An efficient streaming algorithm for spectral proper orthogonal decomposition

Oliver T. Schmidt

California Institute of Technology, Pasadena, CA 91125, USA

Abstract

A streaming algorithm to compute the spectral proper orthogonal decomposition (SPOD) of stationary random processes is presented. As new data becomes available, an incremental update of the truncated eigenbasis of the estimated cross-spectral density (CSD) matrix is performed. The algorithm converges orthogonal sets of SPOD modes at discrete frequencies that are optimally ranked in terms of energy. We define measures of error and convergence, and demonstrate the algorithm's performance on two datasets. The first example is that of a high-fidelity numerical simulation of a turbulent jet, and the second optical flow data obtained from high-speed camera recordings of a stepped spillway experiment. For both cases, the most energetic SPOD modes are reliably converged. The algorithm's low memory requirement enable real-time deployment and allow for the convergence of second-order statistics from arbitrarily long streams of data.

Keywords:

1. Introduction

The ability to represent complex dynamics of by a small number of dynamically important modes enables the analysis, modeling and control of high-dimensional systems. Turbulent flows are a prominent example of such systems [1, 2]. Proper orthogonal decomposition (POD), also known as principle component analysis (PCA) or Karhunen-Loève decomposition, is a popular modal decomposition technique to extract coherent structures from experimental and

numerical data. In its most common form [3], POD is conducted in the time domain. It is computed from a time series of snapshots and expands the flow field into a sum of products of spatially orthogonal modes and coefficients with random time dependence. POD modes are optimally ranked in terms of their variance, or energy. These properties make POD modes well suited for low-order models based on Galerkin projection of the Navier-Stokes equations [4, 5].

Besides its definitions in the temporal and spatial domains, POD can also be formulated in the frequency domain. This variant of POD called *spectral proper orthogonal decomposition* (SPOD) dates back to the early work of Lumley [6] and takes advantage of temporal homogeneity. This makes it ideally suited for statistically (wide-sense) stationary data [7]. SPOD provides orthogonal modes at discrete frequencies that are optimally ranked in terms of energy, and that evolve coherently in both space and time. Perhaps predictably, this optimal space-time representation of the data comes at a cost—very long time series are necessary in order to converge the second-order space-time statistics. This data demand also becomes apparent when comparing SPOD to other modal decomposition techniques, for example the popular dynamic mode decomposition (DMD) [8]. For stationary data, SPOD modes correspond to optimally averaged DMD modes computed from an ensemble of stochastic realizations of a process [7], for example multiple repetitions of the same experiment. In §5, we use two databases consisting of 10,000 and 19,782 snapshots, respectively. Evidently, the SPOD problem quickly becomes computationally unmanageable for data with large spatial dimensions.

In this paper, we address this issue by proposing a low-storage *streaming SPOD* algorithm that incrementally updates the SPOD as new data becomes available. Similar algorithms are often referred to as *incremental*, *learning*, *updating*, *on-the-fly* or *online* algorithms in the literature. Streaming algorithms for the DMD have been developed recently [9, 10], for example. The proposed streaming SPOD algorithm utilizes incremental updates of the singular value decomposition (SVD) of the cross-spectral density (CSD) matrix of the data. SVD updating is an active research topic for almost half a century, see e.g.

[11, 12]. In this work, we build on Brand’s [13] incremental singular value decomposition (SVD) by specializing the method to updates of the estimated CSD matrix. Originally developed for computer vision and audio feature extraction [14], the algorithm has been employed for recommender systems [15], semantics [16], design optimization [17], and a wide spectrum of other machine learning and data mining applications.

The paper is organized as follows. We first introduce standard or *batch* SPOD in §2 before deriving the streaming algorithm in §3. Measures of error and convergence are defined in §4. In §5, we demonstrate streaming SPOD on two datasets: a high-fidelity large eddy simulation (LES) of a turbulent jet and experimental optical flow from high-speed camera data of a stepped spillway. The effect of eigenbasis truncation is addressed in §6. In §7, we conclude with a discussion of the algorithm’s computational efficiency and utility in real-time and big data settings.

2. Batch SPOD

SPOD is the frequency-domain counterpart of standard time-domain or spatial POD. SPOD yields time-harmonic modes that represent structures that evolve coherently in both time and space. The method is based on an eigendecomposition of the CSD, which in turn is estimated from an ensemble of realizations of the temporal discrete Fourier transform (DFT) in practice. The CSD can be estimated using standard spectral estimation techniques such as Welch’s method [see e.g. 18] from an ensemble of snapshots. The SPOD formalism is derived from a space-time POD problem under the assumption of wide-sense stationarity. The reader is referred to [7] for the derivation of the method and an assessment of its properties. In particular, the method’s relations to DMD and the resolvent operator are interesting from a modeling perspective, as they link SPOD to concepts from hydrodynamic stability theory and dynamical systems.

Figure 1 serves a visual guide through the batch algorithm. We start with an

Figure 1: Illustration of the batch SPOD algorithm. Each rectangular slice represents a snapshot and the numbers in parentheses denote the equations in the text. The data is first segmented, then Fourier transformed, then reordered by frequency, and finally diagonalized into SPOD modes.

ensemble of n_t snapshots $\mathbf{q}_i = \mathbf{q}(t_i) \in \mathbb{R}^n$ of a wide-sense stationary process $\mathbf{q}(t)$ sampled at discrete times $t_1, t_2, \dots, t_{n_t} \in \mathbb{R}$. By $\mathbf{q}$ we denote the state vector. Its total length n is equal to the number of grid points $n_x = \prod_i^{n_{\text{dim}}} n_{x_i}$ times the number of variables n_{var} , where n_{dim} is the number of spatial dimensions. The temporal mean corresponds to the ensemble average defined as

$$\bar{\mathbf{q}} = \frac{1}{n_t} \sum_{i=1}^{n_t} \mathbf{q}_i \in \mathbb{R}^n. \quad (1)$$

