat{\mathbf{\Phi}}_k' + N_0 \tilde{\Phi}_{k,0} \tilde{\Phi}_{k,0}' \right) = \sqrt{\lambda}\, tr\, \tilde{\mathbf{\Phi}}_k \tilde{\mathbf{\Phi}}_k' \end{aligned} \qquad (5)$$

*Step 3:* Let $\eta_k \doteq [1, 1, \ldots, 1]' \in \mathbb{R}^{N_k+1}$, and let $\{W_{k,i}\}_{i=1}^{N_k} \subset \mathbb{R}^{N_k+1}$ be an orthonormal basis for $\eta_k^{\perp}$ (that is, let it be a maximal orthonormal set of zero-mean vectors in $\mathbb{R}^{N_k+1}$). Let $\check{\mathbf{U}}_k \doteq [U_{k,i}]_{i=1}^{N_k}$, $\check{\mathbf{\Sigma}}_k \doteq diag\{\sigma_{k,i}\}_{i=1}^{N_k}$ and $\check{\mathbf{W}}_k = [W_{k,i}]_{i=1}^{N_k}$, and define the alternative truncated sub-ensemble as the columns of the matrix

$$\check{\mathbf{\Psi}}_k \doteq \check{\mathbf{U}}_k \check{\mathbf{\Sigma}}_k \check{\mathbf{W}}_k' = \sum_{i=1}^{N_k} \sigma_{k,i} U_{k,i} W_{k,i}' \qquad \Rightarrow \qquad \check{\mathbf{\Psi}}_k \check{\mathbf{\Psi}}_k' = \check{\mathbf{\Phi}}_k \check{\mathbf{\Phi}}_k'$$

and its (unknown) counterpart

$$\mathbf{\Psi}_k = \check{\mathbf{\Psi}}_k + \sqrt{\frac{N_0}{N_k + 1}} \Phi_{k,0} \eta_k'$$

In these terms there holds

$$tr\, \mathbf{\Psi}_k \mathbf{\Psi}_k' = tr\, \left( \check{\mathbf{\Psi}}_k \check{\mathbf{\Psi}}_k' + N_0 \Phi_{k,0} \Phi_{k,0}' \right) = tr\, \left( \check{\mathbf{\Phi}}_k \check{\mathbf{\Phi}}_k' + N_0 \Phi_{k,0} \Phi_{k,0}' \right) \geq \sqrt{\lambda}\, tr\, \tilde{\mathbf{\Phi}}_k \tilde{\mathbf{\Phi}}_k'$$

*Step 4:* The added component, when compared with Step 4 of Section III, is the computation of the global mean from the previously computed sub-ensemble means (here we explicitly use the fact that the sub-ensembles are equal in size[a]):

$$\Phi_0 = mean\, \{\Phi_{k,0}\}_{k=1}^K$$

Having computed $\Phi_0$, the values of the sub-ensembles $\mathbf{\Psi_k}$ becomes explicit. The completion of Step 4, and thus, of the algorithm, therefore proceed precisely as in Section III, leading to the same conclusion:

**Observation IV.1** *The projected ensemble satisfies the resolution bound $\lambda$:*

$$\frac{tr\, \pi_r \tilde{\mathbf{\Phi}} \tilde{\mathbf{\Phi}}' \pi_r}{tr\, \tilde{\mathbf{\Phi}} \tilde{\mathbf{\Phi}}'} > \lambda$$

# V.  Computational Complexity of Building Covariance

## A.  Full Covariance

Since the full covariance is calculated using the entire data set $\mathbf{\Phi}$, the global mean $\mathbf{\Phi}_0$ must be calculated at the outset to produce the zero mean set $\tilde{\mathbf{\Phi}} = \mathbf{\Phi} - \mathbf{\Phi}_0 \eta'$, where $\eta$ is the column vector $[1 \ldots 1]^T$ of dimension $N$. This zeroing process requires $M(N-1)$ additions to produce the global mean

$$\mathbf{\Phi}_0 = \frac{1}{N} \sum_{i=1}^{N} \Phi_i$$

---

[a]Alternative approaches, based on sub-ensemble partitioning with variable size and weight will lead to obvious counterparts.

and then another $MN$ subtractions to zero the set, resulting in a total of $M(2N-1)$ addition operations to produce $\tilde{\boldsymbol{\Phi}}$.

