Journal of Computational Physics 380 (2019) 355–377

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A parallel and streaming Dynamic Mode Decomposition algorithm with finite precision error analysis for large data

Sreevatsa Anantharamu, Krishnan Mahesh [*]

Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 12 May 2018
Received in revised form 9 October 2018
Accepted 5 December 2018
Available online 4 January 2019

Keywords:
DMD
Full-Orthogonalization-Arnoldi
Error analysis
Parallel
Large data

ABSTRACT

A novel technique based on the Full Orthogonalization Arnoldi (FOA) is proposed to perform Dynamic Mode Decomposition (DMD) for a sequence of snapshots. A modification to FOA is presented for situations where the matrix A is unknown, but the set of vectors $\{A^{i-1}v_1\}_{i=1}^{N}$ are known. The modified FOA is the kernel for the proposed projected DMD algorithm termed, FOA based DMD. The proposed algorithm to compute DMD modes and eigenvalues i) does not require Singular Value Decomposition (SVD) for snapshot matrices X with $\kappa_2(X) \ll 1/\epsilon_m$, where $\kappa_2(X)$ is the 2-norm condition number of the snapshot matrix and ϵ_m is the relative round-off error or machine epsilon, ii) has an optional rank truncation step motivated by round off error analysis for snapshot matrices X with $\kappa_2(X) \approx 1/\epsilon_m$, iii) requires only one snapshot at a time, thus making it a 'streaming' method even with the optional rank truncation step, iv) consumes less memory and requires less floating point operations to obtain the projected matrix than existing projected DMD methods and v) lends itself to easy parallelism as the main computational kernel involves only vector additions, dot products and matrix vector products. The new technique is therefore well-suited for DMD of large datasets on parallel computing platforms. We show both theoretically and using numerical examples that for FOA based DMD without rank truncation, the finite precision error in the computed projection of the linear mapping is $O(\epsilon_m \kappa_2(X))$. The proposed method is also compared to existing projected DMD methods for computational cost, memory consumption and relative round off error. Error indicators are presented that are useful to decide when to stop acquiring new snapshots. The proposed method is applied to several examples of numerical simulations of fluid flow.

1. Introduction

Numerical simulations of both laminar and turbulent flows with $O(10^{5-10})$ degrees of freedom generate high-dimensional datasets comprising of velocity and pressure field at multiple time instants. These datasets can be low-dimensional when expressed in appropriate bases, often termed as 'modes'. The high-dimensional dataset can be represented as a linear combination of few of these modes to reasonable accuracy. A review of different modal decomposition techniques to identify these modes for fluid flows is given in Taira et al. [1] and Rowley and Dawson [2].

Dynamic Mode Decomposition (DMD) is a data-driven modal decomposition technique that identifies a set of modes from multiple snapshots of the observable vectors (defined in section 2). Each of these modes are assigned an eigenvalue

[*] Corresponding author.
E-mail address: kmahesh@umn.edu (K. Mahesh).

https://doi.org/10.1016/j.jcp.2018.12.012
0021-9991/© 2018 Elsevier Inc. All rights reserved.

which denotes growth/decay rate and oscillation frequency of the mode. The obtained modes and corresponding eigenvalues together capture the dynamics of the underlying system. Rowley et al. [3] related the DMD modes and DMD eigenvalues to the eigenfunctions and eigenvalues of the Koopman operator. The Koopman operator [4] is an infinite dimensional linear operator that describes the evolution of linear and nonlinear dynamical systems. These connections make DMD applicable to nonlinear systems such as those governed by the Navier–Stokes equations.

DMD for sequential snapshots uniformly separated in time was first introduced by Schmid and Sesterhenn [5]. Schmid [6] proposed two algorithms which used different choices of basis vectors to perform Galerkin projection of the assumed linear mapping between the snapshots. One algorithm was based on Arnoldi method with no orthogonalization, with the snapshot vectors as the basis vectors, and the other relied on singular value decomposition (SVD) of the sequence of snapshots, and used the resulting left singular vectors as the basis vectors. The SVD based method was seen to have better finite precision accuracy than the Arnoldi based method. Tu et al. [7] modified the SVD version of Schmid [6] proposed for a sequence of snapshots, to consider snapshot pairs. They also proposed Exact DMD where the DMD modes and eigenvalues are defined as the eigendecomposition of the minimum Frobenius norm mapping that relates the snapshot pairs in a least-squares sense. This essentially involved a new definition of the DMD modes from the eigenvalues and eigenvectors of the projected linear mapping. The drawback of the SVD based DMD method discussed in Schmid [6] and Tu et al. [7] (both projected and exact DMD) is that it requires access to all snapshots at once to compute the projected linear mapping. Hemati et al. [8] proposed a streaming version of Tu et al. [7] to process large and streaming datasets which requires access to only the current snapshot pair to compute the projected linear mapping, and also a compression procedure to reduce the effect of noise in the dataset on the DMD modes. However, as shown in this paper through numerical experiments, the finite precision error in computing the projected linear mapping from the streaming version of Hemati et al. [8] without compression can be large. Two DMD algorithms that are the same in theory might can have very different finite precision error. It is therefore important to have an estimate of finite precision error of the computed projection while devising DMD algorithms. A parallel version of the DMD algorithm proposed in Schmid [6] was presented in Sayadi and Schmid [9]. Scaling of the algorithm was shown upto 1024 processors. However, this method is not streaming as it requires access to all snapshot vectors to compute the projected linear mapping. The underlying parallel algorithm used the TSQR algorithm of Demmel et al. [10] to compute QR factorization of the snapshot vectors followed by SVD of a small upper triangular matrix. Several other variants of DMD apart from projected DMD and Exact DMD [7] include optimized DMD [11], extended DMD [12], kernel DMD [13], noise corrected DMD [14], forward–backward DMD [14], total least-squares DMD [14], recursive DMD [15], sparsity promoting DMD [16] and optimal mode decomposition [17].

Numerical experiments on the error due to the number of snapshots and spatial resolution were performed by Duke et al. [18] for synthetically generated data with noise, and that due to the choice of observables is presented in Zhang et al. [19]. Approximate solution to eigenvalue problems using Arnoldi's method and Galerkin projection [20] also provide estimates for the error associated with each eigenvector and eigenvalue pair. We utilize these results which have been used for numerical solution of eigenvalue problems in the context of DMD where the linear mapping is not explicitly known.

The contribution of this paper is a novel parallel streaming DMD algorithm suitable for large data along with its finite precision error analysis. DMD is inherently not a backward stable (refer Higham [21] for more on backward stability) procedure to compute projection of linear mapping using orthonormal basis vectors. This is because it relies on the snapshot vectors and not on the knowledge of linear mapping A. This leads to dependence of the finite precision error in the computed projection on the condition number of the snapshot matrix. Also, this warrants care in the design of DMD algorithms which might otherwise lead to dependence on higher powers of condition number and subsequently large finite precision error. In this paper, we examine the finite precision arithmetic properties of the proposed and few other existing DMD methods. The proposed method allows quantitative estimates of whether additional snapshots are needed, and the finite precision error contributions to the estimated modes. These properties are obtained in a readily parallelizable and streaming formulation, which makes reliable DMD representation of large datasets possible.

The proposed method is derived from the Full Orthogonalization Arnoldi (FOA) procedure used to compute the projection of a given linear mapping A onto the Krylov subspace. However, the FOA procedure explicitly requires the knowledge of operating A onto the successively generated Arnoldi vectors v_j. We avoid this by reformulating FOA such that in the jth step, the new method relies only on the additional knowledge of $A^j\psi_1$ to compute Av_j, where ψ_1 is a random starting vector used in FOA. This results in a streaming DMD algorithm which requires access to only the current snapshot vector $\psi_{j+1} := A^j\psi_1$.

The paper is organized as follows. Section 2, discusses the proposed method in the broader context of projected DMD methods. Section 3 presents the proposed FOA based DMD. First, the algorithm is derived in batch processed form by drawing parallels to the Rayleigh–Ritz procedure using FOA in section 3.1. Then, the method is recast in streaming form. Properties of the algorithm which include computational cost, memory consumption and finite precision error in computation of the projected linear mapping are discussed and compared to exiting projected DMD algorithms in sections 3.2.1 and 3.2.2. Error indicators for DMD eigenvalues and eigenvectors computed using FOA based DMD and their finite precision quality are discussed in section 3.2.3. Parallel scaling of the proposed algorithm is presented in section 3.2.5. Section 4 demonstrates FOA based DMD algorithm for three test cases: i) linearized channel flow at $Re = 10000$, ii) flow over a circular cylinder at $Re = 100$ and iii) jets in cross-flow at two different jet velocity to cross flow velocity ratios.

2. Projected DMD methods: background

Let $x_j \in \mathbb{C}^n$ be the state of system at time t_j. The states x_j are assumed to be equally spaced in time, i.e. $t_j = (j-1)\Delta t$ where Δt is the time separation between successive states. Let $\psi(x) := [\psi^1(x) \quad \psi^2(x) \quad \dots \quad \psi^M(x)]^T$ be a vector of functions i.e. each $\psi^i(x) : \mathbb{C}^n \to \mathbb{C}$. Each $\psi^i(x)$ is termed an observable [3], and $\psi(x)$ is termed vector of observables. The snapshot vector $\psi_j \in \mathbb{C}^M := \psi(x_j)$, i.e. $\psi_j = [\psi^1(x_j) \quad \psi^2(x_j) \quad \dots \quad \psi^M(x_j)]^T$. Let $X_i^j \in \mathbb{C}^{M\times(j-i+1)}$ be the snapshot matrix formed by stacking the snapshot vectors from time t_i to t_j:

$$X_i^j := [\psi_i \quad \psi_{i+1} \quad \dots \quad \psi_{j-1} \quad \psi_j]. \tag{1}$$

A common state vector is the velocity at all points in the domain of fluid simulation and a common vector of observables is the state vector itself. Let N denote the total number of snapshots collected. A linear mapping A is assumed to relate the successive snapshot vectors as

$$\psi_{i+1} = A\psi_i, \quad X_2^N = AX_1^{N-1}. \tag{2}$$

As discussed by Schmid [6], for linear time evolution of the observable vectors, no assumption is involved. For nonlinear time evolution of the observable vectors, for $N \le R+1$ where R is the rank of the snapshot matrix, a linear mapping can always be defined to connect the snapshot vectors over the time interval $[0, N\Delta t]$. However, such a linear mapping may only be an approximation to the nonlinear evolution of the observable vectors. In section 3.2.4, we discuss that the projected linear mapping A contains information of the interpolant through the sampled observable vectors which approximates the nonlinear time evolution of the system. A linear mapping between the successive snapshots implies that the range of matrix X_1^{N-1} is also a Krylov subspace associated with the assumed linear mapping A and the starting vector ψ_1,

$$K_{N-1}(A,\psi_1) := span\{\psi_1, A\psi_1, ..., A^{N-2}\psi_1\}. \tag{3}$$

The columns of X_1^{N-1} can be linearly dependent and need not always form the set of basis vectors for the above Krylov subspace. If they are linearly dependent, then N is modified to include the linearly independent snapshot sequence only.

The Galerkin statement for the approximate eigenvalue problem is to find $\lambda \in \mathbb{C}, v \in K_{N-1}(A,\psi_1)$ pair s.t. $Av - \lambda v$ is orthogonal to all the vectors in the subspace $K_{N-1}(A,\psi_1)$ in the ℓ^2 inner product. Suppose span of columns of $Q \in \mathbb{C}^{M\times N-1}$ and $V \in \mathbb{C}^{M\times N-1}$ are two sets of basis vectors for $K_{N-1}(A,\psi_1)$, then the approximate eigenvector v can be written as Qz for $z \in \mathbb{C}^{N-1}$. The Galerkin problem for $\lambda \in \mathbb{C}, z \in \mathbb{C}^{N-1}$ pair is then the generalized eigenvalue problem,

$$\begin{aligned} V^H(AQz - \lambda Qz) &= 0, \\ V^H AQz &= \lambda V^H Qz. \end{aligned} \tag{4}$$

If $V = Q$, then we obtain,

$$Q^H AQz = \lambda Q^H Qz. \tag{5}$$

Furthermore, if columns of Q are orthonormal, we obtain

$$Q^H AQz = \lambda z. \tag{6}$$

The goal of all projected DMD methods is to compute Q (which may or may not be orthonormal) and $Q^H AQ$ from the snapshot vectors X_1^N alone without knowledge of the linear mapping A. Some possible choices of Q are: matrix of snapshots X_1^{N-1} [5,6,3], left singular vectors of economy SVD of X_1^{N-1} [6,9] and orthonormal matrix from QR factorization of X_1^{N-1} (Hemati et al. [8]).