We collect the mean-subtracted snapshots in a data matrix

$$\mathbf{Q} = [\mathbf{q}_1 - \bar{\mathbf{q}}, \mathbf{q}_2 - \bar{\mathbf{q}}, \dots, \mathbf{q}_{n_t} - \bar{\mathbf{q}}] \in \mathbb{R}^{n \times n_t} \quad (2)$$

of rank $d \leq \min\{n, n_t - 1\}$. With the goal of estimating the CSD, we apply Welch's method to the data by segmenting $\mathbf{Q}$ into n_{blk} overlapping blocks

$$\mathbf{Q}^{(l)} = [\mathbf{q}_1^{(l)} - \bar{\mathbf{q}}, \mathbf{q}_2^{(l)} - \bar{\mathbf{q}}, \dots, \mathbf{q}_{n_{\text{freq}}}^{(l)} - \bar{\mathbf{q}}] \in \mathbb{R}^{n \times n_{\text{freq}}} \quad (3)$$

containing n_{freq} snapshots each. If n_{ovlp} is the number of snapshots by which the blocks overlap, then the j -th column of the l -th block $\mathbf{Q}^{(l)}$ is given as

$$\mathbf{q}_j^{(l)} = \mathbf{q}_{j+(l-1)(n_{\text{freq}}-n_{\text{ovlp}})} - \bar{\mathbf{q}}. \quad (4)$$

We assume that each block can be regarded as a statistically independent realization under the ergodicity hypothesis. The purpose of the segmentation step is to artificially increase the number of ensemble members, i.e. Fourier realizations. This method is useful in the common scenario where a single long dataset with equally sampled snapshots is available, for example from a numerical simulation. In situations where the data presents itself in form of independent realizations from the beginning, segmenting is not applied. This is the case, for example, if an experiment is repeated multiple times. Next, the temporal (row-wise) discrete Fourier transform

$$\hat{\mathbf{Q}}^{(l)} = [\hat{\mathbf{q}}_1^{(l)}, \hat{\mathbf{q}}_2^{(l)}, \dots, \hat{\mathbf{q}}_{n_{\text{freq}}}^{(l)}] \in \mathbb{R}^{n \times n_{\text{freq}}} \quad (5)$$

of each block is calculated. A windowing function can be used to mitigate spectral leakage. All realizations of the Fourier transform at the k -th frequency

are subsequently collected into a new data matrix

$$\hat{\mathbf{Q}}_k = [\hat{\mathbf{q}}_k^{(1)}, \hat{\mathbf{q}}_k^{(2)}, \dots, \hat{\mathbf{q}}_k^{(n_{\text{blk}})}] \in \mathbb{R}^{n \times n_{\text{blk}}}. \quad (6)$$

At this point, we introduce the weighted data matrix

$$\mathbf{X}_k = \frac{1}{\sqrt{n_{\text{blk}}}} \mathbf{W}^{\frac{1}{2}} \hat{\mathbf{Q}}_k = [\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \dots, \mathbf{x}_k^{(n_{\text{blk}})}] \in \mathbb{R}^{n \times n_{\text{blk}}}, \quad (7)$$

where $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a positive-definite Hermitian matrix that accounts for quadrature and possibly other weights associated with the discretized inner product

$$\langle \mathbf{a}, \mathbf{b} \rangle_E = \mathbf{a}^* \mathbf{W} \mathbf{b}. \quad (8)$$

The inner product (8) induces the spatial energy norm $\|\cdot\|_E = \sqrt{\langle \cdot, \cdot \rangle_E}$ by which we wish to rank the SPOD modes. The product

$$\mathbf{S}_k = \mathbf{X}_k \mathbf{X}_k^* \in \mathbb{R}^{n \times n} \quad (9)$$

defines the weighted CSD matrix of the k -th frequency. A factor of $\frac{1}{n_{\text{blk}}}$ seen in other definitions of the CSD is absorbed into our definition of the weighted data matrix in equation (7).

SPOD is based on the eigenvalue decomposition

$$\mathbf{S}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^* \quad (10)$$

of the CSD matrix, where $\mathbf{\Lambda}_k = \text{diag}(\lambda_{k_1}, \lambda_{k_2}, \dots, \lambda_{k_{n_{\text{blk}}}}) \in \mathbb{R}^{n_{\text{blk}} \times n_{\text{blk}}}$ is the matrix of ranked (in descending order) eigenvalues and $\mathbf{U}_k = [\mathbf{u}_{k_1}, \mathbf{u}_{k_2}, \dots, \mathbf{u}_{k_{n_{\text{blk}}}}] \in \mathbb{R}^{n \times n_{\text{blk}}}$ the corresponding matrix of eigenvectors. Equivalently, we may consider the economy SVD of the weighted data matrix

$$\mathbf{X}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^*, \quad (11)$$

where $\mathbf{\Sigma}_k = \text{diag}(\sigma_{k_1}, \sigma_{k_2}, \dots, \sigma_{k_{n_{\text{blk}}}}) \in \mathbb{R}^{n_{\text{blk}} \times n_{\text{blk}}}$ is the matrix of singular values and $\mathbf{V}_k = [\mathbf{v}_{k_1}, \mathbf{v}_{k_2}, \dots, \mathbf{v}_{k_{n_{\text{blk}}}}] \in \mathbb{R}^{n \times n_{\text{blk}}}$ the right singular vector matrix. This can be shown by rewriting the CSD in terms of the SVD of the data matrix as $\mathbf{S}_k = \mathbf{X}_k \mathbf{X}_k^* = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^* \mathbf{V}_k \mathbf{\Sigma}_k \mathbf{U}_k^* = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^*$. Throughout this paper, we

assume that all Fourier realizations of the flow are linearly independent, which is the case for any truly random process. In the final step, the SPOD modes ϕ and modal energies σ are found as

$$\Phi_k = \mathbf{W}^{-\frac{1}{2}} \mathbf{U}_k = [\phi_{k_1}, \phi_{k_2}, \dots, \phi_{k_{n_{\text{blk}}}}] \in \mathbb{R}^{n \times n_{\text{blk}}} \quad (12)$$

$$\text{and } \Sigma_k = \text{diag}(\sigma_{k_1}, \sigma_{k_2}, \dots, \sigma_{k_{n_{\text{blk}}}}) \in \mathbb{R}^{n_{\text{blk}} \times n_{\text{blk}}}, \quad (13)$$

respectively. The weighting of the eigenvectors in equation (12) guarantees orthonormality

$$\Phi_k^* \mathbf{W} \Phi_k = \mathbf{I} \quad (14)$$

under the inner product (8).