Computation of the complete covariance matrix among $N$ zero mean snapshots $\tilde{\Phi}_i \in \tilde{\boldsymbol{\Phi}} \subset \mathbb{R}^{M \times N}$ requires $N^2$ inner product calculations. Since the covariance is the symmetric matrix $C = \left( \tilde{\boldsymbol{\Phi}}, \tilde{\boldsymbol{\Phi}} \right)$, this is immediately reduced to $N(N+1)/2$ inner products needed to produce the upper diagonal matrix used in eigenvalue routines. Each inner product is the discretized version of (II), approximated by the sum

$$(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \int_{\Omega} \boldsymbol{\Phi} \cdot \boldsymbol{\Psi} dV \approx \sum_{i=1}^{M} \boldsymbol{\Phi}_i \cdot \boldsymbol{\Phi}_i w_i, \tag{6}$$

where $M$ is the total number of grid points times the number of spatial dimensions, and each $w_i$[b] is the weight assigned to each grid point used to perform the numeric integration. This implies that each inner product requires $2M$ multiplications and $M-1$ additions. Thus, in order to calculate the covariance in $d$ dimensional space

$$[N(N+1)/2](M-1) + M(2N-1) \tag{7}$$

additions and

$$N(N+1)M \tag{8}$$

multiplications are performed.

## B. Partitioned Covariance

*Stage 1:* Compare this to the computations performed in the iterative POD algorithm when dividing the set $\tilde{\boldsymbol{\Phi}}$ into $K$ partitions[c], each of size $L_1$. In the first stage, each partition $\boldsymbol{\Phi}_k \subset \mathbb{R}^{M \times L_1}$ is zeroed using the local mean $\boldsymbol{\Phi}_{k,0}$, and the upper diagonal portion of each covariance matrix need be calculated. Since $K$ such matrices must be calculated, this results in

$$K \left\{ [L_1(L_1+1)/2](M-1) + M(2L_1-1) \right\} \tag{9}$$

addition operations and

$$K \left\{ L_1(L_1+1)M \right\} \tag{10}$$

multiplications in the first stage. This partitioning process results in $K$ sets of local modes, resulting in a total $L_2$ modes that are passed on to stage 2.

*Stage 2:* The first step in stage 2 of the iterative algorithm requires producing new snapshot sets from each set of partition modes found in stage 1. This is performed by multiplying each of the mode sets $\boldsymbol{\Psi}_k$ by its appropriate zero-mean orthonormal set $W_k$. This procedure requires

$$(L_2 - K)(L_2 + K)M \tag{11}$$

additions and

$$ML_2(L_2 + K) \tag{12}$$

multiplications. Following this, the new reconstructed snapshots require a correction factor to correct for the local mean zeroing process used on each partition. To do this, each of the $K$ partition means $\boldsymbol{\Phi}_{k,0}$ must be multiplied by the correction factor

$$\alpha_k = \sqrt{\frac{N_0}{N_k + 1}}$$

an additional $KM$ multiplications. The mean correction also requires

$$M(L_2 + K) \tag{13}$$

---

[b]note that the vector $w$ is actually the $d$-repetition of the weights of the grid points, one for each of the $d$-dimensional components of the velocity.

[c]The assumption is made that the set $\tilde{\boldsymbol{\Phi}}$ divides evenly into $K$ partitions. In this case the maximum number of computations is performed. Otherwise, the set $\tilde{\boldsymbol{\Phi}}$ is reduced to contain the maximum number of snapshots possible to contain $K$ partitions of equal size.

American Institute of Aeronautics and Astronautics

additions. From here, the covariance is built exactly as in stage 1 with matrix of size $M \times (L_2 + K)$, resulting in

$$[(L_2 + K)(L_2 + K + 1)/2](M - 1) \tag{14}$$

additions and

$$(L_2 + K)(L_2 + K + 1)M \tag{15}$$

multiplications.

*Totals:* Taking advantage of the fact that $L_1 = N/K$, the totals for the iterative algorithm are thus

$$\frac{N(N + K)(M - 1)}{2K} + (2N - K)M + \left[ (L_2 - K + 1)M + \frac{(L_2 + K + 1)}{2}(M - 1) \right] (L_2 + K) \tag{16}$$

additions,

$$\left[ \frac{N(N + K)}{K} + K + (L_2 + K)(2L_2 + K + 1) \right] M \tag{17}$$

multiplications. The expression (16) can be simplified by noting that in any fluid simulation the number of data points $M$ is much larger than 1, and so $(M - 1) \approx M$ this leads to a simplified addition count of

$$\left[ (2N - K) + \frac{N(N + K)}{2K} + \frac{1}{2}(3L_2 - K + 3)(L_2 + K) \right] M. \tag{18}$$

The ratio of equations

$$\frac{[(18) + (17)]}{[(7) + (8)]} \tag{19}$$

is precisely the arithmetic savings expected when using the iterative algorithm instead of that which requires calculation of the complete covariance matrix. This is quadratic with respect to the chosen number of partitions $K$, allowing for optimal partitioning provided some *a priori* knowledge of the second stage covariance dimension $L_2$. For our purposes, as described in the following section, the dimension $L_2$ is estimated by making an assumption as to the average number of modes $N_k$ retained when truncating each partition $\hat{\mathbf{\Phi}}_k$, allowing for optimization in $K$.