3. FOA based DMD

3.1. Algorithm

The batch processed form (i.e. all snapshots processed at once) of the proposed FOA based DMD is shown in Algorithm 1. V_1^N is the matrix formed by stacking the set of vectors $\{v_i\}_{i=1}^N$ as columns, and $h_{i,j}$ and $\beta_{i,j}$ are the entries of the matrices $\bar{H}_N$ and β_N respectively. The algorithm takes snapshot matrix X_1^N as input and computes DMD modes and eigenvalues with or without rank truncation. Rank truncation before computing DMD modes reduces the finite precision error in the computed projection of the linear mapping A, and is useful for snapshot matrices with $\kappa_2(X_1^{N-1}) \approx 1/\epsilon_m$, where $\kappa_2(X_1^{N-1})$ is the 2-norm condition number defined as $\|X_1^{N-1}\|_2 \|X_1^{N-1^\dagger}\|_2$. For snapshot matrices with condition number $\kappa_2(X_1^{N-1}) \ll 1/\epsilon_m$, no rank truncation is required before computing the DMD modes and eigenvalues. We use Classical Gram Schmidt (CGS) with reorthogonalization as the orthogonalization kernel in FOA based DMD. This is because parallel implementation

Algorithm 1: FOA based DMD in batch processed form.

```
Collect $N$ snapshots and form $X_1^N$. Create space for matrices $V_1^N \in \mathbb{C}^{M\times N}$, $\bar{H}_N \in \mathbb{C}^{N,N-1}$ and $\beta_N \in \mathbb{C}^{N,N}$ and initialize to 0.
Construct initial vector $v_1$ from the first snapshot, $v_1 := \frac{\psi_1}{\|\psi_1\|_2}$
$\beta_{1,1} = \|\psi_1\|_2$
for $j = 1$ to $N-1$ do
    $\beta_{1,j+1} = h_{1,1:j-1}\beta_{1:j-1,j}$
    for $i = 2$ to $j$ do
        $\beta_{i,j+1} = h_{i,i-1:j-1}\beta_{i-1:j-1,j}$
    end for
    $w = \frac{1}{\beta_{j,j}}\left(\psi_{j+1} - \sum_{i=1}^{j}\beta_{i,j+1}v_i\right)$
    $h_{1:j,j} = V_1^{j^H} w$
    $w = w - V_1^j h_{1:j,j}$
    $s_{1:j} = V_1^{j^H} w$
    $h_{1:j,j} = h_{1:j,j} + s_{1:j}$
    $w = w - V_1^j s_{1:j}$
    $h_{j+1,j} = \|w\|_2$; $v_{j+1} = \frac{w}{h_{j+1,j}}$
    for $i = 1$ to $j+1$ do
        $\beta_{i,j+1} = \beta_{i,j+1} + h_{i,j}\beta_{j,j}$
    end for
end for
Define $H_{N-1} := \bar{H}_N(1:N-1, 1:N-1)$.
if rank truncation then
    Compute SVD of $\beta_{N-1} := \beta_N(1:N-1, 1:N-1)$, i.e., $\beta_{N-1} = U\Sigma W^H$.
    Choose the truncated rank $r$ and $U_r := U(:, 1:r)$ and form $U_r^H H_{N-1} U_r$.
    Compute right eigenvectors $\{z_i\}_{i=1}^r$ and eigenvalues $\{\lambda_i\}_{i=1}^r$ of $U_r^H H_{N-1} U_r$.
    DMD modes are $\{V_1^{N-1} U_r z_i\}$ and DMD eigenvalues are $\{\lambda_i\}_{i=1}^r$.
else
    Compute right eigenvectors $\{z_i\}_{i=1}^{N-1}$ and eigenvalues $\{\lambda_i\}_{i=1}^{N-1}$ of $H_{N-1}$.
    DMD modes are $\{V_1^{N-1} z_i\}_{i=1}^{N-1}$ and DMD eigenvalues are $\{\lambda_i\}_{i=1}^{N-1}$.
end if
```

Algorithm 2: Rayleigh Ritz procedure with Arnoldi's method.

```
1: Given $A$. Create space for matrices $V_1^N \in \mathbb{C}^{M\times N}$ and $\bar{H}_N \in \mathbb{C}^{N,N-1}$.
2: Construct initial vector $v_1$ from the first snapshot. $v_1 := \frac{\psi_1}{\|\psi_1\|_2}$
3: for $j = 1$ to $N-1$ do
4:     $w = Av_j$
5:     $h_{1:j,j} = V_1^{j^H} w$
6:     $w = w - V_1^j h_{1:j,j}$
7:     $s_{1:j} = V_1^{j^H} w$
8:     $h_{1:j,j} = h_{1:j,j} + s_{1:j}$
9:     $w = w - V_1^j s_{1:j}$
10:    $h_{j+1,j} = \|w\|_2$; $v_{j+1} = \frac{w}{h_{j+1,j}}$
11: end for
12: Define $H_{N-1} := \bar{H}_N(1:N-1, 1:N-1)$.
13: Compute right eigenvectors $\{z_i\}_{i=1}^{N-1}$ and eigenvalues $\{\lambda_i\}_{i=1}^{N-1}$ of $H_{N-1}$.
14: Approximate eigenvectors are $\{V_1^{N-1} z_i\}_{i=1}^{N-1}$ and approximate eigenvalues are $\{\lambda_i\}_{i=1}^{N-1}$.
```

of CGS sends lesser number of messages than Modified Gram Schmidt (MGS), and CGS makes use of matrix vector products which are more efficient to compute. Reorthogonalization ensures orthogonality of V_1^N upto machine precision [22].

The batch processed form of FOA based DMD can be derived from the Rayleigh–Ritz procedure using the Arnoldi method shown in Algorithm 2. The Rayleigh–Ritz procedure computes approximate eigenvectors and eigenvalues of linear mapping A. However, we use a modified Arnoldi method where the matrix A is unknown, but the set of vectors $\{\psi_i\}_{i=1}^N$ is known. Steps 2 to 10 of Algorithm 2 constitute the Arnoldi method which generates an orthonormal basis for the Krylov subspace, $K_{N-1}(A, \psi_1)$. The vectors v_i generated from the Arnoldi method, in addition to being orthonormal satisfy the property [20]

$$\begin{aligned} AV_1^{N-1} &= V_1^N \bar{H}_N, \\ AV_1^{N-1} &= V_1^{N-1} H_{N-1} + h_{N,N-1} v_N e_{N-1}^H, \end{aligned} \tag{7}$$

where $H_{N-1} := \bar{H}_N(1:N-1, 1:N-1)$ is the projection of A onto the Krylov subspace $K_{N-1}(A, \psi_1)$, i.e.,

$$V_1^{N-1^H} A V_1^{N-1} = H_{N-1}, \tag{8}$$

and hence the eigenvalues and eigenvectors of H_{N-1} can be used to approximate those of A (steps 13 and 14 of Algorithm 2).

The Rayleigh Ritz procedure with the Arnoldi method cannot be directly used to perform projected DMD, as the step 4 in Algorithm 2 requires knowledge of the assumed linear mapping A, which is not explicitly known before hand in DMD. However, we can compute Av_j by using a linear combination of the snapshot vector ψ_{j+1} and the set of vectors $\{v_i\}_{i=1}^{j}$. The Arnoldi vectors v_i's generated by the above algorithm have the following property [20],

$$span\{v_1,\ldots,v_j\} = span\{\psi_1,\ldots,\psi_j\} = span\{\psi_1, A\psi_1,\ldots,A^{j-1}\psi_1\}. \tag{9}$$

So, ψ_j may be expressed as a linear combination of $\{v_i\}_{i=1}^{j}$,

$$\psi_j = \sum_{i=1}^{j} \beta_{i,j} v_i. \tag{10}$$

In matrix form,

$$\psi_j = V_1^j \beta_{1:j,j}, \tag{11}$$

where V_1^j is the matrix formed by vectors $\{v_i\}_{i=1}^{j}$ as columns. Suppose, we are in the jth iteration of the Arnoldi method. We can develop a method to compute Av_j from knowledge of $\bar{H}_j$, the non-zero entries of jth column of β_N and the Arnoldi relation until the $(j-1)$th iteration using

$$\begin{aligned}\psi_{j+1} = A\psi_j &= AV_1^j\beta_{1:j,j}\\ &= AV_1^{j-1}\beta_{1:j-1,j} + Av_j\beta_{j,j}\\ &= V_1^j\bar{H}_j\beta_{1:j-1,j} + Av_j\beta_{j,j},\end{aligned} \tag{12}$$

where $\bar{H}_j$ is the $j \times (j-1)$ top left submatrix of $\bar{H}_N$. Rearranging,

$$Av_j = \frac{1}{\beta_{j,j}}\left(\psi_{j+1} - v_1\sum_{i=1}^{j-1} h_{1,i}\beta_{i,j} - \sum_{k=2}^{j} v_k \sum_{i=k-1}^{j-1} h_{k,i}\beta_{i,j}\right). \tag{13}$$

Once the jth step of Arnoldi iteration is performed and the non-zero entries of the jth column of $\bar{H}_N$ are computed, the $(j+1)$th column of β_N can be computed using the relation $Av_j = \sum_{i=1}^{j+1} h_{i,j}v_i$ in equation (12) which leads to

$$\psi_{j+1} = V_1^j\bar{H}_j\beta_{1:j-1,j} + V_1^{j+1}h_{1:j+1,j}\beta_{j,j} = V_1^{j+1}\bar{H}_{j+1}\beta_{1:j,j}. \tag{14}$$

The above equation shows that the non-zero entries in the $(j+1)$th column of β_N can be computed as,

$$\begin{aligned}\beta_{1,j+1} &= \sum_{i=1}^{j} h_{1,i}\beta_{i,j},\\ \beta_{k,j+1} &= \sum_{i=k-1}^{j} h_{k,i}\beta_{i,j} \quad ; \quad k = 2,\ldots,j+1.\end{aligned} \tag{15}$$

The steps 1–19 of Algorithm 1 incorporating these modifications, constitutes what we call the modified FOA procedure.

Essentially, we are factorizing X_1^N into V_1^N which is a matrix with orthonormal columns obtained from the Arnoldi method, and β_N which is an upper triangular matrix. Incorporating these changes yields the FOA based DMD shown in Algorithm 1 except the rank truncation part whose rationale will be discussed in section 3.2.2. Even though the Rayleigh Ritz procedure and FOA based DMD are theoretically equivalent, their computer implementations will yield different computed projection of A whose error will be analyzed in 3.2.2.

The batch processed version of FOA based DMD in Algorithm 1 can be equivalently recast in streaming form shown in Algorithm 3. The streaming update routine shown in Algorithm 4 is called in each pass of the streaming algorithm where a single, or a set of snapshot vectors, are processed to a compute better projection of the assumed linear mapping between the successive snapshots. Specifically, in each pass of the 'while loop' in Algorithm 3, the snapshot matrix X is formed by stacking p new consecutive snapshot vectors as columns. Then, the streaming FOA based DMD update routine shown in Algorithm 4 is called to update the matrix H, matrix V with Arnoldi-generated orthonormal basis, the upper triangular matrix β, and q which stores the current number of columns of V. Once these quantities are updated, the DMD modes and eigenvalues can be optionally evaluated with or without rank truncation. Usually, in streaming algorithms, p is set to 1 and only one new snapshot vector is used to update the matrices. However, in Algorithm 4, snapshots can also be streamed in batches of p. This would reduce the number of times the matrices are resized (step 2 of Algorithm 4) while processing a fixed total number of snapshots. For a given total number of snapshots, the computed Arnoldi vectors, the projection of the linear mapping, and the upper triangular matrix computed using the streaming form, are exactly the same as those computed using the batch processed form since the exact same floating point operations are carried out in the same order, in both algorithms. So, unless otherwise stated, we refer to the batch processed form of FOA based DMD in the following sections. Also, from Algorithm 3, note that FOA based DMD retains its streaming property even with rank truncation. This is because FOA based DMD with rank truncation additionally requires computing the SVD of β_{N-1} and the new projected matrix $U_r^H H_{q-1} U_r$ which does not require previous snapshots.

Algorithm 3: FOA based DMD in streaming form.

```
$q = 0$.
while true do
    Collect $p$ new snapshot vectors in $X$.
    $(H, V, \beta, q)$ = streamingFOAbasedDMDupdate $(X, p, H, V, \beta, q)$.
    if compute DMD modes and eigenvalues then
        Define $H_{q-1} := H_{1:q-1,1:q-1}$
        if rank truncation then
            Compute SVD of $\beta_{q-1} := \beta_{1:q-1,1:q-1}$, i.e., $\beta_{q-1} = U\Sigma W^H$.
            Choose the truncated rank $r$ and $U_r := U_{:,1:r}$.
            Compute right eigenvectors $\{z_i\}_{i=1}^r$ and eigenvalues $\{\lambda_i\}_{i=1}^r$ of $U_r^H H_{q-1} U_r$.
            DMD modes are $\{V_{:,1:q-1} U_r z_i\}_{i=1}^r$ and DMD eigenvalues are $\{\lambda_i\}_{i=1}^r$.
        else
            Compute right eigenvectors $\{z_i\}_{i=1}^{q-1}$ and eigenvalues $\{\lambda_i\}_{i=1}^{q-1}$ of $H_{q-1}$.
            DMD modes are $\{V_{:,1:q-1} z_i\}_{i=1}^{q-1}$ and DMD eigenvalues are $\{\lambda_i\}_{i=1}^{q-1}$.
        end if
    end if
    if stopping criterion == true then
        exit do while loop.
    end if
end while
```

Algorithm 4: Streaming FOA based DMD update routine.

```
$(H, V, \beta, q)$ = streamingFOAbasedDMDupdate $(X, p, H, V, \beta, q)$
Expand $H$ to $(p+q) \times (p+q-1)$ Upper Hessenberg matrix,
$V$ to $M \times (p+q)$ matrix,
$\beta$ to $(p+q) \times (p+q)$ upper triangular matrix and set all new entries to 0.
$jb := q$
if $q == 0$ then
    $\beta_{1,1} = \|X_{:,1}\|_2$
    $V_{:,1} := \frac{X_{:,1}}{\|X_{:,1}\|_2}$
    $jb := 1$
end if
for $j = jb$ to $p+q-1$ do
    $\beta_{1,j+1} = H_{1,1:j-1}\beta_{1:j-1,j}$
    for $i = 2$ to $j$ do
        $\beta_{i,j+1} = H_{i,i-1:j-1}\beta_{i-1:j-1,j}$
    end for
    $w = \frac{1}{\beta_{j,j}}\left(X_{:,j-q+1} - \sum_{i=1}^{j}\beta_{i,j+1}V_{:,i}\right)$
    $H_{1:j,j} = V_{:,1:j}{}^H w$
    $w = w - V_{:,1:j}H_{1:j,j}$
    $s_{1:j} = V_{:,1:j}{}^H w$
    $H_{1:j,j} = H_{1:j,j} + s_{1:j}$
    $w = w - V_{:,1:j}s_{1:j}$
    $H_{j+1,j} = \|w\|_2$; $V_{:,j+1} = \frac{w}{H_{j+1,j}}$
    for $i = 1$ to $j+1$ do
        $\beta_{i,j+1} = \beta_{i,j+1} + H_{i,j}\beta_{j,j}$
    end for
end for
$q = q + p$
```

3.2. Properties

3.2.1. Computational cost and memory consumption

This section presents the cost (number of floating point operations) and memory requirements of FOA based DMD, and compares it to existing projected DMD methods. We show in A.1 that the cost of computing the projection of linear mapping

Table 1
Cost (approximate) comparison of projected DMD methods without rank truncation.