3. Streaming SPOD

Our goal is to develop an algorithm that recursively updates the d most energetic SPOD modes for each frequency from a streaming dataset. We require the algorithm to converge a fixed number of modes d to be able to operate within a strictly limited amount of memory. We start by adapting Brand's [13] incremental SVD algorithm to the special case of updating the eigendecomposition of the estimated CSD matrix. The best rank- d approximation used to truncate the eigenbasis and the initialization of the algorithm are discussed later in §3.1 and §3.2, respectively. A graphical illustration of the streaming algorithm is shown in figure 2.

The block-wise sample mean is readily updated through the recursive relation

$$\bar{\mathbf{q}}^{(m)} = \frac{m-1}{m} \bar{\mathbf{q}}^{(m-1)} + \frac{1}{m} \left[\frac{1}{n_{\text{freq}}} \sum_{j=1}^{n_{\text{freq}}} \mathbf{q}_j^{(m)} \right]. \quad (15)$$

Analogously, a rank-1 update of the CSD takes the form

$$\mathbf{S}_k^{(m)} = \frac{m-1}{m} \mathbf{S}_k^{(m-1)} + \frac{m-1}{m^2} \mathbf{x}_k^{(m)} \mathbf{x}_k^{*(m)} \quad (16)$$

and can be performed once the m -th Fourier realization $\hat{\mathbf{q}}_k^{(m)}$ becomes available. Note that we use the sample CSD as an unbiased estimator for the unknown

Figure 2: Illustration of the streaming SPOD algorithm. Numbers in parentheses denote the equations. As soon as a new block of data becomes available, the Fourier transform is computed and the old eigenbases for each frequency are augmented by the orthogonal complement from the new data. The basis rotation and truncation conclude the update.

population CSD. The Fourier transform requires storage of n_{freq} snapshots during runtime, preferably in memory. The computational requirements of the batch and streaming algorithm are compared in more detail in §7. The update formula for the CSD, equation (16), can be rewritten in terms of the data matrix $\mathbf{X}_k$ as

$$\mathbf{X}_k^{(m)} \mathbf{X}_k^{*(m)} = \frac{m-1}{m} \mathbf{X}_k^{(m-1)} \mathbf{X}_k^{*(m-1)} + \frac{m-1}{m^2} \mathbf{x}_k^{(m)} \mathbf{x}_k^{*(m)} \quad (17)$$

by using definition (9). Analogous to equation (7), we denote by

$$\mathbf{X}_k^{(m)} = [\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \dots, \mathbf{x}_k^{(m)}] = \frac{1}{\sqrt{m}} \mathbf{W}^{\frac{1}{2}} \hat{\mathbf{Q}}_k^{(m)} \in \mathbb{R}^{n \times m} \quad (18)$$

the data matrix containing the first m weighted Fourier realizations at the k -th frequency. We now insert the SVD of the data matrix

$$\mathbf{X}_k^{(m)} = \mathbf{U}_k^{(m)} \boldsymbol{\Sigma}_k^{(m)} \mathbf{V}_k^{*(m)} \quad (19)$$

into the update formula (17) to obtain an updating scheme

$$\mathbf{U}_k^{(m)} \boldsymbol{\Sigma}_k^{2(m)} \mathbf{U}_k^{*(m)} = \frac{m-1}{m} \mathbf{U}_k^{(m-1)} \boldsymbol{\Sigma}_k^{2(m-1)} \mathbf{U}_k^{*(m-1)} + \frac{m-1}{m^2} \mathbf{x}_k^{(m)} \mathbf{x}_k^{*(m)} \quad (20)$$

for the eigendecomposition of the CSD at iteration level m in terms of the eigendecomposition at level $m-1$ and the newly arrived data $\mathbf{x}_k^{(m)}$. For brevity, we factorize equation (20) and consider the data matrix

$$\mathbf{X}_k^{(m)} = \left[\sqrt{\frac{m-1}{m}} \mathbf{U}_k^{(m-1)} \boldsymbol{\Sigma}_k^{(m-1)} \mathbf{V}_k^{*(m-1)} \quad \sqrt{\frac{m-1}{m^2}} \mathbf{x}_k^{(m)} \right] \quad (21)$$

$$= \begin{bmatrix} \mathbf{U}_k^{(m-1)} & \mathbf{x}_k^{(m)} \end{bmatrix} \begin{bmatrix} \sqrt{\frac{m-1}{m}} \boldsymbol{\Sigma}_k^{(m-1)} \mathbf{V}_k^{*(m-1)} & \mathbf{0} \\ \mathbf{0} & \sqrt{\frac{m-1}{m^2}} \end{bmatrix} \quad (22)$$

instead of the product $\mathbf{X}_k^{(m)} \mathbf{X}_k^{*(m)}$. With the goal in mind to update $\mathbf{U}_k^{(m-1)}$ with the new data $\mathbf{x}_k^{(m)}$, equation (21) is factored into the matrix product given by equation (22). We seek to find the updated set of left singular vectors $\mathbf{U}_k^{(m)}$ in the column space of the augmented eigenbasis $\begin{bmatrix} \mathbf{U}_k^{(m-1)} & \mathbf{x}_k^{(m)} \end{bmatrix}$ and start by restoring orthonormality. The component of $\mathbf{x}_k^{(m)}$ that is orthogonal to $\mathbf{U}_k^{(m-1)}$ can readily be found from a partial step of the modified Gram-Schmidt (MGS) algorithm as

$$\mathbf{u}_k^{\perp(m)} = \mathbf{x}_k^{(m)} - \mathbf{U}_k^{(m-1)} \mathbf{U}_k^{*(m-1)} \mathbf{x}_k^{(m)}. \quad (23)$$

Using equation (23), the multiplicand is recast into a product of a modified multiplicand $\begin{bmatrix} \mathbf{U}_k^{(m-1)} & \frac{\mathbf{u}_k^{\perp(m)}}{\|\mathbf{u}_k^{\perp(m)}\|} \end{bmatrix}$ with orthonormal columns and a matrix as

$$\begin{bmatrix} \mathbf{U}_k^{(m-1)} & \mathbf{x}_k^{(m)} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_k^{(m-1)} & \frac{\mathbf{u}_k^{\perp(m)}}{\|\mathbf{u}_k^{\perp(m)}\|} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{U}_k^{*(m-1)} \mathbf{x}_k^{(m)} \\ \mathbf{0} & \|\mathbf{u}_k^{\perp(m)}\| \end{bmatrix}. \quad (24)$$