## VI.   Numerical Results

The algorithm is demonstrated in POD analysis of simulation data gathered for a flat plate in both two and three dimensions.

### A.   2-D Simulation Benchmark Case

First considered here is the two dimensional incompressible flow simulation using the fast immersed boundary method described in[33] . This simulation was run at the Reynolds number of Re = 300, where the flow is non-dimensionalized with respect to the incoming flow velocity $U_\infty = 1$ and the plate's chord, $c = 1$. The computational domain is

$$\Omega_{2D} = \{ \mathbf{x} = (x, y) : -5 \leq x \leq 15, \ -5 \leq y \leq 5 \} \tag{20}$$

where the center of the plate center is at the origin. The simulation employs a rectangular $251 \times 251$ grid (i.e., comprising $63,001$ points) covering the physical domain $\Omega$ and is conducted using a time step of $dt = .01$ convective time units (CTU), with snapshots saved every 10 time steps.

The data trajectory under study begins at the natural attractor, at an angle of attack (AOA) of $30^o$, where the flow is fully separated and characterized by periodic, alternating vortex shedding from the leading and the trailing edges of the plate. The AOA is then ramped down to zero, and after the flow settles, ramped back to AOA = $30^o$ (Figure 2), thus dynamically crossing the bifurcation into and out of vortex shedding, both ways. The entire simulation lasted 200 convective time units, captured by 2000 snapshots.

This transient is characterized not only by the transition out of, and then back into, instability, but also by changes in the dominant temporal and spatial frequencies in the instability regime, as the angle of attack and the wake diameter change, and by substantial dynamic changes in the mean field, both during

American Institute of Aeronautics and Astronautics
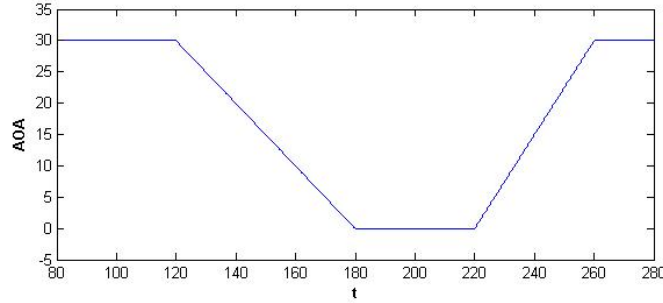
**Figure 2. Angle of attack of the flat plate. The domain $80 \leq T \leq 280$ is chosen such that motion begins well after the simulated flow has settled on the attractor.**

the changes of the AOA and during the time interval where AOA=$0^o$. As a demonstration of the techniques proposed in this article we require the POD compression to fully capture this complex behavior at the high level of $\lambda = 0.99$. The number of modes required for this selected $\lambda$ by a standard POD approximation is 34. In this two dimensional case, the nature of the modes found by POD approximation can be characterized as contributing the resolution of either the non-oscillatory mean field correction,[14,34] or of harmonics of the periodic vortex shedding.

The number of partitions used in an application of the accelerated algorithm was chosen by an *a priori* estimate that each individual partition will require an average of 12 modes to satisfy the resolution bound $\sqrt{\lambda}$ (Figure 4), optimizing the ratio (19) for $K$ by describing $L_2$ as the function