Method	Cost without rank truncation
FOA based DMD	$5MN^2 + N^3/3$
SVD based DMD	$8MN^2 + 22N^3$
Streaming DMD	$10MN^2 + 16N^3/3$

Table 2
Cost (approximate) comparison of projected DMD methods with rank truncation.

Method	Cost with rank truncation
FOA based DMD	$5MN^2 + 37N^3/3 + 2rN^2 + 2r^2N$
SVD based DMD	$6MN^2 + 20N^3 + 2MrN + 2r^2N$

Table 3
Memory requirement (approximate) comparison of projected DMD methods without rank truncation.

Method	Memory required without rank truncation
FOA based DMD	$MN + N^2$
SVD based DMD	$2MN + 2N^2$
Streaming DMD	$2MN + 4N^2$

Table 4
Memory requirement (approximate) comparison of projected DMD methods with rank truncation.

Method	Memory required with rank truncation
FOA based DMD	$MN + 2N^2 + r^2$
SVD based DMD	$2MN + N^2 + r^2$

A onto the Krylov subspace $K_{N-1}(A, \psi_1)$ using snapshot matrix X_1^N without rank truncation is approximately $5MN^2 + N^3/3$, and with rank truncation is $5MN^2 + 37N^3/3 + 2rN^2 + 2r^2N$, where r is the truncated rank. The approximate number of floating point operations for SVD based DMD (with and without rank truncation) of Schmid [6] and streaming DMD of Hemati et al. [8] to obtain corresponding projected matrices is derived in A.2 and A.3 respectively, and compared in Table 1. Note that only terms which are cubic in the dimensions M, N of the problem are retained. From Table 1, it can be seen that FOA based DMD without rank truncation requires the least number of floating point operation count when compared to the existing projected DMD methods. We also compare the cost of rank truncated FOA based DMD and SVD based DMD in Table 2. Note that we do not consider the compressed streaming DMD of Hemati et al. [8] since this algorithm does not give the same result as rank truncated FOA based DMD and SVD based DMD methods (A.3). Table 2 shows that even with rank truncation, the cost of FOA based DMD is smaller than SVD based DMD.

We also derive the memory (number of floating point numbers) required for the three projected DMD methods considered (both with and without rank truncation) in Appendix A. Table 3 shows that without rank truncation, the FOA based DMD method utilizes approximately half the space as the other two projected DMD methods. Also, when rank truncation is used, from Table 4 we see that FOA based DMD still uses less memory than the SVD based DMD method.

3.2.2. *Finite precision error in computed projected linear mapping*

DMD computes projection of the assumed linear mapping A generally through orthonormal basis vectors (as this leads to well-conditioned eigenvalue problem) which are generated using the snapshot vectors $\{A^{i-1}\psi_1\}_{i=1}^N$. Since, we do not know explicitly the mapping A, as we will see in this section, the finite precision error in the computed projection depends on the condition number of the snapshot matrix. However, depending on the algorithm used to compute the projection, the error in projection might even depend on different powers of the condition number and might lead to larger error. This makes finite precision error analysis important for DMD algorithms. However, such an issue is not present if we explicitly know the linear mapping A and use the standard Arnoldi procedure in Algorithm 2, as this leads to computed projection of A which in a relative sense depends only on machine precision [23].

We first review the numerical stability results of QR factorization using CGS with reorthogonalization. Then, we perform finite precision error analysis of FOA based DMD and motivate rank truncation as a method to obtain small error due to finite precision arithmetic. Note that in addition to reducing the finite precision error, rank truncation also helps in rejecting statistical noise in the accumulated data. We also perform finite precision error analysis of SVD based DMD and streaming DMD and compare to FOA based DMD. Quantities with ˆ over them indicate computed counterparts in finite precision arithmetic and ϵ_m denotes machine precision. In deriving error estimates, we only consider real matrices and vectors. c_i's are defined as constants that are moderate powers of M, N, i.e. $c_i := c_i(M, N)$.

3.2.2.1. FOA based DMD The finite precision error analysis of QR factorization using CGS and MGS was performed by Björck and Paige [24] and discussed in Higham [21]. Giraud et al. [22] showed that CGS and also MGS with reorthogonalization not only retain the backward stability of QR factorization, but also yield vectors which are orthonormal upto machine precision. The finite precision results of QR factorization using CGS with reorthogonalization to orthonormalize the vectors obtained by Giraud et al. [22] is mentioned next as we will use these to discuss the finite precision properties of FOA based DMD.

Let $Y \in \mathbb{R}^{M\times N}$ be the matrix whose QR factorization is to be computed. Then QR factorization using CGS with reorthogonalization produces $\hat{Q}$ and $\hat{R}$ under the numerical nonsingularity of Y [22] such that

$$Y + \Delta Y = \hat{Q}\hat{R}, \quad \|\Delta Y\|_F \leq c_1\epsilon_m\|Y\|_F, \tag{16}$$

$$\|I - \hat{Q}^H\hat{Q}\|_2 \leq c_2\epsilon_m. \tag{17}$$

Equation (16) shows that computed factors $\hat{Q}$ and $\hat{R}$ are the exact QR factorization of a perturbed matrix bounded by the machine epsilon in a relative sense. This is called the backward error estimate. Equation (17) shows that the computed $\hat{Q}$ is orthonormal upto machine precision. Using the above estimates for QR decomposition Giraud et al. [22] also derived backward error estimates for the standard Arnoldi method (used in Algorithm 2) using the corresponding orthogonalization kernel. Even though the modified FOA is theoretically equivalent to the standard Arnoldi method, due to the unavailability of A in FOA based DMD, the error estimates for backward error of computed quantities of the standard Arnoldi do not apply. However, the computed Arnoldi vectors using modified FOA will still be orthonormal upto machine precision.

To obtain the backward error estimate of FOA based DMD without rank truncation, we first decompose $A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1}$ into different components which we can individually estimate as follows.

$$A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1} = A\left(\hat{V}_1^{N-1}\hat{\beta}_{N-1} - X_1^{N-1}\right) + \left(X_2^N - \hat{V}_1^N\hat{\beta}_{1:N,2:N}\right) + \hat{V}_1^N\left(\hat{\beta}_{1:N,2:N} - \hat{\hat{H}}_N\hat{\beta}_{N-1}\right) \tag{18}$$

We show in Appendix B (Equation (B.18)) that the error in computed factorization of X_1^{N-1} into the orthonormal vectors $\hat{V}_1^{N-1}$ and upper triangular matrix $\hat{\beta}_{N-1}$ is of order $\epsilon_m\|X_1^{N-1}\|_2$ which is important, as it implies that we can reliably use the columns of $\hat{V}_1^{N-1}$ as basis vectors for the range of X_1^{N-1}. Also, we show in Appendix B (Equation (B.18)) that each term in the right hand side of Equation (18) is of order $\epsilon_m\|X_1^{N-1}\|_2$ i.e.,

$$\|A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1}\|_F \leq C_1\left(\|A\|_2, \|\hat{\hat{H}}_N\|_2, M, N\right)\epsilon_m\|X_1^{N-1}\|_2 + O\left(\epsilon_m^2\right), \tag{19}$$

where $C_1\left(\|A\|_2, \|\hat{\hat{H}}_N\|_2, M, N\right)$ is a constant which is a function of $\|A\|_2$, $\|\hat{\hat{H}}_N\|_2$, M and N. The backward error of the Arnoldi relation using FOA based DMD can then be obtained as follows.

$$\begin{aligned}\|A\hat{V}_1^{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\|_F &= \|\left(A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1}\right)\hat{\beta}_{N-1}^{-1}\|_F,\\ &\leq \|A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1}\|_F\|\hat{\beta}_{N-1}^{-1}\|_2\,(\because \|AB\|_F \leq \|A\|_F\|B\|_2)\,.\end{aligned} \tag{20}$$

From Equation (B.17), we also see that $\|\hat{\beta}_{N-1}^{-1}\|_2 \leq \|X_1^{N-1^\dagger}\|_2 + O(\epsilon_m)$, where $X_1^{N-1^\dagger}$ is the pseudoinverse of X_1^{N-1}. Using this in Equation (20), we get the following backward error estimates,

$$\begin{aligned}\|A\hat{V}_1^{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\|_F &\leq C_1\epsilon_m\kappa_2\left(X_1^{N-1}\right) + O\left(\epsilon_m^2\right),\\ \|\hat{V}_1^{N-1^H}A\hat{V}_1^{N-1} - \hat{H}_{N-1}\|_F &\leq C_1\epsilon_m\kappa_2\left(X_1^{N-1}\right) + O\left(\epsilon_m^2\right),\end{aligned} \tag{21}$$

where $\kappa_2\left(X_1^{N-1}\right)$ is the 2-norm condition number defined as $\|X_1^{N-1}\|_2\|X_1^{N-1^\dagger}\|_2$.

The estimates obtained in Equation (21) are important because they imply that the error in the computed projection $\|\hat{V}_1^{N-1^H}A\hat{V}_1^{N-1} - \hat{H}_{N-1}\|_F$ is of order $\epsilon_m\kappa_2\left(X_1^{N-1}\right)$. So, as the condition number of the snapshot matrix increases, the error in the computed projection due to finite precision arithmetic increases. As we will see later, this $\kappa_2\left(X_1^{N-1}\right)$ dependence in the computed projection is also true for SVD based DMD and sometimes it might also depend on higher powers of $\kappa_2\left(X_1^{N-1}\right)$ which is the case in streaming DMD. This dependence primarily arises since we are working with the snapshot vectors and not with the linear mapping A itself, which is the case for the standard Arnoldi method (used in Algorithm 2). So, DMD methods without rank truncation should be used to compute DMD modes and eigenvalues when $\kappa_2\left(X_1^{N-1}\right) \ll 1/\epsilon_m$. Next, we discuss the rationale behind FOA based DMD with rank truncation, to reduce the finite precision error in the computed projected matrix which can be used when $\kappa_2\left(X_1^{N-1}\right) \approx 1/\epsilon_m$.

From equation (19), note that the error in $A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1}$ is small and does not depend on condition number of snapshot matrix. It is in Equation (20) when we multiply by inverse of $\hat{\beta}_{N-1}$ that we introduce dependence of $A\hat{V}_1^{N-1} - \hat{V}_1^N\hat{\hat{H}}_N$ on the condition number of snapshots. If the smallest singular value of $\hat{\beta}_{N-1}$ is very close to 0, then $\|\hat{\beta}_{N-1}^{-1}\|_2$ is very large thereby indicating large error in $A\hat{V}_1^{N-1} - \hat{V}_1^N\hat{\hat{H}}_N$ and $\hat{V}_1^{N-1^H}A\hat{V}_1^{N-1} - \hat{H}_{N-1}$. So, the computed projection of A onto the Krylov subspace will have large error. However, we can choose a subspace of the Krylov subspace on which we can more accurately compute the projection of A.

Consider the SVD (in exact arithmetic) of the upper triangular matrix $\hat{\beta}_{N-1}$ i.e. $\hat{\beta}_{N-1} = U\Sigma W^H$. Since, $X_1^{N-1} - \hat{V}_1^{N-1}\hat{\beta}_{N-1}$ is of size $\epsilon_m\|X\|_2$, the SVD of X_1^{N-1} is then

$$X_1^{N-1} = \hat{V}_1^{N-1}U\Sigma W^H + O(\epsilon_m\|X_1^{N-1}\|_2). \tag{22}$$

Define U_r as that first ‘r’ columns of matrix U, W_r as the first ‘r’ columns of matrix W and Σ_r as the $r \times r$ sub-matrix of Σ. Choosing the range of first r singular vectors of X_1^{N-1} as the subspace to perform Galerkin projection and $\hat{V}_1^{N-1}U_r$ as the corresponding basis (i.e. $V = Q = V_1^{N-1}U_r$ in Equation (4)), we can obtain the corresponding backward error as follows.

$$\begin{aligned}
\|A\hat{V}_1^{N-1}U_r - \hat{V}_1^N\hat{\hat{H}}_N U_r\|_F &= \|\left(A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1}\right)\hat{\beta}_{N-1}^{-1}U_r\|_F, \\
&\le \|A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1}\|_F\|\hat{\beta}_{N-1}^{-1}U_r\|_2 \; (\because \|AB\|_F \le \|A\|_F\|B\|_2), \\
&= \|A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1}\|_F\|\Sigma_r^{-1}\|_2, \\
\|\hat{V}_1^{N-1}U_r{}^H A\hat{V}_1^{N-1}U_r - U_r^H\hat{H}_{N-1}U_r\|_F &\le \|A\hat{V}_1^{N-1}U_r - \hat{V}_1^N\hat{\hat{H}}_N U_r\|_F(1 + O(\epsilon_m)) \\
&\le C_1\epsilon_m\|X_1^{N-1}\|_2\|\Sigma_r^{-1}\|_2 + O(\epsilon_m^2) \text{ (Using Equation (19))}
\end{aligned} \tag{23}$$

Here, we considered the SVD in exact arithmetic as the finite precision error in the computed SVD will only contribute to terms of order ϵ_m^2.