Inserting equation (24) into equation (22) yields the expression

$$\begin{aligned} \mathbf{X}_k^{(m)} &= \begin{bmatrix} \mathbf{U}_k^{(m-1)} & \frac{\mathbf{u}_k^{\perp(m)}}{\|\mathbf{u}_k^{\perp(m)}\|} \end{bmatrix} \\ &\quad \times \begin{bmatrix} \mathbf{I} & \mathbf{U}_k^{*(m-1)} \mathbf{x}_k^{(m)} \\ \mathbf{0} & \|\mathbf{u}_k^{\perp(m)}\| \end{bmatrix} \begin{bmatrix} \sqrt{\frac{m-1}{m}} \boldsymbol{\Sigma}_k^{(m-1)} \mathbf{V}_k^{(m-1)} & \mathbf{0} \\ \mathbf{0} & \sqrt{\frac{m-1}{m^2}} \end{bmatrix} \end{aligned} \quad (25)$$

for the updated data matrix. Multiplying equation (25) with its conjugate transpose yields the updated CSD

$$\mathbf{X}_k^{(m)} \mathbf{X}_k^{*(m)} = \begin{bmatrix} \mathbf{U}_k^{(m-1)} & \frac{\mathbf{u}_k^{\perp(m)}}{\|\mathbf{u}_k^{\perp(m)}\|} \end{bmatrix} \mathbf{M} \begin{bmatrix} \mathbf{U}_k^{*(m-1)} \\ \frac{\mathbf{u}_k^{\perp(m)}}{\|\mathbf{u}_k^{\perp(m)}\|} \end{bmatrix}, \quad (26)$$

where

$$\mathbf{M} = \frac{m-1}{m^2} \begin{bmatrix} m \boldsymbol{\Sigma}_k^{(m-1)^2} + \mathbf{U}_k^{*(m-1)} \mathbf{x}_k^{(m)} \mathbf{x}_k^{*(m)} \mathbf{U}_k^{(m-1)} & \|\mathbf{u}_k^{\perp(m)}\| \mathbf{U}_k^{*(m-1)} \mathbf{x}_k^{(m)} \\ \|\mathbf{u}_k^{\perp(m)}\| \mathbf{x}_k^{*(m)} \mathbf{U}_k^{(m-1)} & \|\mathbf{u}_k^{\perp(m)}\|^2 \end{bmatrix} \quad (27)$$

is a $m \times m$ Hermitian matrix. The remaining task is to recast the right-hand side of equation (26) into SVD form. This is achieved through an eigendecomposition $\mathbf{M} = \tilde{\mathbf{U}} \tilde{\boldsymbol{\Sigma}}^2 \tilde{\mathbf{U}}^*$ of $\mathbf{M}$. Equivalently, we may factor $\mathbf{M}$ as $\mathbf{M} = \mathbf{K} \mathbf{K}^*$ first, where

$$\mathbf{K} = \sqrt{\frac{m-1}{m^2}} \begin{bmatrix} \sqrt{m} \boldsymbol{\Sigma}_k^{(m-1)} & \mathbf{U}_k^{*(m-1)} \mathbf{x}_k^{(m)} \\ \mathbf{0} & \|\mathbf{u}_k^{\perp(m)}\| \end{bmatrix} = \tilde{\mathbf{U}} \tilde{\boldsymbol{\Sigma}} \tilde{\mathbf{V}}^*, \quad (28)$$

and compute the SVD of $\mathbf{K}$. Inserting the eigendecomposition $\mathbf{M} = \mathbf{K} \mathbf{K}^* = \tilde{\mathbf{U}} \tilde{\boldsymbol{\Sigma}}^2 \tilde{\mathbf{U}}^*$ into equation (26) yields

$$\begin{aligned} \mathbf{X}_k^{(m)} \mathbf{X}_k^{*(m)} &= \underbrace{\begin{bmatrix} \mathbf{U}_k^{(m-1)} & \frac{\mathbf{u}_k^{\perp(m)}}{\|\mathbf{u}_k^{\perp(m)}\|} \end{bmatrix}}_{\mathbf{U}_k^{(m)}} \underbrace{\tilde{\mathbf{U}} \tilde{\boldsymbol{\Sigma}} \tilde{\mathbf{U}}^*}_{\boldsymbol{\Sigma}_k^{(m)}} \begin{bmatrix} \mathbf{U}_k^{*(m-1)} \\ \frac{\mathbf{u}_k^{\perp(m)}}{\|\mathbf{u}_k^{\perp(m)}\|} \end{bmatrix} \\ &= \mathbf{U}_k^{(m)} \boldsymbol{\Sigma}_k^{(m)^2} \mathbf{U}_k^{*(m)}. \end{aligned} \quad (29)$$

By noting that $\tilde{\mathbf{U}}$ is a rotation matrix that preserves orthonormality, the spectral theorem guarantees that this decomposition is unique, and therefore corresponds to the updated eigendecomposition of the CSD matrix. The updates of the eigenbasis and eigenvalues hence take the form

$$\mathbf{U}_k^{(m)} = \left[\mathbf{U}_k^{(m-1)} \quad \frac{\mathbf{u}_k^{\perp(m)}}{\|\mathbf{u}_k^{\perp(m)}\|} \right] \tilde{\mathbf{U}} \quad \text{and} \quad (30)$$

$$\boldsymbol{\Sigma}_k^{(m)} = \tilde{\boldsymbol{\Sigma}}, \quad (31)$$

respectively. Besides the rotation (30), the implementation of the algorithm requires the MGS step (23) and the construction and SVD of $\mathbf{K}$, as defined in equation (28). Note that the large matrices $\mathbf{X}_k^{(m)} \mathbf{X}_k^{*(m)}$ and $\mathbf{U}_k^{(m-1)} \mathbf{U}_k^{*(m-1)}$ appearing in the derivation are never computed in the actual algorithm. Up to this point, no approximations have been made.