$$L_2(K) = (12+1)K.$$

This assumption is based on analysis of single vortex shedding periods, and produces the estimated optimal value $K = 19$ (Figure 3).
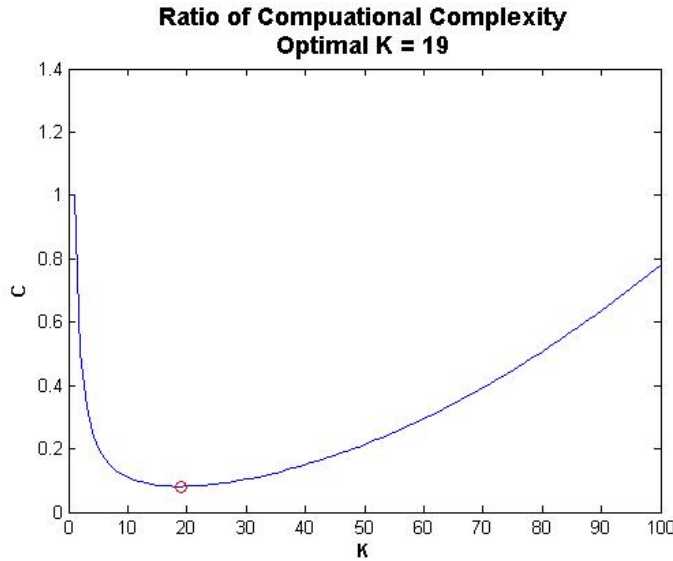


**Figure 3. Solution for choosing optimal partitioning of the two dimensional benchmark data set using the assumption mean($N_k$)=12, $k = 1, \ldots, K$**

The approximate POD algorithm required the clearly suboptimal 42 modes, as compared with the optimal of 34 modes in the standard POD approximation, to satisfy the resolution bound

$$tr\pi_r\tilde{\mathbf{\Phi}}\tilde{\mathbf{\Phi}}'\pi_r \geq \lambda tr\tilde{\mathbf{\Phi}}\tilde{\mathbf{\Phi}}'$$

American Institute of Aeronautics and Astronautics

as introduced in (4). However, the CPU time dedicated to POD arithmetics was reduced by a factor of $\times 15.9$, as compared with the POD algorithm, and close to ideal prediction. Furthermore, the first 30 POD eigenvalues, shown in Figure 5 are nearly identical to those produced by the approximate algorithm, and the turbulent kinetic energy within the system is well preserved by the approximation (Figures 6,7).
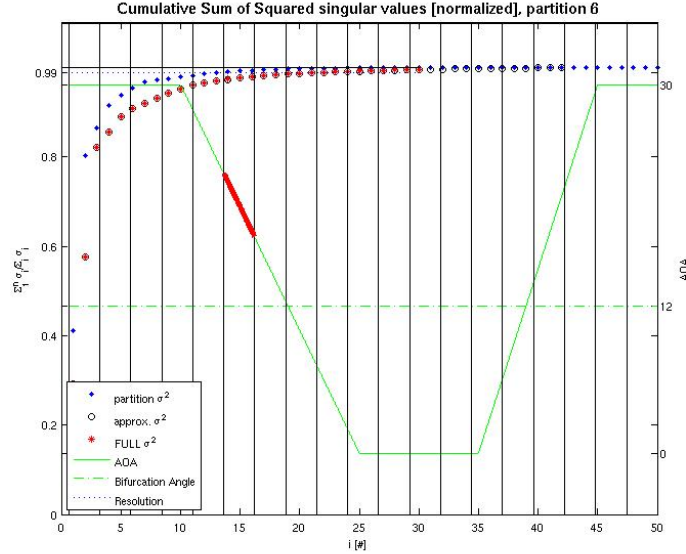


**Figure 4. An example of the local singular values found for one of the 19 partitions within the iterative algorithm, compared to the final result.**

## B.  3-D Simulation

The three dimensional data set is produced by a simulation of the incompressible flow over a variable pitch flat plate of chord length $c = 1$ and aspect ratio $AR = 2$, at $Re = 300$. The same non-dimensionalization conventions as in the two dimensional case are used here. This simulation was carried over the domain

$$\Omega_{3D} = \{ \mathbf{x} = (x, y, z) \, : \, -4.0368 \leq x \leq 6.1, \; -5 \leq y \leq 5, \; -5 \leq z \leq 5 \, \}, \tag{21}$$

using a rectangular grid of size $126 \times 56 \times 81$, thus producing data snapshots of dimension $1.7 \cdot 10^6$. As before, the simulation advances in increments of $dt = 0.01$ CTU, saving every $50^{th}$ snapshot, i.e., at 0.5 CTU intervals, and a total of 200 snapshots for the simulation considered here.