The span of the columns of $\hat{V}_1^{N-1}U_r$ represent a subspace of the Krylov subspace (equation 3). The computed projection of A onto the subspace formed by the columns of $\hat{V}_1^{N-1}U_r$ is then $U_r^H\hat{H}_{N-1}U_r$. Since, $\|\Sigma_r^{-1}\|_2$ is much less than $\|X_1^{N-1^\dagger}\|_2$ (for a suitably chosen r), Equation (23) tells us that the error in the computed projection of A onto subspace $\hat{V}_1^{N-1}U_r$ is smaller than that onto $\hat{V}_1^{N-1}$. This explains the rationale behind the rank truncation option of the proposed FOA based DMD algorithm shown in Algorithm 1. Also, the DMD eigenvalue λ of $U_r^H\hat{H}_{N-1}U_r$ and DMD eigenvector $V_1^{N-1}U_r z$, where z is the eigenvector corresponding to λ, are nearly the same as the ones that one would obtain using DMD with rank truncated SVD [6].

3.2.2.2. SVD based DMD Next, we perform finite precision error analysis of SVD based DMD [6] without rank truncation. The computed SVD of X_1^{N-1} has the property [25],

$$\begin{aligned}
&X_1^{N-1} + E = \left(\hat{U} + \delta\hat{U}\right)\hat{S}\left(\hat{W} + \delta\hat{W}\right)^H, \\
&\|E\|_2 \le p_1\epsilon_m\|X_1^{N-1}\|_2, \\
&\|\delta\hat{U}\|_2 \le p_2\epsilon_m, \quad \|\delta\hat{V}\|_2 \le p_3\epsilon_m,
\end{aligned} \tag{24}$$

where p_1, p_2, p_3 are polynomials of reasonable degree in the matrix dimensions of X_1^{N-1}. First, we estimate how close $A\hat{U}\hat{S}\hat{W}^H$ is to X_2^N.

$$\begin{aligned}
A\hat{U}\hat{S}\hat{W}^H &= A\left(X_1^{N-1} + E - \delta\hat{U}\hat{S}(\hat{W} + \delta\hat{W})^H - \hat{U}\hat{S}\delta\hat{W}^H\right) \\
&= X_2^N + A\left(E - \delta\hat{U}\hat{S}(\hat{W} + \delta\hat{W})^H - \hat{U}\hat{S}\delta\hat{W}^H\right).
\end{aligned} \tag{25}$$

Multiplying the above equation by $\hat{W}\hat{S}^{-1}$ from the right, we have

$$A\hat{U} = X_2^N\hat{W}\hat{S}^{-1} + A\left(E - \delta\hat{U}\hat{S}(\hat{W} + \delta\hat{W})^H - \hat{U}\hat{S}\delta\hat{W}^H\right)\hat{W}\hat{S}^{-1}. \tag{26}$$

Multiplying the above equation by U^H from left and accounting for the error in matrix–matrix multiplication,

$$\begin{aligned}
\hat{U}^H A\hat{U} - fl(\hat{U}^H X_2^N\hat{W}\hat{S}^{-1}) = &\left[\hat{U}^H X_2^N\hat{W}\hat{S}^{-1} - fl(\hat{U}^H X_2^N\hat{W}\hat{S}^{-1})\right] + \\
&\left[\hat{U}^H A\left(E - \delta\hat{U}\hat{S}(\hat{W} + \delta\hat{W})^H - \hat{U}\hat{S}\delta\hat{W}^H\right)\hat{W}\hat{S}^{-1}\right].
\end{aligned} \tag{27}$$

Observe that each of the two bracketed terms in right hand side is of size $O(\epsilon_m\kappa_2(X_1^{N-1}))$. So, the error in the computed projection of A is $O(\epsilon_m\kappa_2(X_1^N))$ even using the SVD approach.

To obtain the finite precision error of rank truncated SVD, multiplying the Equation (25) by $\hat{W}_r\hat{S_r}^{-1}$, where r is the chosen truncated rank, $\hat{W}_r$ is the first r columns of $\hat{W}$ and $\hat{S}_r$ is the top left $r \times r$ submatrix of $\hat{S}$, we obtain

$$A\hat{U}_r = X_2^N \hat{W}_r \hat{S}_r^{-1} + A\left(E - \delta\hat{U}\hat{S}(\hat{W}+\delta\hat{W})^H - \hat{U}\hat{S}\delta\hat{W}^H\right)\hat{W}_r\hat{S}_r^{-1}. \tag{28}$$

Similar to the analysis of SVD based DMD, multiplying the above equation by U_r^H on left and accounting for the error in matrix–matrix multiplication,

$$\begin{aligned}&\hat{U}_r^H A\hat{U}_r - fl(\hat{U}_r^H X_2^N \hat{W}_r\hat{S}_r^{-1}) \\ &\quad = \left[\hat{U}_r^H X_2^N \hat{W}_r\hat{S}_r^{-1} - fl(\hat{U}_r^H X_2^N \hat{W}_r\hat{S}_r^{-1})\right] + \left[\hat{U}_r^H A\left(E - \delta\hat{U}\hat{S}(\hat{W}+\delta\hat{W})^H - \hat{U}\hat{S}\delta\hat{W}^H\right)\hat{W}_r\hat{S}_r^{-1}\right].\end{aligned} \tag{29}$$

We can see from the above equation that size of the 2 bracketed terms in right hand side is $\epsilon_m \|X_1^{N-1}\|_2 \|\hat{S}_r^{-1}\|_2$ which is small when compared to the $O(\epsilon_m \kappa_2\left(X_1^{N-1}\right))$ in Equation (27) for a suitably chosen r. Hence, the finite precision error in the computed projected matrix for SVD based DMD with rank truncation is smaller than that without rank truncation.

3.2.2.3. Streaming DMD Streaming DMD [8] uses QR decomposition to compute the basis vectors for the Krylov subspace (Equation (3)). We do not consider compression and derive the finite precision error for the computed projection of the linear mapping.

$$\begin{aligned} AX_1^{N-1} &= X_2^N, \\ A\left(\hat{Q}_X\hat{R}_X + E_X\right) &= \left(\hat{Q}_Y\hat{R}_Y + E_Y\right), \end{aligned} \tag{30}$$

where $\hat{Q}_X, \hat{R}_X$ and $\hat{Q}_Y, \hat{R}_Y$ is the computed QR decomposition of X_1^{N-1} and X_2^N respectively. From Equation (16), it follows that the backward errors E_X and E_Y, in the computed factors are bounded by $c_1\epsilon_m\|X_1^{N-1}\|_2$ and $c_1\epsilon_m\|X_2^N\|_2$ respectively. In streaming DMD, the projected linear mapping is computed as $\hat{Q}_X^H\hat{Q}_Y\left(\hat{R}_X\hat{R}_Y^H\right) fl\left(\hat{R}_X\hat{R}_X^H\right)^{-1}$ using an incremental method to compute the matrix products $\hat{R}_X\hat{R}_Y^H$ and $\hat{R}_X\hat{R}_X^H$. The finite precision arithmetic error $\|\hat{Q}_X^H A\hat{Q}_X - \hat{Q}_X^H\hat{Q}_Y\left(\hat{R}_Y\hat{R}_X^H\right) fl\left(\hat{R}_X\hat{R}_X^H\right)^{-1}\|_F$ can be computed as follows. Multiplying Equation (30) by $\hat{R}_X^H$ from the right and rearranging, we get

$$A\hat{Q}_X\hat{R}_X\hat{R}_X^H = \hat{Q}_Y\hat{R}_Y\hat{R}_X^H - AE_X\hat{R}_X^H + E_Y\hat{R}_X^H. \tag{31}$$

Multiplying by $\left(\hat{R}_X\hat{R}_X^H\right)^{-1}$ on the right and $\hat{Q}_X^H$ from the left we have,

$$\begin{aligned} \hat{Q}_X^H A\hat{Q}_X - \hat{Q}_X^H\hat{Q}_Y\hat{R}_Y\hat{R}_X^H\left(\hat{R}_X\hat{R}_X^H\right)^{-1} &= \hat{Q}_X^H\left(E_Y\hat{R}_X^H - AE_X\hat{R}_X^H\right)\left(\hat{R}_X\hat{R}_X^H\right)^{-1}, \\ \hat{Q}_X^H A\hat{Q}_X - \hat{Q}_X^H\hat{Q}_Y\hat{R}_Y\hat{R}_X^H fl\left(\hat{R}_X\hat{R}_X^H\right)^{-1} &= \hat{Q}_X^H\left(E_Y\hat{R}_X^H - AE_X\hat{R}_X^H\right)\left(\hat{R}_X\hat{R}_X^H\right)^{-1} \\ &\quad - \hat{Q}_X^H\hat{Q}_Y\hat{R}_Y\hat{R}_X^H\left(fl\left(\hat{R}_X\hat{R}_X^H\right)^{-1} - \left(\hat{R}_X\hat{R}_X^H\right)^{-1}\right). \end{aligned} \tag{32}$$

Observe that the size of right hand side in the above equation can be as large as order $\epsilon_m\left(\kappa_2(X_1^{N-1})\right)^2$. This is in contrast to FOA based and SVD based DMD without rank truncation whose finite precision arithmetic error in the computed projection is only order $\epsilon_m\kappa_2(X_1^{N-1})$. So, for a fixed condition number of snapshots, the finite precision error in the computed projection from streaming DMD is higher than FOA based and SVD based DMD. This is primarily because streaming DMD involves inversion of the matrix product $\hat{R}_X\hat{R}_X^H$.

Fig. 1 shows numerical evidence for the above obtained error estimates for different projected DMD methods without rank truncation. The linear mapping A is chosen to be a Vandermonde matrix $\in \mathbb{R}^{50\times 50}$ generated in Matlab using $vander(linspace(0, 1, 50))$. Snapshots are generated by the application of A onto a random starting vector. The number of snapshots is varied to obtain a range of condition numbers of the snapshot matrix. The finite precision error in the computed projection is plotted as a function of the condition number of the snapshot matrix for each of the three projected DMD methods in Fig. 1. As we can see from the figure, error in the computed projection using streaming DMD can depend on $\left(\kappa_2(X_1^{N-1})\right)^2$ whereas that from FOA based and SVD based DMD depends on $\kappa_2(X_1^{N-1})$.

3.2.3. Error indicators for DMD modes and eigenvalues

DMD computes approximate eigenvector and eigenvalues of the assumed linear mapping A. In exact arithmetic, using matrix $\bar{H}_N$ and the eigenvector z_i it is possible to obtain error estimates associated with DMD eigenvalues and DMD

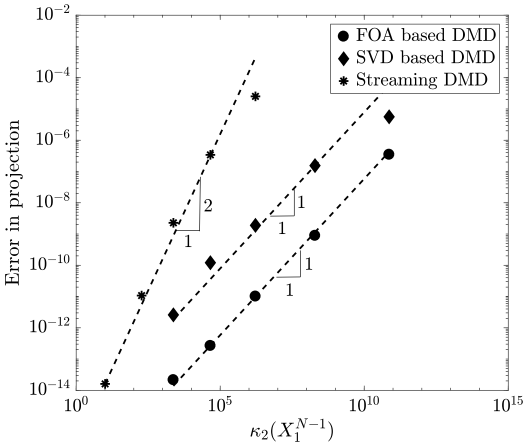


Fig. 1. Finite precision error in computed projection using different projected DMD methods.

eigenvectors [20] in ℓ^2 norm computed using FOA based DMD by using the relation

$$\|AV_1^{N-1}z_i - \lambda_i V_1^{N-1}z_i\|_2 = \|h_{N,N-1}v_N e_{N-1}^H z_i\|_2. \tag{33}$$

By monitoring the value of $\|h_{N,N-1}v_N e_{N-1}^H z_i\|_2$ it is possible to monitor the accuracy of DMD eigenvector and eigenvalue pairs. This can be reasoned by the Bauer–Fike theorem [20] which states that suppose (λ_a, v_a) is an approximate eigenvalue–eigenvector pair, then the error in λ_a is related to the residual $\|Av_a - \lambda_a v_a\|_2$ such that

$$|\lambda_a - \lambda| \leq \kappa_2(V)\frac{\|Av_a - \lambda_a v_a\|_2}{\|v_a\|_2} \tag{34}$$

where V is the matrix of right eigenvectors of A (assuming that its eigenvectors are linearly independent). So, if the DMD eigenvector has unit magnitude, the error in DMD eigenvalue λ_i computed using FOA based DMD is indicated by $\|h_{N,N-1}v_N e_{N-1}^H z_i\|_2$. This can be used to devise stopping criterions for acquiring new snapshots when using FOA based DMD.

Suppose, we use FOA based DMD with rank truncation, the error indicator for the DMD eigenvectors and DMD eigenvalues can be obtained as follows. In exact arithmetic,

$$\begin{aligned}
AV_1^{N-1}U_r z_i - \lambda_i V_1^{N-1}U_r z_i &= V_1^{N-1}(I - U_r U_r^H)H_{N-1}U_r z_i + h_{N,N-1}v_N e_{N-1}^H U_r z_i,\\
\|AV_1^{N-1}U_r z_i - \lambda_i V_1^{N-1}U_r z_i\|_2 &= \|V_1^{N-1}(I - U_r U_r^H)H_{N-1}U_r z_i + h_{N,N-1}v_N e_{N-1}^H U_r z_i\|_2,\\
\|AV_1^{N-1}U_r z_i - \lambda_i V_1^{N-1}U_r z_i\|_2 &= \left\|\begin{bmatrix}(I - U_r U_r^H)H_{N-1}\\ h_{N,N-1}e_{N-1}^H\end{bmatrix}U_r z_i\right\|_2.
\end{aligned} \tag{35}$$

The error in DMD eigenvector $V_1^{N-1}U_r z_i$ (unit magnitude in ℓ^2 norm) and eigenvalue λ_i pair computed using FOA based DMD with rank truncation is indicated by $\left\|\begin{bmatrix}(I - U_r U_r^H)H_{N-1}\\ h_{N,N-1}e_{N-1}^H\end{bmatrix}U_r z_i\right\|_2$.