3.1. Eigenbasis truncation

The recursive rank-1 updates described by equation (30) add an additional vector to the eigenbasis of the CSD matrix at each step. In practice, however, we are interested in converging a fixed number d of the most energetic SPOD modes only. Fortunately, the basic property of the SVD guarantees that this best rank- d approximation is readily obtained by truncating the basis after the d -th vecotor. Formally, we express this by partitioning the updated eigenbasis and matrix of singular values as

$$\mathbf{U}_k^{(m)} = \left[\mathbf{U}_k'^{(m)} \quad \mathbf{u}_{k_{d+1}} \right] \quad \text{and} \quad \boldsymbol{\Sigma}_k^{(m)} = \begin{bmatrix} \boldsymbol{\Sigma}_k'^{(m)} & \mathbf{0} \\ \mathbf{0} & \sigma_{k_{d+1}} \end{bmatrix}, \quad (32)$$

respectively, and letting

$$\mathbf{U}_k^{(m)} \leftarrow \mathbf{U}_k'^{(m)} \quad \text{and} \quad \boldsymbol{\Sigma}_k^{(m)} \leftarrow \boldsymbol{\Sigma}_k'^{(m)} \quad (33)$$

as we update the basis during runtime. At this point, a truncation error is introduced as the vector component $\mathbf{u}_{k_{d+1}}$ that is orthogonal to the retained d eigenvectors is discarded. The batch SPOD algorithm, on the contrary, guarantees that every eigenvector is orthogonal to all other $n_{\text{blk}} - 1$ eigenvectors. We address the error resulting from the basis truncation in §4.

As before, the final step of the algorithm consists of obtaining the SPOD modes by weighting the CSD eigenvectors according to $\Phi_k^{(m)} = \mathbf{W}^{-\frac{1}{2}} \mathbf{U}_k^{(m)} = [\phi_{k_1}^{(m)}, \phi_{k_2}^{(m)}, \dots, \phi_{k_d}^{(m)}] \in \mathbb{R}^{n \times d}$.

3.2. Initialization

Once the first Fourier realization becomes available, the eigenbasis is initialized as $\mathbf{U}_k^{(1)} \leftarrow [\mathbf{x}_k^{(1)}, \mathbf{0}, \dots, \mathbf{0}] \in \mathbb{R}^{n \times d}$ and subsequently updated as $\mathbf{U}_k^{(2)} = [\mathbf{u}_{k_1}^{(2)}, \mathbf{u}_{k_2}^{(2)}, \mathbf{0}, \dots, \mathbf{0}]$ at iteration level $m = 2$, and so on. Correspondingly, the singular value matrix is initialized with the first Fourier realization as $\Sigma_k^{(1)} \leftarrow \text{diag}(\mathbf{x}_k^{*(1)} \mathbf{x}_k^{(1)}, 0, \dots, 0) \in \mathbb{R}^{d \times d}$ before being updated as $\Sigma_k^{(2)} = \text{diag}(\sigma_{k_1}^{(2)}, \sigma_{k_2}^{(2)}, 0, \dots, 0)$. The truncation of the eigenbasis is performed once the iteration level exceeds the number of desired SPOD modes, i.e. when $m \geq d + 1$.

Alternatively, the eigenbasis can be initialized from a previously computed SPOD basis as $\mathbf{U}_k^{(1)} = \mathbf{W}^{\frac{1}{2}} \Phi_k^{\text{old}}$. Initializing the algorithm with an initial SPOD basis Φ_k^{old} obtained from a batch computation or a streaming computation with a larger value d has the benefit of reducing the truncation error. This follows directly from the best rank- d property of the SVD.

4. Errors and convergence

The errors of the approximation can be quantified by comparing the rank- d solutions at the m -th iteration level to the reference solution Φ_k^{batch} and Σ_k^{batch} obtained from the batch algorithm described in §2.

Errors with respect to batch solution. We define two error quantities

$$e_j^{\phi, \text{batch}}(m) = \sum_{k=1}^{n_{\text{freq}}} \left(1 - \max_j \left\langle \phi_{k_j}^{(m)}, \phi_{k_j}^{\text{batch}} \right\rangle_E \right) \quad \text{and} \quad (34)$$

$$e_j^{\lambda, \text{batch}}(m) = \sum_{k=1}^{n_{\text{freq}}} \left| \frac{\lambda_{k_j}^{(m)} - \lambda_{k_j}^{\text{batch}}}{\lambda_{k_j}^{\text{batch}}} \right|. \quad (35)$$

that measure the error in the j -th eigenvector and eigenvalue, respectively. The eigenvector error given by equation (34) is defined in terms of the inner product

(8) and compares the patterns of two modes, i.e. it is 0 for identical and 1 for orthogonal modes. The maximum over the mode rank index is taken to ensure that the most similar modes are compared to each other. This is important as similar modes can swap order between iterations.

Convergence with respect to previous solution. Estimates for the convergence of the eigenvectors and eigenvalues are defined analogously in terms of their values at the previous iteration level $m - 1$. The resulting measures of convergence

$$e_j^{\phi, \text{prev}}(m) = \sum_{k=1}^{n_{\text{freq}}} \left(1 - \max_j \left\langle \phi_{k_j}^{(m)}, \phi_{k_j}^{(m-1)} \right\rangle_E \right) \text{ and} \quad (36)$$

$$e_j^{\lambda, \text{prev}}(m) = \sum_{k=1}^{n_{\text{freq}}} \left| \frac{\lambda_{k_j}^{(m)} - \lambda_{k_j}^{(m-1)}}{\lambda_{k_j}^{(m-1)}} \right|, \quad (37)$$

for the j -th eigenfunction and eigenvalue, respectively, can be monitored during runtime.

5. Examples

This section demonstrates the performance of the proposed streaming SPOD algorithm on two examples. The first example is a high-fidelity numerical simulation of a turbulent jet [19], and the second example optical flow obtained from a high-speed movie of a stepped spillway experiment [20, 21]. An overview of the databases and SPOD parameters is presented in table 1. The SPOD parameters are chosen according to best practice. A discussion of how to choose them is beyond the scope of this work. The same applies to detailed physical interpretations of the results. Here, we focus on the performance and convergence of the streaming algorithm as compared to its offline counterpart. We use a Hanning window for the Fourier transformation and set the number of retained SPOD modes to $d = 5$. The effect of eigenbasis truncation is discussed in more detail in §6.

Case	Database				SPOD parameters			
	$\mathbf{q}$	n_{x_1}	n_{x_2}	n_t	n_{freq}	n_{ovlp}	n_{blk}	d
Jet	p^{symm}	175	39	10,000	256	128	78	5
Spillway	$[u, v]$	224	160	19,782	512	256	77	5

Table 1: Parameters for the two example databases and the SPOD. The spectral estimation parameters n_{freq} , n_{ovlp} and n_{blk} are identical for batch and streaming SPOD. d is the number of desired modes for the streaming algorithm.