Before pitching of the plate begins, the simulation is run for 40 CTU at AOA=$0^o$ so that the flow field has time to overcome the transients present in the impulsive start. This final field is then used as the initial state for the pitching plate simulation. Once pitching begins, the angle of attack $\alpha$ is described by

$$\dot{\alpha}(t) = \alpha_{max} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[ -\frac{(t - t_a)^2}{2\sigma^2} \right] \quad \text{and} \quad \alpha(t) = \alpha_{max} \frac{1}{2} \left[ 1 + \text{erf}\left( \frac{t - t_a}{\sqrt{2}\sigma} \right) \right]$$

where $\alpha_{max} = 30^o$, $t_a = 3\sigma$, and $\sigma = 3$ are chosen. The motion of the plate is considered complete after $t = 18$, where the remainder of the simulation allows the flow to settle to the attractor (FIgure 8).

The periodic flow of the attractor is sampled at very low temporal resolution, which places a practical bound on the choice of $K$ used in the approximate POD algorithm. Specifically, we found that a partition size of 20 snapshots was the smallest that could be used while still capturing full periods of the dominant modes of the system (Figure 9, forcing the upper bound $K \leq 10$.

Another limitation occurs due to the much higher complexity, inherent in the three dimensional system. Along with the low temporal sampling rate, this leads to the retention of considerably more modes per partition than observed in the two dimensional case. Again in the test case of this particular data set, each partition of 20 snapshots was only reduced to an average of 7-8 modes, even when given a lower resolution
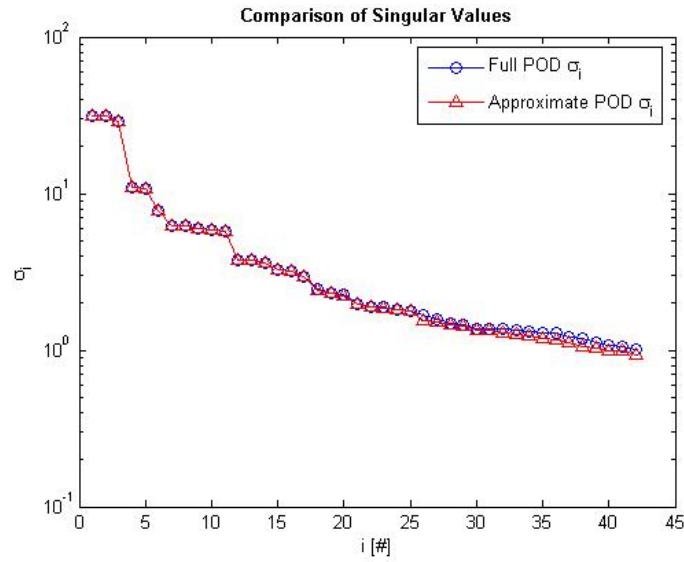
**Figure 5. Singular values as solved by both the actual and the approximate POD algorithms, both using the resolution bound $\lambda = 0.99$**
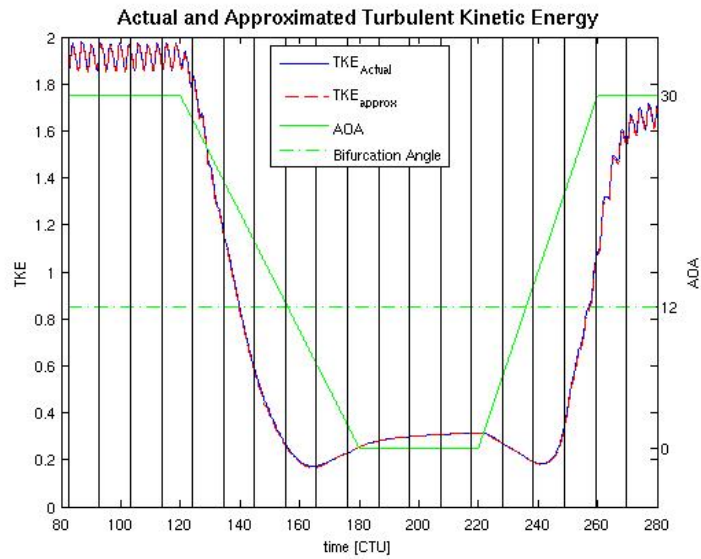