Next, we investigate how close the computed error indicator for the DMD eigenvector and eigenvalue obtained using FOA based DMD without rank truncation is to the actual error using the finite precision results derived in previous section. From Equation (19) and assuming $\hat{\beta}_{N-1}$ to be full rank, we have

$$\|A\hat{V}_1^{N-1}\hat{\beta}_{N-1}y_i - \hat{V}_1^N \hat{\hat{H}}_N \hat{\beta}_{N-1}y_i\|_2 \leq C_1\epsilon_m \|X_1^{N-1}\|_2 \|y_i\|_2, \tag{36}$$

$$\hat{\beta}_{N-1}y_i = z_i, \tag{37}$$

where λ_i, z_i is an exact eigenvalue eigenvector pair of $\hat{H}_{N-1}$. Here, we assume for simplicity that the y_i is the exact solution of $\hat{\beta}_{N-1}y_i = z_i$ and z_i is the exact eigenvector of $\hat{H}_{N-1}$. Then, the accuracy of the error indicator is

$$\|A\hat{V}_1^{N-1}z_i - \lambda_i\hat{V}_1^{N-1}z_i - \hat{h}_{N,N-1}\hat{v}_N e_{N-1}^H z_i\|_2 \leq C_1\epsilon_m \|X_1^{N-1}\|_2 \|y_i\|_2. \tag{38}$$

The above equation tells us that the computed error indicator $\hat{h}_{N,N-1}\hat{v}_N e_{N-1}^H z_i$ is a good estimate of the actual error $A\hat{V}_1^{N-1}z_i - \lambda_i\hat{V}_1^{N-1}z_i$ if the magnitude of y_i obtained from solving Equation (37) is small. This is essentially the same as saying that the component of z_i along the left singular vectors corresponding to near zero singular values of $\hat{\beta}_{N-1}$ is small.

However, for FOA based DMD with rank truncation, the computed error indicator $\|\begin{bmatrix}(I-U_rU_r^H)H_{N-1}\\ h_{N,N-1}e_{N-1}^H\end{bmatrix}U_rz\|_2$ for the computed DMD eigenvector and eigenvalue will be very close to the actual error for a suitably chosen truncated rank r. This is because the backward error in Equation (23) is small for appropriately chosen r and is controlled by $\|\Sigma_r^{-1}\|_2$ rather than $\|\hat{\beta}_{N-1}^{-1}\|_2$. Numerical experiments shown in section 4.1 support the above arguments regarding finite precision quality of the FOA based DMD error indicators.

3.2.4. Reconstruction error and interpolation arguments

Let $\varphi_j := V_1^{N-1}z_j;\ j=1,\dots,N-1$. Since, we have assumed that the DMD eigenvectors are linearly independent, $span\{\varphi_i\}_{i=1}^{N-1}$ is same as $K_{N-1}(A,\psi_1)$. So, we should be able to reconstruct at least the first $N-1$ snapshots using $\{\varphi\}_{j=1}^{N-1}$ exactly [3].

Rowley et al. [3] showed that all snapshot vectors except the last one in the sequence can be exactly reconstructed from the DMD modes. The reconstruction error of the last snapshot was shown to be that of its orthogonal projection onto the previous ones. Here, we obtain the same results using the Arnoldi basis vectors in the context of FOA based DMD. Suppose $\psi=\sum_{j=1}^{N-1}c_j\varphi_j$. Then, say we approximate $A\psi$ as $\sum_{j=1}^{N-1}\lambda_jc_j\varphi_j$. The error involved in this approximation can be obtained as follows

$$\begin{aligned}A\psi-\sum_{j=1}^{N-1}\lambda_jc_j\varphi_j&=\sum_{j=1}^{N-1}c_j\left(A\varphi_j-\lambda_j\varphi_j\right)\\ &=\sum_{j=1}^{N-1}c_j\left(h_{N,N-1}v_Ne_{N-1}^Hz_j\right)=h_{N,N-1}\left(\sum_{j=1}^{N-1}c_je_{N-1}^Hz_j\right)v_N,\end{aligned}\tag{39}$$

$$\|A\psi-\sum_{j=1}^{N-1}\lambda_jc_j\varphi_j\|_2=|h_{N,N-1}|\,|\sum_{j=1}^{N-1}c_je_{N-1}^Hz_j|.\tag{40}$$

$\psi=\sum_{j=1}^{N-1}c_j\varphi_j$ can be written in matrix form as $\psi=[\varphi]\{c\}$ where $[\varphi]$ is matrix formed by stacking φ_j as columns and $\{c\}$ is the vector of coefficients, i.e. $[\varphi]=[\varphi_1\quad\dots\quad\varphi_{N-1}]$ and $\{c\}=[c_1\quad\dots\quad c_{N-1}]^H$. Let $[z]:=[z_1\quad\dots\quad z_{N-1}]$. Then, we have

$$\begin{aligned}&\psi=V_1^{N-1}[z]\{c\},\\ &\psi=V_1^{N-1}\{\tilde{c}\}\quad(\{\tilde{c}\}:=[z]\{c\}).\end{aligned}\tag{41}$$

Suppose $h_{N,N-1}$ is not 0. Then, $\|A\psi-\sum_{j=1}^{N-1}\lambda_jc_j\varphi_j\|_2$ is 0 if $\sum_{j=1}^{N-1}c_je_{N-1}^Hz_j$ is 0, i.e. the last entry in the vector $\{\tilde{c}\}$ should be 0 or equivalently, $\psi=V_1^{N-2}\{\bar{c}\}$ where $\{\bar{c}\}=[\tilde{c}_1\quad\dots\quad\tilde{c}_{N-2}]^T$. Therefore, the approximation $\sum_{j=1}^{N-1}\lambda_jc_j\varphi_j$ to $A\psi$ is exact if at least one of the below conditions are satisfied.

1. ψ is in the range of V_{N-2}.
2. $h_{N,N-1}=0$, i.e. if the last snapshot can be written as linear combination of the previous snapshots.

The above 2 conditions are essentially consequences of the fact that we use Galerkin projection onto the subspace $K_{N-1}(A,\psi_1)$.

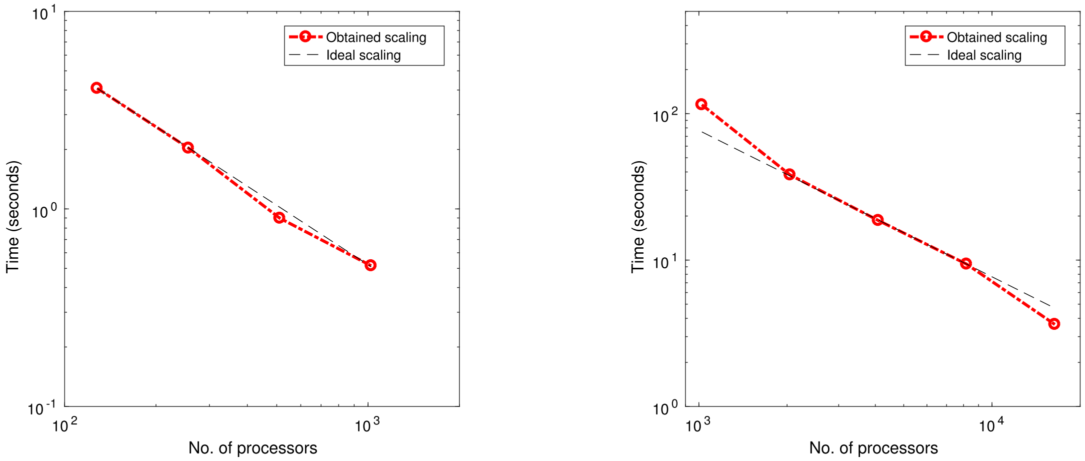
The FOA based DMD method stops if linearly dependent snapshots are present, i.e. say ψ_{j+1} can be expressed as a linear combination of $\{\psi_1,\dots,\psi_j\}$. Then, in the jth step of Arnoldi method, $h_{j+1,j}$ is 0. The Galerkin method with the previously computed v_i's will give exact eigenvectors and eigenvalues associated with the linear mapping A that connects the snapshot vectors.

In a continuous sense, DMD generates an interpolant through the snapshot vectors. The interpolant can be defined as

$$\begin{aligned}I^N\psi(t)&:=\sum_{j=1}^{N-1}e^{\omega_jt}d_j\varphi_j,\\ I^N\psi_k&:=I^N\psi(t_k),\end{aligned}\tag{42}$$

where $\omega_j:=ln(\lambda_j)/\Delta t$, Δt is the time spacing between the snapshots, $t_k:=(k-1)\Delta t$ and $\psi_1=\sum_{j=1}^{N-1}d_j\varphi_j$. Equation (40) implies that snapshots ψ_1 to ψ_{N-1} can always be exactly reconstructed from the DMD eigenmodes and DMD eigenvalues, i.e.

$$\psi_k=I^N\psi_k\quad k=1,\dots,N-1\tag{43}$$



(a) Strong scaling on TACC Stampede2 cluster (8 million, 101 snapshots).

(b) Strong scaling on TACC Stampede2 cluster (240 million, 201 snapshots).

Fig. 2. Strong scaling of H_{N-1} computation using FOA based DMD (Algorithm. 1).

and the error associated with reconstruction of $I^N\psi_{N-1}$ is

$$\|I^N\psi_N - \psi_N\|_2 = |h_{N,N-1}|\,|\sum_{j=1}^{N-1} d_j\lambda_j^{N-2}e_{N-1}^H z_j|. \tag{44}$$

In the linear case, the interpolant $I^N\psi(t)$ approximates the evolution $e^{Bt}\psi_1$, where $A = e^{B\Delta t}$. For the nonlinear case, suppose we use P snapshots in the FOA based DMD algorithm starting from snapshot ψ_1 and $h_{P,P-1}$ turns to be 0, then DMD eigenvectors and eigenvalues are the exact eigenpairs of A. All P snapshots can be constructed exactly using DMD eigenvectors and DMD eigenvalues and consequentially the interpolant $I^P\psi(t)$. Now, if we use $I^P\psi(t)$ to predict the observable vectors at future times, the error in the prediction is an indication of how close the span of observables is to a Koopman function and eigenvalues [7] of the system. Numerical experiments on evaluating the quality of approximate Koopman eigenfunctions obtained from DMD for different choices of set of observables is carried out in Zhang et al. [19].

The coefficients $\{c\}$ of first snapshot ψ_1 in the basis formed by DMD eigenmodes can be obtained by solving the following matrix problem of size $(N-1)\times(N-1)$.

$$\begin{aligned}[\varphi]\{c\} &= \psi_1,\\ [z]\{c\} &= \|\psi_1\|_2 e_1.\end{aligned} \tag{45}$$

If rank truncated FOADMD is used, then the coefficients $\{c\}$ of the first snapshot in eigenvector basis such that $\|\psi_i - [V_1^{N-1}U_r][z]\{c\}\|_2$ is minimized, can be obtained by solving

$$[z]\{c\} = \|\psi_1\|_2 U_r^H e_1. \tag{46}$$

3.2.5. Parallel scaling

The computational kernels of FOA based DMD algorithm include vector additions, dot products and matrix vector multiplications which can be parallelized. The snapshot data can be partitioned row-wise among different processors. Figs. 2a and 2b show strong scaling for the computation of projected matrix H_{N-1} using FOA based DMD algorithm on Texas Advanced Computing Center (TACC) Stampede2 Knights Landing cluster upto 16384 processors respectively. Both plots show time taken to generate Arnoldi vectors and Hessenberg matrix for different number of processors. The strong scaling shown in Fig. 2a utilized 101 snapshots of size 8 million each and in Fig. 2b used 201 snapshots of size 240 million each. The snapshot matrix was filled with random numbers as the operation count and scaling properties of the algorithm is independent of the content of the snapshot matrix. Figs. 2a and 2b show good scaling properties of the algorithm. Since we use CGS with reorthogonalization, $O(N)$ number of messages are exchanged between processors. This is in contrast to MGS orthogonalization kernel which would require $O(N^2)$ number of messages.

4. Numerical experiments

We illustrate application of FOA based DMD (Algorithm 1) on three problems, i) snapshots from a linearized channel flow simulation, ii) snapshots of vorticity field from flow over a circular cylinder and iii) snapshots from turbulent flow simulation of jets in cross flow.

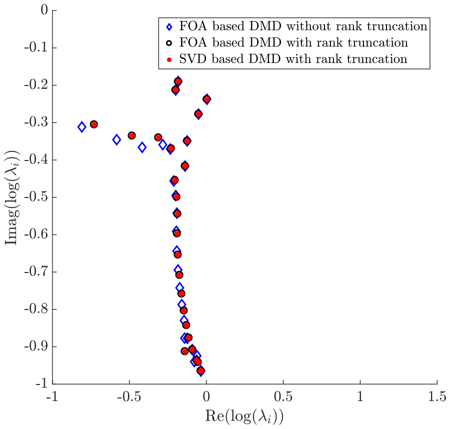


Fig. 3. Comparison of DMD eigenvalues for linearized channel flow simulation snapshots.

Table 5
Error $\|P_A - \hat{P}_A\|_2$ in computed projection of A for FOA based DMD, SVD based DMD and streaming DMD for linearized channel flow case. P_A is the exact projection and $\hat{P}_A$ is the computed projection.