5.1. Example 1: large eddy simulation of a turbulent jet

The turbulent jet is a typical examples of a stationary flow. A number of studies, see e.g. [22] for an early experimental and [23] for a recent numerical example, use SPOD to analyse jet turbulence. Our first example is a LES of a Mach 0.9 jet at a jet diameter-based Reynolds number of $1.01 \cdot 10^6$ [19]. The LES was calculated using the unstructured flow solver “Charles” [24]. The database consists of 10,000 snapshots of the symmetric component of the pressure field obtained as the zeroth azimuthal Fourier component of the flow. We choose to resolve 129 positive frequencies by setting $n_{\text{freq}} = 256$. Each block therefore consists of 256 snapshots. We further use a 50% overlap by letting $n_{\text{ovlp}} = 128$. This results in a total of 77 blocks for the spectral estimation, each of which represents one realization of the temporal Fourier transform. The first snapshot of the database is visualized in figure 3. The chaotic nature of the flow becomes apparent at first glance.

Figure 4(a) shows the batch SPOD spectrum obtained for the spectral estimation parameters listed in table 1. Each line represents the energy spectrum associated with a single mode index j . The total number of modes is equal to the number of blocks, i.e. $n_{\text{blk}} = 77$ in this example. Most of the energy is concentrated in the large-scale structures that dominate at low frequencies. The roll-off of the distribution at higher frequencies is indicative of an energy cascade that transfers anergy from larger to smaller scales. Over a certain frequency interval $0.1 \lesssim f \lesssim 0.6$, the first mode is significantly more energetic than the other

Figure 3: Fluctuating pressure of the first snapshot of the turbulent jet LES: (a) streamwise plane; (b) symmetric pressure component only. The boundary layer inside the nozzle is turbulent, whereas the flow inside the potential core is laminar. The potential core collapses after approximately 5 jet diameters.

Figure 4: SPOD energy spectra of the turbulent jet obtained using batch SPOD and streaming SPOD: (a) all $n_{\text{blk}} = 77$ eigenvalues computed using batch SPOD (—); (b) $d = 5$ leading eigenvalues calculated using streaming SPOD (o). The batch solution (—) is shown for comparison. j indicates the mode index from black ($j = 1$, most energetic) to light gray ($j = n_{\text{blk}}$ in (a) and $j = d$ in (b), least energetic).

Figure 5: Side-by-side comparison of SPOD modes calculated using batch SPOD (left column) and streaming SPOD (right column) for the jet example.

modes. This low-rank behavior has important physical implications discussed elsewhere [25, 26]. The spectra of the five leading modes are replicated in figure 4(b) and compared to the results obtained using streaming SPOD ($\circ$ symbols). It can be seen that both results are almost indistinguishable. This provides a first indication that the streaming SPOD algorithm accurately approximates the SPOD eigenvalues. We will quantify this observation in the context of figure 6 below.

After establishing that the modal energies are well approximated by the streaming algorithm, we next examine the modal structures. Figure 5 shows a side-by-side comparison of the first ($j = 1$), third ($j = 3$) and fifth ($j = 5$) modes at two representative frequencies ($f = 0.1$, top half and $f = 0.6$, bottom half). The leading modes (first and fourth row) computed using streaming SPOD are almost indistinguishable from the reference solution for both frequencies.

The third modes (second and fifth row) still compare well. More differences become apparent for the fifth modes. It has to be kept in mind though, that the subdominant modes are in general more difficult to converge. This exemplifies the importance of being able to converge second-order statistics from long data sequences.

In figure 6, we next investigate the errors and convergence behavior for the jet example in terms of the quantities defined in equations (34)-(37). The eigenvalue error in figure 6(a) drops by approximately one order of magnitude from beginning to end. As anticipated from figure 4(b), the eigenvalue error is generally small, i.e. below the per mil range after the first iteration. The eigenvalue convergence is addressed in figure 6(b). Starting from the end of the initialization phase (gray shaded area), the convergence measure drops by about two orders of magnitude. The error and convergence of the eigenvectors are investigate in figure 6(c) and 6(d), respectively. Is is observed that the eigenvector error drops monotonically for all five modes. The similarity of the leading batch and streaming SPOD modes previously seen in figure 5 is reflected by the small error of 0.6%. Similarly, the differences in the fifth modes result in a 25% error according to the metric. Since the eigenvalue is accurately predicted at the same time, we conclude that this large error is primarily a result of the slow statistical convergence of the subdominant modes. The inset in figure 6(d) exemplifies this slow convergence. After about 50 iterations, the convergence rates of most modes are below 1%.

5.2. Example 2: optical flow of a stepped spillway

The second example is that of a laboratory stepped spillway [20]. Stepped spillways are hydraulic structures designed to control flow release and to achieve high energy dissipation. The two-phase flow of the laboratory spillway is filmed using a high-speed camera and an optical flow algorithm [27, 21] was used to estimate the streamwise and normal velocity components of the air-water mixture. The parameters of the optical flow database are summarized in table 1.

Figure 6: Streaming SPOD error and convergence for the turbulent jet: (a) eigenvalue error; (b) eigenvalue convergence; (c) eigenvector error; (d) eigenvector convergence. The magenta lines show the cumulative error in (a,c) and the mean of the convergence metric in (b,d), respectively. The shaded area demarcates the initial region $1 \leq m \leq d + 1$ in which the eigenbasis is still rank deficient.

Figure 7: First snapshot of the stepped spillway: (a) raw video frame; (b) streamwise velocity computed using optical flow. The air-water flow is characterized by instability growth, air entrainment and strong turbulence.

Figure 7 shows an example of the raw video data and a processed snapshot of the instantaneous streamwise velocity component. As for the jet example, we will not address the complex multi-phase physics of the setup, but focus on the performance of the streaming SPOD algorithm instead. We chose the spillway as a second example to investigate the algorithm’s performance under high noise conditions. The high noise level of the measurement is apparent in figure 7(b).

As before, spectra obtained using batch SPOD and the streaming version are compared in figure 8. It is observed that the modal energies asymptote towards a constant value for $f \gtrsim 200$. The plateau seen at these frequencies indicates the noise floor of the measurement. An inspection of the SPOD modes confirms this conjecture. Modes in this region are dominated by noise and show no apparent structure (not shown).