**Figure 6. Turbulent Kinetic Energy of the approximated space compared to that of the original snapshot matrix.**
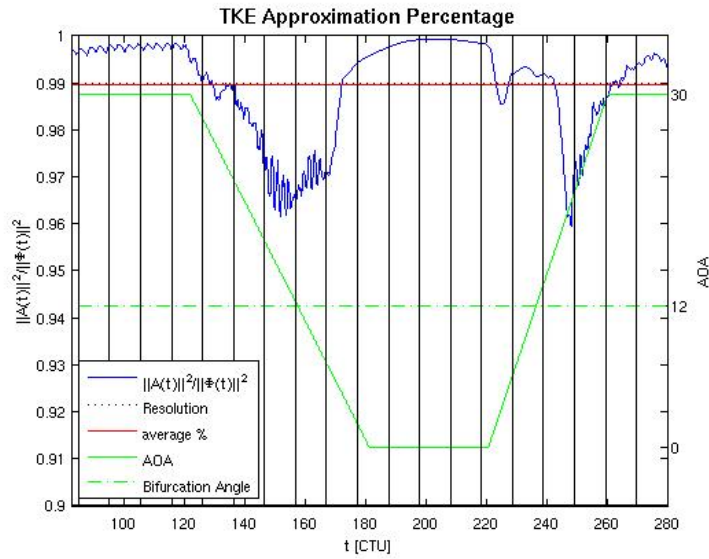
American Institute of Aeronautics and Astronautics

**Figure 7. Percentage of the TKE preserved in the approximation of the two dimensional data set performed by the iterative POD.**
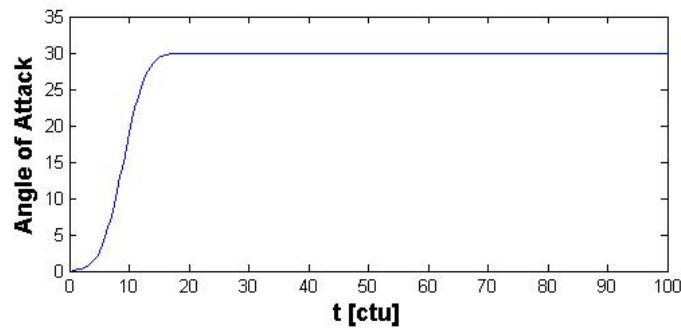


**Figure 8. Angle of Attack $\alpha(t)$ used in the three dimensional pitching plate simulation.**
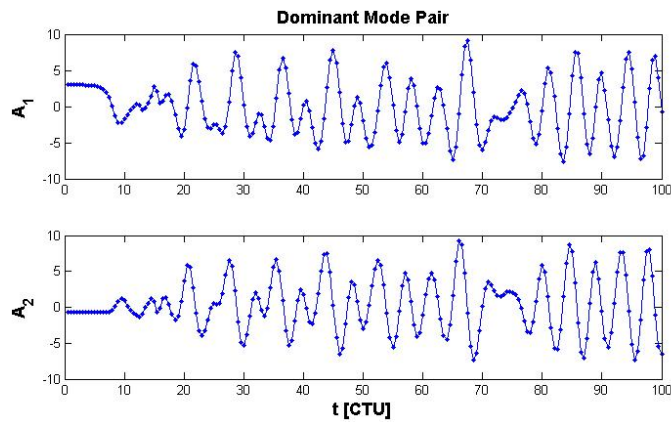


**Figure 9. Dominant mode coefficients of the three dimensional pitching plate simulation. Note the long transient as the flow responds to the motion of the plate, finally settling to the attractor at $t > 80$.**

American Institute of Aeronautics and Astronautics

bound than we specified of $\lambda = 0.8$[d]. This proved to reduce the data to 47 modes after stage 1, finally resulting in the 8 modes needed to approximate the complete snapshot set, representing the transient.
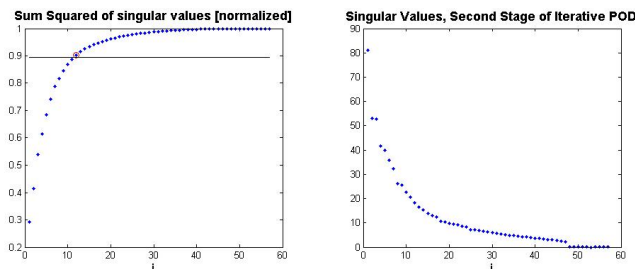


**Figure 10. Singular values of the three dimensional data after the second stage of the accellerated algorithm. A single shift mode and the dominant pair are clearly present ($\sigma_i$, $i = 1, 2, 3$), the next four ($i = 4, \ldots, 7$) appear as a single harmonic group.**