$\|P_A - \hat{P}_A\|_2$	FOA based DMD	SVD based DMD	Streaming DMD
Without rank truncation	2.22e+01	6.03+00	1.05e+09
With rank truncation	5.77e−04	2.09e−03	–

4.1. Linearized channel flow simulation at $Re = 10000$

The dataset used to perform DMD analysis of linearized channel flow simulation was obtained from Jovanović et al. [16]. For a description of the numerical method and mesh resolution refer Jovanović et al. [16]. This dataset is extremely ill-conditioned. The condition number of snapshot matrix is $\approx 10^{17}$. Even though, 100 snapshots are present, the numerical rank of the dataset is 26. We define numerical rank as the number of singular values of snapshot matrix that are larger than $N * eps(\|A\|_2)$, where $eps(x)$ is the distance between $|x|$ and the next larger double precision floating point number. Also, the linear mapping A is available. So, we can compute the error involved in projection of A, DMD eigenvectors and eigenvalues and assess quality of proposed error indicators when the snapshots become extremely ill-conditioned. From the same set of snapshots as used in Jovanović et al. [16] we compute the DMD eigenvectors and eigenvalues and compare to the SVD based DMD. Streaming DMD without compression would not yield reliable results for this case due to the very high condition number of snapshot matrix. The size of each snapshot is 150 and a total of 100 snapshots are used. DMD eigenvalues obtained using FOA based DMD with and without rank truncation is compared to SVD based DMD in Fig. 3. The DMD eigenvectors and eigenvalues are ordered based on the values of coefficients obtained by projection of the first snapshot onto the DMD eigenmodes using equation (45).

Table 5 shows the error in computed projection of A. The exact projection P_A is calculated as $fl(Q^H AQ)$. Since, matrix–matrix multiplication with orthogonal matrix is backward-stable [21], error in evaluation of P_A will be very small. Finite precision error analysis of SVD based DMD and FOA based DMD without rank truncation showed that the error in computed projection is $O(\kappa_2(X_1^N)\epsilon_m)$. This is consistent with the numerical results shown in Table 5. Also, the large error associated with streaming DMD is because of its dependence on higher powers of $\kappa_2(X_1^{N-1})$ as explained in section 3.2.2. The increase in accuracy of computed projection with rank truncated versions of SVD based DMD and FOA based DMD seen in Table 5 is explained in section 3.2.2.

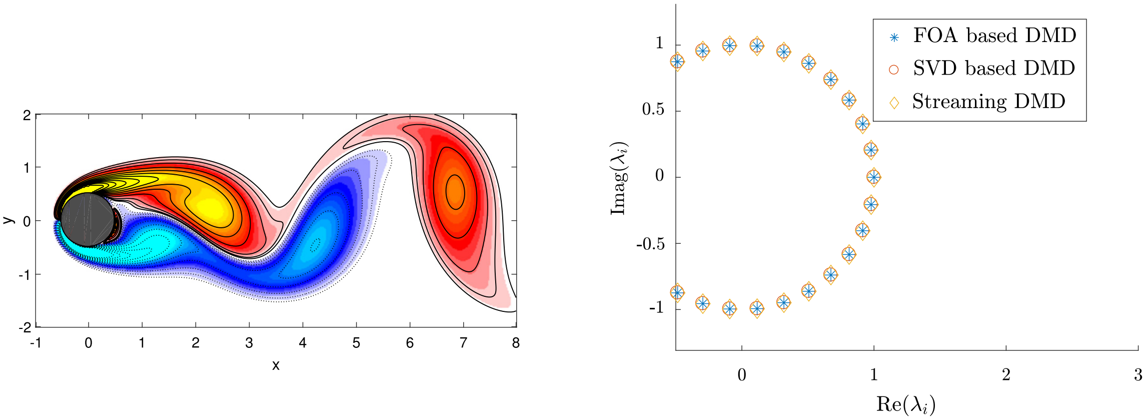
Next, we consider the accuracy of computed DMD eigenvectors and eigenvalues using FOA based DMD with and without rank truncation. Tables 6 and 7 show the comparison of actual error (Error$_a$) in eigenvalue–eigenvector pair to the prediction using the error indicator (Error$_p$) with and without rank truncation respectively for the 8 DMD modes with least error. The actual error and predicted error is defined as $\|AV_1^{N-1}z - \lambda V_1^{N-1}z\|_2$ and $|h_{N,N-1}e_{N-1}^H z|$ respectively for the case without rank truncation and for the case with rank truncation, they are defined as $\|AV_1^{N-1}U_r z - \lambda V_1^{N-1}U_r z\|_2$ and $\|V_1^{N-1}\left(I - U_r U_r^H\right) H_{N-1}U_r z + h_{N,N-1}v_N e_{N-1}^H U_r z\|_2$ respectively, where λ and z are the eigenvalue and eigenvector pair of the projected problem. As, we can see from Tables 6 and 7, the error in dominant eigenvalues and corresponding eigenvectors are of the same order of magnitude. However, the predicted error is more close to the actual error for the case with rank truncation than without rank truncation. This is because of the smaller backward error for the Arnoldi relation (equation (23)) for the case with rank truncation.

Table 6
Comparison of actual error $\|AV_1^{N-1}z - \lambda V_1^{N-1}z\|_2$ with the predicted error without rank truncation.

Eigenvalue	Error$_a$	Error$_p$
9.76e−01+(−2.36e−01)i	3.75e−14	5.37e−15
9.14e−01+(−2.60e−01)i	1.07e−10	1.62e−11
8.30e−01+(−3.02e−01)i	5.79e−08	9.44e−09
8.18e−01+(−1.57e−01)i	1.90e−06	2.69e−07
5.62e−01+(−7.19e−01)i	1.39e−05	2.53e−06
5.56e−01+(−7.56e−01)i	8.03e−06	1.44e−06
5.50e−01+(−7.93e−01)i	5.32e−06	9.38e−07
7.97e−01+(−3.52e−01)i	3.77e−06	6.44e−07

Table 7
Comparison of actual error $\|AV_1^{N-1}U_r z - \lambda V_1^{N-1}U_r z\|_2$ with the predicted error with rank truncation.

Eigenvalue	Error$_a$	Error$_p$
9.76e−01+(−2.36e−01)i	3.40e−13	3.40e−13
9.14e−01+(−2.60e−01)i	6.01e−10	6.00e−10
8.30e−01+(−3.02e−01)i	1.53e−07	1.53e−07
8.18e−01+(−1.57e−01)i	6.32e−06	6.32e−06
5.62e−01+(−7.20e−01)i	1.30e−05	1.30e−05
5.57e−01+(−7.56e−01)i	5.66e−06	5.66e−06
5.50e−01+(−7.93e−01)i	3.02e−04	3.02e−04
7.97e−01+(−3.52e−01)i	7.58e−06	7.57e−06



(a) Snapshot of flow over circular cylinder at Re=100.

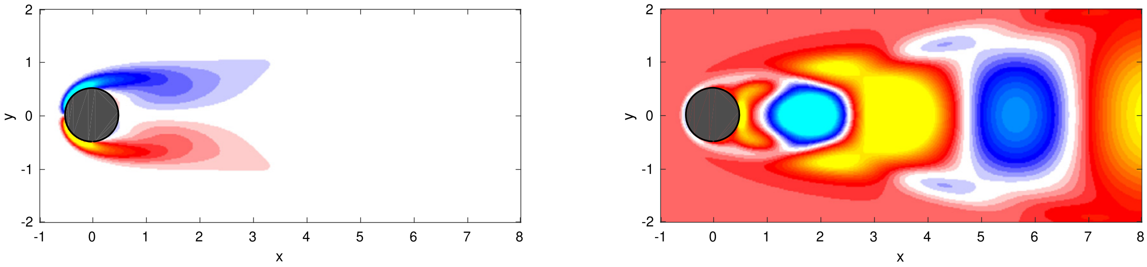
(b) Comparison of DMD eigenvalues obtained from FOA based DMD, SVD based DMD [6] and streaming DMD [8].

Fig. 4. Cylinder test case.

4.2. Flow over cylinder at $Re = 100$

We consider the flow over a circular cylinder at Reynolds number ($Re := VD/\nu$) of 100 (V, D and ν denote freestream velocity, diameter of cylinder and kinematic viscosity of fluid respectively). The data set for this case is obtained from Kutz et al. [26]. For details on the numerical method used to generate the cylinder snapshots refer Kutz et al. [26]. A total of 151 snapshots of vorticity field separated by time interval $\Delta t = 0.2$ were used. Fig. 4a shows a snapshot of the vorticity field. The cylinder is placed at $(0, 0)$ and the diameter of the cylinder is 1. The flow is from left to right. It is known that as we increase Re, for $Re > 47$ vortices are shed from the cylinder at a particular frequency dependent on Re. These vortices can be clearly seen from Fig. 4a. At $Re = 100$, the Strouhal number $St = fD/V$ is 0.16. We can capture this frequency by performing DMD of the snapshots of vorticity field. The condition number of the snapshots for flow over cylinder at $Re = 100$ is $\approx 10^7$.

Fig. 4b compares DMD eigenvalues associated with 21 dominant eigenmodes from the proposed FOA based DMD method (without truncation), SVD based DMD (with rank truncation) and streaming DMD. The DMD eigenmodes from FOA based DMD are ranked based on the magnitude of c_j when the first snapshot ψ_1 is represented in the basis of DMD eigenmodes (equation 45). Also, only few of the computed DMD eigenmodes have sufficiently large values of c_j (not shown). These DMD eigenmodes give an appropriate basis to represent the dataset. In SVD based DMD, only the first 21 singular vectors are



(a) DMD eigenmode for $\frac{log(\lambda)}{2\pi\Delta t} = 0$.

(b) Real part of DMD eigenmode for $\frac{log(\lambda)}{2\pi\Delta t} = 0.1654i$.

Fig. 5. DMD modes from snapshots of vorticity of flow over cylinder using FOA based DMD without rank truncation.

Table 8
Predicted error in DMD eigenvector and eigenvalues for the 5 most dominant eigenvalues.

$\frac{log(\lambda)}{2\pi\,\Delta t}$	Error_{DMD}
−0.0000+(0.0000)i	3.24e−08
0.0000+(0.1654)i	7.60e−08
0.0000+(−0.1654)i	7.60e−08
−0.0000+(0.3308)i	2.18e−07
−0.0000+(−0.3308)i	2.18e−07

used to evaluate the DMD eigenvalues and eigenmodes. Since, the condition number of the snapshot matrix is $\approx 10^7 \ll \epsilon_m$ streaming DMD returns reliable eigenvalues. Similar to FOA based DMD, the DMD modes obtained from streaming DMD are sorted based on the magnitude of the coefficients when the first snapshot ψ_1 is represented as their linear combination. Fig. 4b shows good agreement of the computed DMD eigenvalues between the different projected DMD methodologies considered.

Figs. 5a and 5b show the first 2 dominant DMD eigenmodes obtained from FOA based DMD. DMD eigenmode in Fig. 5a has DMD eigenvalue of 1 and represents the mean vorticity field. The second dominant DMD eigenmode shown in Fig. 5b corresponds to the vortex shedding frequency. The frequency of oscillation of DMD mode is given by $imag(log(\lambda)/(2\pi\,\Delta t))$ where λ is corresponding DMD eigenvalue. Fig. 5b corresponds to DMD eigenvalue, $\lambda = 0.9875 - 0.2063i$ whose associated frequency is 0.1654, which is exactly the St at $Re = 100$.

The snapshots matrix is full rank (measured using numerical rank). So, rank truncation is not required for this case. Table 8 shows the quality of the five most dominant DMD eigenvectors and eigenvalues. Since, the condition number of the snapshots is very small when compared with $\frac{1}{\epsilon_m}$, the computed error indicator $|h_{N,N-1}e^H_{N-1}z|$ should be close to the actual error $\|AV_1^{N-1}z - \lambda V_1^{N-1}z\|_2$.

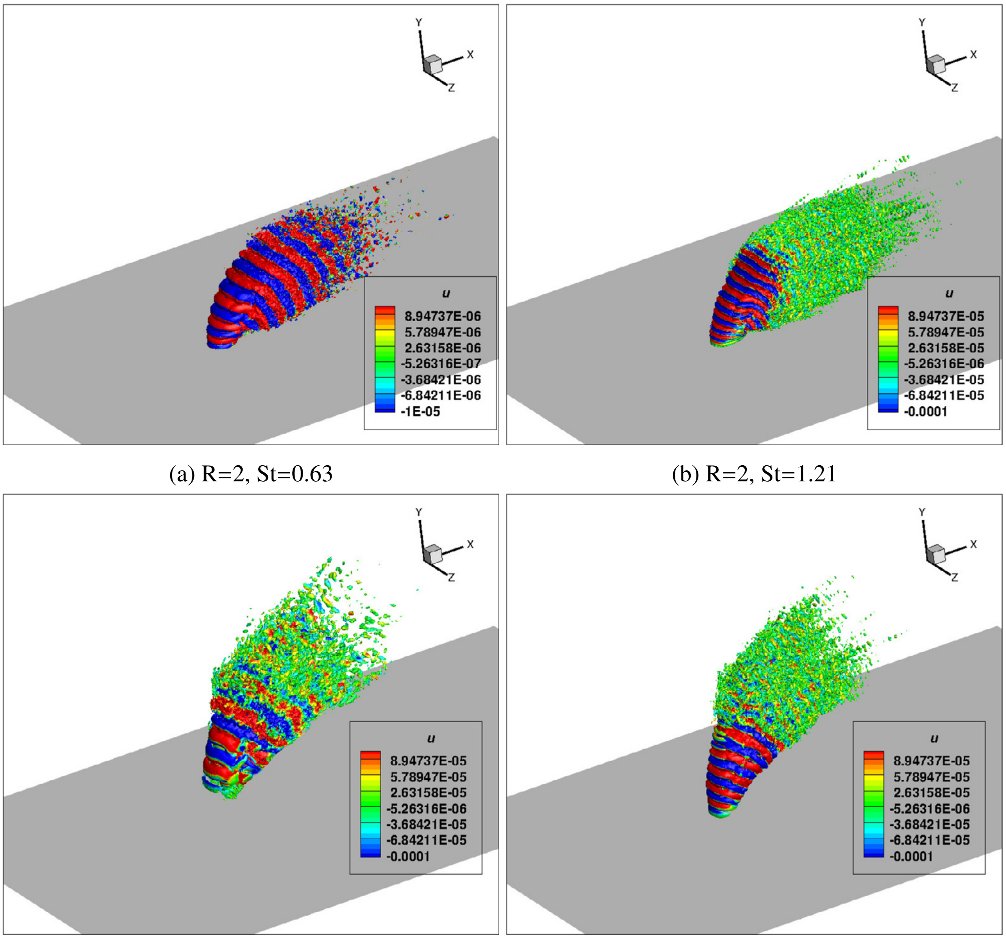
4.3. Jets in cross flow

Next, we consider a large dataset obtained from Direct Numerical Simulation (DNS) of turbulent jets in cross flow. DMD on this dataset has been previously performed by Iyer and Mahesh [27] using the algorithm of Schmid [6] using snapshot matrix as the basis vectors. Here, we use FOA based DMD to obtain the DMD modes and eigenvalues and compare to previously obtained results. For more information on the problem, simulation, timestep and grid, the reader is referred to Iyer and Mahesh [27]. Here, we consider jet velocity to cross-flow velocity ratios (R) of 2 and 4. The size of each snapshot is 240 million. A total of 80 snapshots were used for R = 4, and 250 snapshots were used for R = 2 case.