The comparison in figure 9 shows that the SPOD modes computed using the streaming algorithm closely resemble their batch SPOD counterparts. At the lower frequency (left), the SPOD modes are comprised of surface waves and oscillations of the shear-layer between the step ridges. Surface waves are the dominant structures at higher frequencies (right). Increasingly high noise levels

Figure 8: Same as figure 4, but for the spillway example.

Figure 9: Side-by-side comparison of SPOD modes calculated using batch SPOD (first and third column) and streaming SPOD (second and fourth column) for the spillway example.

Figure 10: Same as figure 6 but for the spillway example.

are observed in the less energetic modes, in particular for the higher frequency case.

The eigenvalue error is studied in figure 10(a). Initially, the error is significantly larger as compared to the turbulent jet case shown in figure 6(a). Subsequently, a faster drop-off allows the eigenvalue error to recover values similar to those found for the jet example. The eigenvalue convergence behavior shown in figure 10(b) is very similar to that of the jet example.

The eigenvector error and convergence are plotted in figure 10(c) and 10(d), respectively. Both occasionally metrics undergo rapid changes, most prominently at iteration level $m = 45$. Sudden drops in the error are directly associated with peaks in the convergence measure. This behavior occurs when an

Figure 11: Eigenvalue (left column) and eigenvector (right column) errors of the first SPOD mode for different numbers of basis vectors d : (a,b) turbulent jet; (c,d) spillway.

eigenvector in the truncated basis gets replaced by a different structure. The re-orthogonalization of the eigenbasis after such an event leads a better correspondence with the batch solution. At the same time, the convergence measure spikes as a result of the change in modal structure. The error ranges between 10% (first mode) and 44% (fifth mode). The similarity of the modes depicted in figure 9 and the lower errors in the jet case, as seen in figure 6, strongly suggest that these relatively high errors are mainly associated with measurement noise.

6. Effect of eigenbasis truncation

Spectral estimation parameters aside, the desired number of SPOD modes d is the only additional user input required by the streaming algorithm. The basis truncation inevitably leads to an approximation error that originates from discarding the vector component orthogonal to the span of the existing basis vectors $\mathbf{U}_k^{(m-1)}$.

Figure 12: Eigenvalue (left column) and eigenvector (right column) errors of the fifth SPOD mode for different numbers of basis vectors d : (a,b) turbulent jet; (c,d) spillway.

Figure 11 compares eigenvalue and eigenvector errors of the first SPOD mode for four different values of d . The jet and spillway examples are shown in figure 11(a,b) and 11(c,d), respectively. It is observed that even restricting the basis to a single vector, i.e. the most aggressive truncation possible, does not lead to significant errors. For $d \geq 5$, all error metrics shown in figure 11(a-c) are almost indistinguishable. Small differences are observed in the eigenvector error for the spillway example. In 11(d), the eigenvector error ranges between 10% for $d = 5$ and 2% for $d = 20$. As discussed in the context of figure 9, this discrepancy is mainly related to data noise.

After establishing that retaining only a few eigenvectors is sufficient to minimize the truncation error, we now focus on the effect of truncation on the suboptimal modes. Analogous to figure 11, we compare the truncation errors of the fifth mode in figure 12. Its error characteristics are similar to the ones of the first mode. This can be seen by comparing figure 12(a,b) to figure 11(a,b). By increasing the basis size to $d = 10$, the final truncation errors are notice-

ably reduced, but adding more vectors does not lead to further reduction of the already low errors. For the spillway case shown in figure 12(c,d), the effect of noise in the data becomes apparent once more. Here, increasing the basis size from $d = 10$ to $d = 25$ reduces both the eigenvalue and eigenvector errors. At the same time, however, the earlier comparison of the mode shapes in figure 9 demonstrated that the coherent large-scale structures are accurately captured, even for $d = 5$.

The truncation error analysis suggests that the basis size d should be chosen somewhat larger than the desired number of modes. It is also important to emphasize that the definitions of the truncation errors rely on the batch solution as a reference, which itself may not be statistically fully converged.

7. Discussion

In this work, we introduce an algorithm that recursively updates the SPOD of large or streaming datasets. In §§4-6, we demonstrate that the algorithm is capable of converging the most energetic SPOD modes while lifting the requirement to store potentially prohibitively large amounts of data.

The batch algorithm requires storage of $n_x \times n_{\text{var}} \times n_t$ data points plus another $n_x \times n_{\text{var}} \times \frac{n_{\text{freq}}}{2} + 1 \times n_{\text{blk}}$ points for the spectral estimation of a real signal. In its simplest implementation, all data is loaded into memory simultaneously. If the data too large to be stored in memory, the n_{blk} temporal Fourier transforms can be computed a priori and stored, fully or partly, on hard drive, and then be reloaded and processed frequency by frequency. The drawbacks of this approach are the significantly longer computing time resulting from the read/write operations, and the additional storage requirements. For higher-dimensional data, e.g. with $n_{\text{dim}} \geq 2$, snapshots totaling multiple terabytes are likely to be required to converge the second-order statistics, in particular those of subdominant modes. In such cases, batch SPOD may become computationally intractable altogether.

The streaming SPOD algorithm, on the other hand, has a much lower storage

requirement of $n_x \times n_{\text{var}} \times n_{\text{freq}} + n_x \times n_{\text{var}} \times \frac{n_{\text{freq}}}{2} + 1 \times d$ data points for $\mathbf{X}_k^{(m)}$ and $\mathbf{U}_k^{(m)}$, respectively, plus a number of small fields that do not scale with the large dimensions in space and time. Ideally, $\mathbf{X}_k^{(m)}$ and $\mathbf{U}_k^{(m)}$ are stored and updated in memory during runtime. Besides its low storage requirements, the algorithm achieves computational efficiency by employing MGS steps for the orthogonalization.

A useful implication of the ergodicity assumption is that it offsets the need to store a single long time-series. In §3, we used overlapping blocks to increase the number of Fourier samples in cases where the total number of snapshots is limited. A quite different scenario occurs when dealing with fast data streams. In such a scenario, we can take advantage of the fact that ergodicity permits arbitrarily long gaps between sampling periods of two consecutive data blocks $\mathbf{X}_k^{(m)}$ and $\mathbf{X}_k^{(m+1)}$. This is advantageous in experimental setups such as time-resolved particle image velocimetry (TR-PIV). It suffices to store n_{freq} consecutive snapshots at a time, perform the computationally costly cross-correlations to obtain velocity data, and update the SPOD eigenbasis before continuing to sample data. This procedure allows, in principle, to converge second-order flow statistics over arbitrarily long time horizons.