As in the two dimensional case, again members of the orthonormal set produced by this algorithm can be classified as either modes representing mean field correction (termed *shift* modes), or as harmonically influenced groups. the relative energetic contributions of these modes is indicated by the relative strength of the singular values (Figure 10). Examples of the dominant shift mode and the first harmonic pair are shown in Figures (11,12), respectively. However, in the more complex three dimensional case the grouping of harmonically influenced modes is much less distinct outside of the dominant shedding frequency. The purpose of the companion paper[35] is to to further process the current compression to obtain harmonically coherent structures.
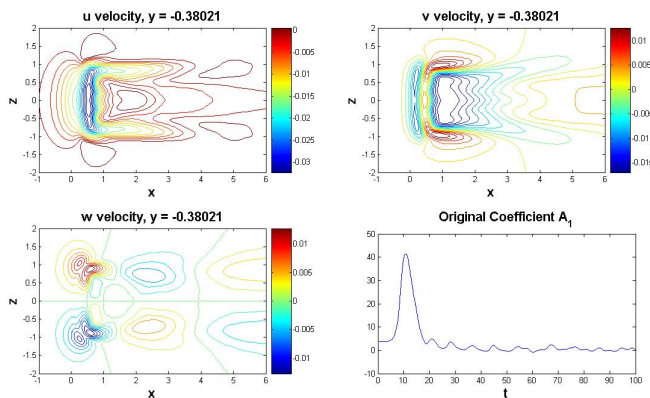


**Figure 11. Original POD mode, the dominant shift mode, viewed from above at depth $y = -0.38021$.**

## VII.   Acknowledgments

---

[d]Higher resolution bounds than $\lambda = 0.8$ have not been achieved at this time, primarily due to memory constraints.
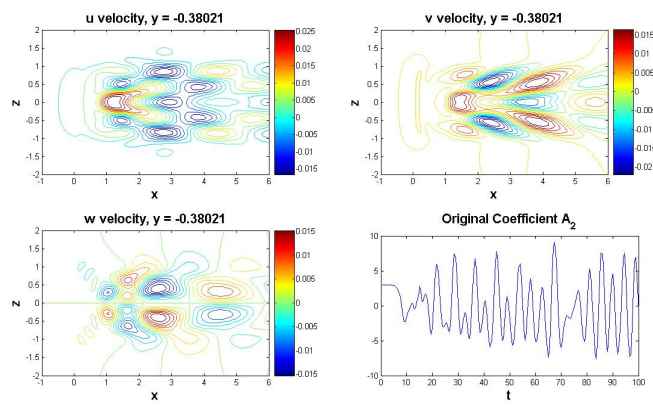
American Institute of Aeronautics and Astronautics

**Figure 12. Original POD mode 2, the first of the dominant harmonic pair, viewed from above at depth $y = -0.38021$.**

American Institute of Aeronautics and Astronautics

# References

[1] L. Sirovich. Turbulence and the dynamics of coherent structures, Parts I–III. *Quart. Appl. Math.*, XLV:561–590, 1987.

[2] P. Holmes, J.L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge University Press, Cambridge, 1998.

[3] T. R. Smith, J. Moehlis, and P. Holmes. Low-dimensional modelling of turbulence using the proper orthogonal decomposition: A tutorial. *Nonlinear Dynamics*, 41:275 – 307, 2005.

[4] K. Karhunen. über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci., Fennicae, Ser. A. I., Math.-Phys.*, 37:1–79, 1946.

[5] M. M. Loève. *Probability Theory.* VanNostrand, 1955.

[6] V.R Algazi and D.J. Sakrison. On the optimality of the Karhunen-Loève expansion. *IEEE Trans. Inform. Theory*, 15:319–321, 1969.

[7] E. N. Lorenz. Empirical orthogonal functions and statistical weather prediction. Technical report, MIT, Department of Meteorology, 1956. Statistical Forecasting Project.

[8] J. Rinne and V. Karhilla. A spectral barotropic model in horizontal empirical orthogonal functions. *Quart. J. Roy. Meteor. Soc.*, 101:365 – 382, 1975.

[9] A. Armou and P. D. Christofides. Finite-dimensional control of nonlinear parabolic PDE systems with time-dependent spatial domains with empirical eigenfunctions. *Int. J. Appl. Math. Comput. Sci.*, 11:287 – 317, 2001.

[10] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Elsevier, 1990.

[11] K. Pearson. "on lines and planes of closed fit to systems of points in space. *Philosophical Magazines*, 2(559-572), 1901.

[12] B.R. Noack, P. Papas, and P.A. Monkewitz. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *J. Fluid Mech.*, 523:339–365, 2005.

[13] A.E. Deane, I.G. Kevrekidis, G.E. Karniadakis, and S.A. Orszag. Low-dimensional models for complex geometry flows: Application to grooved channels and circular cylinders. *Phys. Fluids A*, 3(10):2337–2354, 1991.

[14] B.R. 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.