The parallel implementation discussed in section 3.2.5 was used. The data was split row-wise among different processors. A total of 512 processors were used to process the data. The condition number of snapshot matrix is $\approx 10^2$ which is very small when compared with $1/\epsilon_m$ for double precision datatype. So, FOA based DMD without rank truncation is used to compute the DMD eigenvalues and eigenvectors. Also, in Iyer and Mahesh [27], DMD eigenvalues and eigenvectors were obtained through the projected companion matrix. The normal equations for the last column of the companion matrix were solved using rank truncated SVD. Table 9 shows the good agreement of the Strouhal number (non-dimensionalized with peak jet velocity and diameter of orifice) associated with shear layer modes obtained from FOA based DMD with Iyer and Mahesh [27]. Fig. 6 shows the Q-criterion of shear layer modes at R = 2 and R = 4 and they agree with those obtained by Iyer and Mahesh [27]. Also, the computed error indicator $|h_{N,N-1}e^H_{N-1}z|$ presented in Table 9 shows the better quality of eigenvectors for R = 2 case than R = 4. This is because of a larger set of snapshots used for R = 2 than R = 4. These error indicators are very useful as it allows us to quantitatively compare the quality of DMD modes and eigenvalues from different datasets and also informs the user when to stop adding new snapshots. The finite precision error analysis of error indicators elucidates that their reliability primarily depends on the condition number of the snapshot matrix. In this case, since the condition number is reasonable, the predicted error should be very close to the actual error. The time taken to

Table 9
Comparison of Strouhal number associated with shear layer modes obtained from FOA based DMD and result of Iyer and Mahesh [27] for different jet velocity to cross flow velocity ratios (R) and the predicted error associated with DMD eigenvalues and eigenvectors.

R	[27]	FOA based DMD	Error_{DMD}
2	0.6255	0.6263	0.0194
2	1.2077	1.2071	0.0323
4	0.3804	0.3804	0.0545
4	0.7624	0.7621	0.0518



(a) R=2, St=0.63

(b) R=2, St=1.21

(c) R=4, St=0.38

(d) R=4, St=0.76

Fig. 6. Real part of DMD eigenmode associated with shear layer. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

generate the Arnoldi vectors and projected matrix from the input snapshot matrix is approximately 234 s for 250 snapshots and 25 s for 80 snapshots on Stampede2 Skylake cluster.

5. Conclusion

In this paper, we develop a modified version of the standard FOA method, which forms the kernel of the proposed FOA based DMD algorithm (Algorithm 1). The streaming form of the proposed methodology is shown in Algorithm 3. The cost and memory consumption of FOA based DMD is smaller than that of SVD based DMD [6] and streaming DMD [8]. From finite precision error analysis of the FOA based DMD, the accuracy of computed projection of the linear mapping A is shown to be $O(\kappa_2(X_1^{N-1})\epsilon_m)$ which is same as that for SVD based methods. The finite precision error of streaming DMD is shown to be $O\left(\left(\kappa_2(X_1^{N-1})\right)^2\epsilon_m\right)$ at most. So, DMD methods which are theoretically equivalent can have different finite precision error. These error estimates are validated by considering snapshot matrices of logarithmically increasing condition number. For snapshots with extremely large condition numbers of $O(\frac{1}{\epsilon_m})$, rank truncation may be used within the FOA based DMD algorithm. The increase in accuracy of computed projection onto a subspace of Krylov subspace in rank truncated FOA based DMD method is explained using finite precision error analysis. Error indicators for the computed DMD eigenvectors and eigenvalues are derived for FOA based DMD with and without rank truncation. These error indicators can be used to devise stopping criterion for DMD. Exact reconstruction property and parallel implementation aspects of the proposed method are discussed. The method is easily parallelizable. Scaling of the algorithm is shown upto 16384 processors. The proposed algorithm is tested on three cases of increasing levels of dimensionality and condition number of snapshot matrix. The proposed error indicators are validated and bounds of accuracy of computed projection is assessed using linearized channel flow simulation snapshots at $Re = 10000$. The DMD eigenvectors and eigenvalues are extracted from cylinder simulation snapshots at $Re = 100$ using FOA based DMD and compared with those obtained by SVD based and streaming DMD methods. The capability of the method to perform DMD on very large scale datasets is shown by performing DMD of snapshots obtained from DNS of turbulent jet in cross flow at two different jet velocity to cross flow velocity ratios.

FOA based DMD algorithm is well-suited for DMD of large datasets on parallel computing platforms as it is a streaming method. Finite precision error analysis is important for DMD algorithms as they rely on snapshot vectors and not on the linear mapping. DMD algorithms which are equivalent in theory can have different finite precision error. The error indicators for modes computed from FOA based DMD provides a quantitative means to compare the accuracy of modes computed from different datasets and decide when to stop acquiring new snapshots. Since FOA based DMD relies on the Arnoldi method with full orthogonalization, the method can utilize the advancements made in Arnoldi-based methods for large eigenvalue problems [20] in the context of DMD.

Acknowledgements

The authors would like to acknowledge support by AFOSR under grant FA9550-15-1-0261. The scaling study was made possible through computing resources provided by Minnesota Supercomputing Institute (MSI) and Texas Advanced Computing Center (TACC). The authors would like to thank Dr. Prahladh Iyer for providing the jets in cross flow dataset.

Appendix A. Computational cost and memory requirement

A.1. FOA based DMD

First, we consider the cost of FOA based DMD without rank truncation to obtain the projected matrices for the batch processed algorithm shown in Algorithm 1. The streaming algorithm has the same floating point operation count as the batch processed one.

$$\begin{aligned}&\text{Cost of FOA based DMD without rank truncation}\\&\approx \underbrace{3M}_{\text{step 2}} + \sum_{j=1}^{N-1}\left[\underbrace{2(j-1)}_{\text{step 5}} + \underbrace{\sum_{i=2}^{j} 2(j-i+1)}_{\text{step 6-8}} + \underbrace{2Mj+M}_{\text{step 9}} + \underbrace{8Mj+j}_{\text{step 10-14}} + \underbrace{3M}_{\text{step 15}} + \underbrace{2(j+1)}_{\text{step 16-18}}\right],\\&\approx 5MN(N-1) + (4M)(N-1) + \frac{N(N-1)(2N-1)}{6} + 3N(N-1) + 3M,\\&\approx 5MN^2 + \frac{N^3}{3}\ (\text{neglecting lower order terms}).\end{aligned} \tag{A.1}$$

Note that in the last step of the above equation we have only considered terms which are cubic in the dimension of the problem. The cost of obtaining the projected matrix using FOA based DMD with rank truncation would include additional costs of SVD and matrix multiplications, i.e.,

$$\begin{aligned}&\text{Cost of FOA based DMD with rank truncation}\\&= \text{Cost of FOA based DMD without rank truncation} + \text{Cost of SVD and matrix multiplication}\\&\approx \underbrace{5MN^2 + \frac{N^3}{3}}_{\text{FOA based DMD}} + \underbrace{12N^3}_{\text{step 22 [25]}} + \underbrace{2r(N-1)^2 + 2r^2(N-1)}_{\text{step 23}}\\&\approx 5MN^2 + \frac{37}{3}N^3 + 2rN^2 + 2r^2N\ (\text{neglecting lower order terms}),\end{aligned} \tag{A.2}$$

where r is the truncated rank.

The memory requirement (number of floating point numbers to be stored) of streaming FOA based DMD without rank truncation involves storing the Arnoldi vectors V_1^N, one upper Hessenberg matrix with an additional row $\bar{H}_N$ and upper triangular matrix β_N, i.e.,

$$\begin{aligned}&\text{Memory requirement of streaming FOA based DMD without rank truncation}\\&=\underbrace{MN}_{V_1^N}+\underbrace{\frac{(N-1)(N+2)}{2}}_{\bar{H}_N}+\underbrace{\frac{N(N+1)}{2}}_{\beta_N}\\&\approx MN+N^2\ \text{(including only quadratic terms)}\end{aligned}\tag{A.3}$$

Note that in the implementation, new snapshots which are to be processed can be stored as columns of V_1^N which are then overwritten by the computed orthonormal Arnoldi vectors. With rank truncation, additional memory requirement involves the left singular vectors, the singular values and the new projected full matrix.

$$\begin{aligned}&\text{Memory requirement of streaming FOA based DMD with rank truncation}\\&\approx MN+N^2+\underbrace{(N-1)^2}_{\text{left singular vectors}}+\underbrace{(N-1)}_{\text{singular values}}+\underbrace{r^2}_{\text{projected matrix}}\\&\approx MN+2N^2+r^2\ \text{(including only quadratic terms)}\end{aligned}\tag{A.4}$$

A.2. SVD based DMD [6]

SVD based methods involve computation of SVD and matrix multiplications to compute the projected matrix.

$$\begin{aligned}&\text{Cost of SVD based DMD with rank truncation}\\&=\text{Cost of SVD}+\text{Cost of matrix multiplication},\\&\approx\underbrace{6M(N-1)^2+20(N-1)^3}_{\text{SVD [25]}}+\underbrace{2Mr(N-1)+2r^2(N-1)+r^2+r}_{\text{matrix multiplication}},\\&\approx 6MN^2+20N^3+2MrN+2r^2N\ \text{(neglecting lower order terms).}\end{aligned}\tag{A.5}$$

Here, r is the truncated rank. The cost associated without rank truncation which is obtained by setting $r=N-1$ is $\approx 8MN^2+22N^3$.

SVD based DMD requires all the snapshots X_1^N, singular vectors U and W, $N-1$ singular values and the full projected matrix.

$$\begin{aligned}&\text{Memory requirement for SVD based DMD with rank truncation}\\&=\underbrace{MN}_{X_1^N}+\underbrace{M(N-1)}_{U}+\underbrace{(N-1)^2}_{W}+\underbrace{N-1}_{\text{singular values}}+\underbrace{r^2}_{\text{projected matrix}}\\&\approx 2MN+N^2+r^2\ \text{(including only quadratic terms)}\end{aligned}\tag{A.6}$$

If rank truncation is not involved, the number of floating point numbers that needs to be stored is then $\approx 2MN^2+2N^2$.

A.3. Streaming DMD [8]

The method of Hemati et al. [8] without compression computes the projected matrix from snapshot pairs as

$$\begin{aligned}&AX=Y,\quad X=Q_X\tilde{X},\quad Y=Q_Y\tilde{Y},\\&Q_X^HAQ_X=Q_X^HQ_Y\tilde{Y}\tilde{X}^H\left(\tilde{X}\tilde{X}^H\right)^{-1}.\end{aligned}\tag{A.7}$$

The floating point operation count to compute the projection of A, $Q_X^HAQ_X$ comprises of two QR factorizations (with reorthogonalization), incrementally forming the matrices $\tilde{Y}\tilde{X}^H$ and $\tilde{X}\tilde{X}^H$, solving $\tilde{Y}\tilde{X}^H\left(\tilde{X}\tilde{X}^H\right)^{-1}$ and a few matrix–matrix multiplications. In particular,

$$\begin{aligned}
&\text{Cost of streaming DMD without compression}\\
&\approx \underbrace{8M(N-1)^2}_{\text{2 QR factorizations (with reorthogonalization)}} + \underbrace{\frac{2}{3}N(N-1)(2N-1)}_{\text{incrementally forming } \tilde{Y}\tilde{X}^H \text{ and } \tilde{X}\tilde{X}^H} + \underbrace{2(N-1)^3}_{\text{solving } \tilde{Y}\tilde{X}^H\left(\tilde{X}\tilde{X}^H\right)^{-1}}\\
&\quad + \underbrace{2M(N-1)^2 + 2(N-1)^3}_{\text{matrix multiplications}},\\
&\approx 10MN^2 + \frac{16}{3}N^3 \text{ (neglecting lower order terms).}
\end{aligned} \tag{A.8}$$

The number of floating point numbers to be stored consists of two orthogonal matrices Q_X and Q_Y, two upper triangular matrices $\tilde{X}$ and $\tilde{Y}$, two outer product matrices $\tilde{Y}\tilde{X}^H$ and $\tilde{X}\tilde{X}^H$ and the projected matrix, i.e.,

$$\begin{aligned}
&\text{Memory required for streaming DMD without compression}\\
&= \underbrace{2M(N-1)}_{\text{2 orthogonal matrices}} + \underbrace{N(N-1)}_{\text{2 upper triangular matrices}} + \underbrace{2(N-1)^2}_{\text{2 outer product matrices}} + \underbrace{(N-1)^2}_{\text{projected matrix}}\\
&\approx 2MN + 4N^2 \text{ (including only quadratic terms)}
\end{aligned} \tag{A.9}$$

The authors do note that a special implementation of streaming DMD can be created for sequence of snapshots which has lower computational cost and memory requirement, but the method as is, consumes the above deduced cost and memory consumption.