Acknowledgements. Support of the Office of Naval Research grant No. N00014-16-1-2445 with Dr. Knox Millsaps as program manager is gratefully acknowledged. Special thanks are due to Aaron Towne and Patrick Vogler for reviewing the manuscript and sharing their insights, and to Tim Colonius, Andres Goza and Matthias Kramer for making valuable comments. The author gratefully acknowledges Matthias Kramer and Hubert Chanson for providing the experimental optical flow data. The experiments were undertaken in the hydraulics laboratory at the University of Queensland. The LES study was supported by NAVAIR SBIR project, under the supervision of Dr. John T. Spyropoulos. The main LES calculations were carried out on CRAY XE6 machines at DoD HPC facilities in ERDC DSRC.

References

References

- [1] K. Taira, S. L. Brunton, S. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, L. S. Ukeiley, Modal analysis of fluid flows: An overview, *AIAA Journal* doi:<https://doi.org/10.2514/1.J056060>.
- [2] C. W. Rowley, S. T. Dawson, Model reduction for flow analysis and control, *Annual Review of Fluid Mechanics* 49 (2017) 387–417.
- [3] L. Sirovich, Turbulence and the dynamics of coherent structures, *Quarterly of applied mathematics* 45 (3) (1987) 561–571.
- [4] N. Aubry, P. Holmes, J. L. Lumley, E. Stone, The dynamics of coherent structures in the wall region of a turbulent boundary layer, *Journal of Fluid Mechanics* 192 (1988) 115–173.
- [5] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, *Journal of Fluid Mechanics* 497 (2003) 335–363.
- [6] J. L. Lumley, *Stochastic tools in turbulence*, Academic Press, New York, 1970.
- [7] A. Towne, O. T. Schmidt, T. Colonius, Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis, arXiv:1708.04393, submitted to *Journal of Fluid Mechanics*.
- [8] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *Journal of Fluid Mechanics* 656 (2010) 5–28. doi:10.1017/S0022112010001217.
- [9] M. S. Hemati, M. O. Williams, C. W. Rowley, Dynamic mode decomposition for large and streaming datasets, *Physics of Fluids* 26 (11) (2014) 111701.

- [10] H. Zhang, C. W. Rowley, E. A. Deem, L. N. Cattafesta, Online dynamic mode decomposition for time-varying systems, arXiv preprint arXiv:1707.02876.
- [11] P. Businger, Updating a singular value decomposition, BIT 10 (3) (1970) 376–385.
- [12] R. D. De Groat, R. A. Roberts, SVD update algorithms and spectral estimation applications, in: Circuits, Systems and Computers, 1985. Nineteenth Asilomar Conference on, IEEE, 1985, pp. 601–605.
- [13] M. Brand, Fast low-rank modifications of the thin singular value decomposition, Linear algebra and its applications 415 (1) (2006) 20–30.
- [14] M. Brand, Incremental singular value decomposition of uncertain data with missing values, Proc. Eur. Conf. Computer Vision (2002) 707–720.
- [15] M. Brand, Fast online svd revisions for lightweight recommender systems, in: Proceedings of the 2003 SIAM International Conference on Data Mining, SIAM, 2003, pp. 37–46.
- [16] P. D. Turney, P. Pantel, From frequency to meaning: Vector space models of semantics, Journal of artificial intelligence research 37 (2010) 141–188.
- [17] T. Braconnier, M. Ferrier, J.-C. Jouhaud, M. Montagnac, P. Sagaut, Towards an adaptive pod/svd surrogate model for aeronautic design, Computers & Fluids 40 (1) (2011) 195–209.
- [18] O. M. Solomon, Jr., PSD computations using Welch’s method, NASA STI/Recon Technical Report N 92.
- [19] G. Brès, P. Jordan, M. Le Rallic, V. Jaunet, A. V. G. Cavalieri, A. Towne, S. Lele, T. Colonius, O. T. Schmidt, Importance of the nozzle-exit boundary-layer state in subsonic turbulent jets, submitted to Journal of Fluid Mechanics.

- [20] M. Kramer, H. Chanson, Transition Flow Regime on Stepped Spillways: Air-Water Flow Characteristics and Step-Cavity Fluctuations, *Environmental Fluid Mechanics* (under review).
- [21] G. Zhang, H. Chanson, Application of local optical flow methods to high-velocity free-surface flows: Validation and application to stepped chutes, *Experimental Thermal and Fluid Science* 90 (July 2017) (2017) 186–199.
- [22] M. N. Glauser, S. J. Leib, W. K. George, Coherent structures in the axisymmetric turbulent jet mixing layer, *Turbulent shear flows* 5 (1987) 134–145.
- [23] O. T. Schmidt, A. Towne, T. Colonius, A. V. G. Cavalieri, P. Jordan, G. A. Brès, Wavepackets and trapped acoustic modes in a turbulent jet: coherent structure eduction and global stability, *Journal of Fluid Mechanics* 825 (2017) 11531181. doi:10.1017/jfm.2017.407.
- [24] G. A. Brès, F. E. Ham, J. W. Nichols, S. K. Lele, Unstructured large-eddy simulations of supersonic jets, *AIAA Journal* 55 (4) (2017) 1164–1184.
- [25] O. T. Schmidt, T. Colonius, G. A. Brès, Linear dynamics of large-scale structures in turbulent jets, in: *Tenth International Symposium on Turbulence and Shear Flow Phenomena*, 2017.
- [26] O. T. Schmidt, A. Towne, G. Rigas, T. Colonius, G. A. Brès, Spectral analysis of jet turbulence, arXiv:1711.06296 (submitted to *J. Fluid Mech.*). URL <https://arxiv.org/abs/1711.06296>
- [27] T. Liu, A. Merat, M. H. M. Makhmalbaf, C. Fajardo, P. Merati, Comparison between optical flow and cross-correlation methods for extraction of velocity fields from particle images, *Experiments in Fluids* 56 (8) (2015) 1–23.