[15] K. Cohen, S. Siegel, T. McLaughlin, and J. Myatt. Proper orthogonal decomposition modeling of a controlled Ginzburg-Landau cylinder wake model. In *41st Aerospace Sciences Meeting and Exhibit, Reno NV*, 2003. paper-2003-1292.

[16] W. Cazemier, R.W.C.P. Verstappen, and A.E.P. Veldman. Proper orthogonal decomposition and low-dimensional models for driven cavity flows. *Phys. Fluids*, 10:1685–1699, 1998.

[17] C.W. Rowley, D.R. Williams, T. Colonius, R.M. Murray, and D.G. MacMartin. Linear models for control of cavity flow oscillations. *J. Fluid Mechanics*, 547:317 – 330, 2006.

[18] M. Samimy, M. Debiasi, E. Caraballo, A. Serrani, X. Yuan, J. Little, and J. Myatt. Feedback control of subsonic cavity flows using reduced-order models. *J. Fluid Mechanics*, 579:315–346, 2007.

[19] O. Lehmann, M. Luchtenburg, B.R. Noack, R. King, M. Morzynski, and G. Tadmor. Wake stabilization using POD Galerkin models with interpolated modes. In *44th IEEE Conference on Decision and Control and European Control Conference ECC, Seville, Spain*, 2005.

[20] M. Bergmann, L. Cordier, and J.-P. Brancher. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced order model. *Phys. Fluids*, 17:097101–097121, 2005.

[21] S. Siegel, K. Cohen, J. Seidel, M. Luchtenburg, and T. McLaughlin. Low dimensional modelling of a transient cylinder wake using double proper orthogonal decomposition. *J. Fluid Mech*, 2008. accepted for publication.

[22] J. Borggaard, A. Hay, and D. Pelletier. Interval-based reduced order models for unsteady fluid flow. *Int. J. Num. Anal. & Modeling*, , 2006. submitted.

[23] L. Eldén. Numerical linear algebra in data mining. *Acta Numerica*, pages 327–384, 2006.

[24] H. Park and L. Eldén. Matrix rank reduction for data analysis and feature extraction. In E.J. Kontoghiorghes, editor, *Handbook of parallel computing and statistics*, pages 449 – 476. CRC Press, 2006.

[25] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Incremental svd-based algorithms for highly scalable recommender systems. In *5th International Conference in Computers and Information Technology*, 2005.

[26] E. Drinea, P. Drineas, and P. Huggins. A randomized singular value decomposition algorithm for image processing. In *8th Panhellenic Conference on Informatics (PCI)*, pages 278–288, 2001.

[27] G. Uytterhoeven, F. Van Wulpen, M. Jansen, D. Roose, and A. Bultheel. Waili : a software library for image processing using integer wavelet transforms. In K.M. Hanson, editor, *Medical Imaging 1998: Image Processing*, volume 3338, pages 1490–1501, 1998.

[28] M. E. Hochstenbach. A jacobi–davidson type svd method. *SIAM Journal on Scientific Computing*, 23:606–628, 2001.

[29] P. Drineas, R. Kannan, A. Frieze, S. Vempala, and V. Vinay. Clustering of large graphs via the singular value decomposition. *EEE Journal of Machine Learning*, 56:9–33, 2004.

[30] A.C. Antoulas and D. C. Sorensen. Approximation of large-scale dynamical systems: An overview. *Int. J. Appl. Math. Comp. Sci.*, 11:1093–1121, 2001.

[31] C. Beattie, J. Borggaard, S. Gugercin, and T. Iliescu. A domain decomposition approach to POD. In *Proc. of the 45th IEEE Conference on Decision and Control*, 2006.

[32] M. Morzyński, W. Stankiewicz, B. R. Noack, R. King, F. Thiele, and G. Tadmor. Continuous mode interpolation for control-oriented models of fluid flow. In R. King, editor, *Active Flow Control*, volume 95 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 260–278. Springer Verlag, Berlin, Germany, 2007.

[33] T. Colonius and K. Taira. The immersed boundary method: A projection approach. *.J. Comp. Physics*, 225:2118 — 2137, 2007.

[34] G. Tadmor, O. Lehmann, B. R. Noack, and M. Morzyński. Mean field representation of the natural and actuated cylinder wake. *Phys. Fluids*, 2008. submitted.

[35] G. Tadmor, D. Bissex, B. R. Noack, M. Morzyński, T. Colonius, and K. Taira. Temporal-harmonic specific pod mode extraction. In *Temporal-Harmonic specific POD mode extraction*, 2008. AIAA Paper 2008-4190.