It is important to note that the method of Hemati et al. [8] with compression is not the same as SVD based DMD and FOA based DMD with rank truncation. The compressed streaming DMD of Hemati et al. [8] makes an additional approximation that the rank truncated snapshot pairs are related by the same linear mapping A that relates the snapshot pairs, i.e.,

$$AX_r = Y_r + U_Y\Sigma_Y W_Y^H\left(W_{Xr}W_{Xr}^H - W_{Yr}W_{Yr}^H\right) \tag{A.10}$$

where $X = U_X\Sigma_X W_X^H$, $Y = U_Y\Sigma_Y W_Y^H$, $U_{Xr} := U_X(:, 1{:}r)$, $U_{Yr} := U_Y(:, 1{:}r)$, $\Sigma_{Xr} := \Sigma_X(1{:}r, 1{:}r)$, $\Sigma_{Yr} := \Sigma_Y(1{:}r, 1{:}r)$, $W_{Xr} := W_X(:, 1{:}r)$, $W_{Yr} := W_Y(:, 1{:}r)$, $X_r := U_{Xr}\Sigma_r W_{Xr}^H$ and $Y_r := U_{Xr}\Sigma_r W_{Xr}^H$. In the compressed version of streaming DMD [8], the second term in the right hand side of the above equation is neglected and then the Galerkin projection is performed to obtain approximate eigenvectors of A in the range of U_r. In the case of rank truncated SVD based DMD and FOA based DMD, the second term in right hand side is not neglected and is followed by Galerkin projection to obtain approximate eigenvectors in the range of U_r.

Appendix B. Backward error of FOA based DMD

Rewriting Equation (18),

$$A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\bar{H}}_N\hat{\beta}_{N-1} = A\left(\hat{V}_1^{N-1}\hat{\beta}_{N-1} - X_1^{N-1}\right) + \left(X_2^N - \hat{V}_1^N\hat{\beta}_{1:N,2:N}\right) + \hat{V}_1^N\left(\hat{\beta}_{1:N,2:N} - \hat{\bar{H}}_N\hat{\beta}_{N-1}\right) \tag{B.1}$$

From Equation (15), we see that $\beta_{1:N,2:N}$ is a matrix product of $\bar{H}_N$ and β_{N-1}. Using finite precision arithmetic result of matrix multiplication [21],

$$\|\hat{\beta}_{1:N,2:N} - \hat{\bar{H}}_N\hat{\beta}_{N-1}\|_F \leq \gamma_{N-1}\|\hat{\bar{H}}_N\|_F\|\hat{\beta}_{N-1}\|_F, \tag{B.2}$$

where $\gamma_k := \frac{k\epsilon_m}{1-k\epsilon_m}$. Next, we estimate $X_1^N - \hat{V}_1^N\hat{\beta}_N$ using finite precision inner product estimate for the first column and the finite precision arithmetic Arnoldi relation for the remaining columns.

For the first column $\psi_1 - \hat{v}_1\hat{\beta}_{1,1}$,

$$\begin{aligned}
&\hat{\beta}_{1,1} = fl(\|\psi\|_2) = \|\psi\|_2\,(1+\theta_{M+1}),\\
&\hat{v}_1 = fl\left(\frac{\psi_1}{\hat{\beta}_{1,1}}\right) = \frac{\psi_1}{\|\psi_1\|_2}\frac{(1+\theta_1)}{(1+\theta_{M+1})} = \frac{\psi_1}{\|\psi_1\|_2}(1+\theta_{2M+3})\left(\because \frac{1+\theta_k}{1+\theta_j} = 1+\theta_{k+2j} \text{ for } j>k\right),\\
&\psi_1 - \hat{v}_1\hat{\beta}_{1,1} = -\psi_1\theta_{3M+4},
\end{aligned} \tag{B.3}$$

where θ_k is a real number such that $|\theta_k| \leq \gamma_k$.

For the remaining columns of $X_1^N - \hat{V}_1^N\hat{\beta}_N$, we first recognize that FOA based DMD is QR factorization of the matrix $[v_1, L_1, \ldots, L_{N-1}]$ in exact arithmetic, where $L_j := \frac{1}{\beta_{j,j}}\left(\psi_{j+1} - v_1\sum_{i=1}^{j-1}h_{1,i}\beta_{i,j} - \sum_{k=2}^{j}v_k\sum_{i=k-1}^{j-1}h_{k,i}\beta_{i,j}\right)$. From Equation (16), we have in finite precision arithmetic,

$$\begin{aligned}&\left[\hat{v}_1, \hat{L}_1, \dots, \hat{L}_{N-1}\right] = \hat{V}_1^N \hat{R}_N + \left[0, E^{AR}\right],\\ &\|E^{AR}\|_F \le c_1 \epsilon_m \|\hat{L}_1, \dots, \hat{L}_{N-1}\|_F,\end{aligned} \tag{B.4}$$

where, $\hat{L}_j := fl\left(\frac{1}{\hat{\beta}_{j,j}}\left(\psi_{j+1} - \hat{v}_1 fl\left(\sum_{i=1}^{j-1}\hat{h}_{1,i}\hat{\beta}_{i,j}\right) - \sum_{k=2}^{j}\hat{v}_k fl\left(\sum_{i=k-1}^{j-1}\hat{h}_{k,i}\hat{\beta}_{i,j}\right)\right)\right)$ is the computed counterpart of L_j, $\hat{R}_N := [e_1, \hat{\hat{H}}_N]$ where $e_1 \in \mathbb{R}^N$ is the first canonical basis vector and jth column of E^{AR} is the error in the jth step of Arnoldi, i.e.,

$$fl\left(\frac{1}{\hat{\beta}_{j,j}}\left(\psi_{j+1} - \hat{v}_1 fl\left(\sum_{i=1}^{j-1}\hat{h}_{1,i}\hat{\beta}_{i,j}\right) - \sum_{k=2}^{j}\hat{v}_k fl\left(\sum_{i=k-1}^{j-1}\hat{h}_{k,i}\hat{\beta}_{i,j}\right)\right)\right) = \sum_{i=1}^{j+1}\hat{h}_{i,j}\hat{v}_i + e_j^{AR}, \tag{B.5}$$

where e_j^{AR} is the jth column of E^{AR}. Separating the finite precision error in the evaluation of left hand side as

$$\begin{aligned}e_j^{LHS} := &\frac{1}{\hat{\beta}_{j,j}}\left(\psi_{j+1} - \hat{v}_1 fl\left(\sum_{i=1}^{j-1}\hat{h}_{1,i}\hat{\beta}_{i,j}\right) - \sum_{k=2}^{j}\hat{v}_k fl\left(\sum_{i=k-1}^{j-1}\hat{h}_{k,i}\hat{\beta}_{i,j}\right)\right)\\ &- fl\left(\frac{1}{\hat{\beta}_{j,j}}\left(\psi_{j+1} - \hat{v}_1 fl\left(\sum_{i=1}^{j-1}\hat{h}_{1,i}\hat{\beta}_{i,j}\right) - \sum_{k=2}^{j}\hat{v}_k fl\left(\sum_{i=k-1}^{j-1}\hat{h}_{k,i}\hat{\beta}_{i,j}\right)\right)\right),\end{aligned} \tag{B.6}$$

we have

$$\frac{1}{\hat{\beta}_{j,j}}\left(\psi_{j+1} - \hat{v}_1 fl\left(\sum_{i=1}^{j-1}\hat{h}_{1,i}\hat{\beta}_{i,j}\right) - \sum_{k=2}^{j}\hat{v}_k fl\left(\sum_{i=k-1}^{j-1}\hat{h}_{k,i}\hat{\beta}_{i,j}\right)\right) = \sum_{i=1}^{j+1}\hat{h}_{i,j}\hat{v}_i + e_j^{AR} + e_j^{LHS}. \tag{B.7}$$

Rearranging, adding and subtracting $\sum_{i=1}^{j+1}\hat{\beta}_{i,j+1}\hat{v}_i$,

$$\begin{aligned}\psi_{j+1} - \sum_{i=1}^{j+1}\hat{\beta}_{i,j+1}\hat{v}_i = &\left(-\hat{\beta}_{1,j+1} + fl\left(\sum_{i=1}^{j-1}\hat{h}_{1,i}\hat{\beta}_{i,j}\right) + \hat{h}_{1,j}\hat{\beta}_{j,j}\right)\hat{v}_1\\ &+ \sum_{k=2}^{j}\left(-\hat{\beta}_{k,j+1} + fl\left(\sum_{i=k-1}^{j-1}\hat{h}_{k,i}\hat{\beta}_{i,j}\right) + \hat{h}_{k,j}\hat{\beta}_{j,j}\right)\hat{v}_k\\ &+ \left(-\hat{\beta}_{j+1,j+1} + \hat{h}_{j+1,j}\hat{\beta}_{j,j}\right)\hat{v}_{j+1}\\ &+ e_j^{AR}\hat{\beta}_{j,j} + e_j^{LHS}\hat{\beta}_{j,j}.\end{aligned} \tag{B.8}$$

Estimating the size of each entry in $\psi_{j+1} - \sum_{i=1}^{j+1}\hat{\beta}_{i,j+1}\hat{v}_i$ using $\hat{\beta}_{i,j+1} = fl\left(fl\left(\sum_{k=1}^{j-1}\hat{h}_{i,k}\hat{\beta}_{k,j}\right) + fl\left(\hat{h}_{i,j}\hat{\beta}_{j,j}\right)\right)$; $i = 1, \dots, j$, $\hat{\beta}_{j+1,j+1} = fl\left(\hat{h}_{j+1,j}\hat{\beta}_{j,j}\right)$ and triangle inequality,

$$\begin{aligned}|\psi_{j+1} - \sum_{i=1}^{j+1}\hat{\beta}_{i,j+1}\hat{v}_i| \le &\left(|fl\left(\sum_{i=1}^{j-1}\hat{h}_{1,i}\hat{\beta}_{i,j}\right)| + |\hat{h}_{1,j}||\hat{\beta}_{j,j}|\right)\gamma_2|\hat{v}_1|\\ &+ \sum_{k=2}^{j}\left(|fl\left(\sum_{i=k-1}^{j-1}\hat{h}_{k,i}\hat{\beta}_{i,j}\right)| + |\hat{h}_{k,j}||\hat{\beta}_{j,j}|\right)\gamma_2|\hat{v}_k|\\ &+ |\hat{h}_{j+1,j}||\hat{\beta}_{j,j}|\epsilon_m|\hat{v}_{j+1}|\\ &+ |e_j^{AR}||\hat{\beta}_{j,j}| + |e_j^{LHS}||\hat{\beta}_{j,j}|.\end{aligned} \tag{B.9}$$

Then, using the finite precision dot product error estimates $|fl\left(\sum_{i=1}^{j-1}\hat{h}_{1,i}\hat{\beta}_{i,j}\right) - \sum_{i=1}^{j-1}\hat{h}_{1,i}\hat{\beta}_{i,j}| \le \gamma_{j-1}\sum_{i=1}^{j-1}|\hat{h}_{1,i}||\hat{\beta}_{i,j}|$ and $|fl\left(\sum_{i=k-1}^{j-1}\hat{h}_{k,i}\hat{\beta}_{i,j}\right) - \sum_{i=k-1}^{j-1}\hat{h}_{k,i}\hat{\beta}_{i,j}| \le \gamma_{j-k+1}\sum_{i=k-1}^{j-1}|\hat{h}_{k,i}||\hat{\beta}_{i,j}|$; $k = 2, \dots, j$,

$$\begin{aligned}|\psi_{j+1} - \sum_{i=1}^{j+1}\hat{\beta}_{i,j+1}\hat{v}_i| \le &\gamma_2\sum_{i=1}^{j}|\hat{h}_{1,i}||\hat{\beta}_{i,j}||\hat{v}_1| + \gamma_2\sum_{k=2}^{j}\sum_{i=k-1}^{j}|\hat{h}_{k,i}||\hat{\beta}_{i,j}||\hat{v}_k| + \epsilon_m|\hat{h}_{j+1,j}||\hat{\beta}_{j,j}||\hat{v}_{j+1}|\\ &+ |e_j^{AR}||\hat{\beta}_{j,j}| + |e_j^{LHS}||\hat{\beta}_{j,j}| + O\left(\epsilon_m^2\right).\end{aligned} \tag{B.10}$$

Since, e_j^{LHS} is the finite precision error in the addition and division operations in computing L_j, we have

$$|\hat{\beta}_{j,j}|\,|e_j^{LHS}| \leq \left(|\psi_{j+1}| + \sum_{i=1}^{j-1} |\hat{h}_{1,i}|\,|\hat{\beta}_{i,j}|\,|\hat{v}_1| + \sum_{k=2}^{j} \sum_{i=k-1}^{j-1} |\hat{h}_{k,i}|\,|\hat{\beta}_{i,j}|\,|\hat{v}_k| \right) \gamma_{j+2} + O\left(\epsilon_m^2\right). \tag{B.11}$$

Using Equation (B.11) in (B.10),

$$\begin{aligned}|\psi_{j+1} - \sum_{i=1}^{j+1} \hat{\beta}_{i,j+1}\hat{v}_i| \leq{}& \gamma_2 \sum_{i=1}^{j} |\hat{h}_{1,i}|\,|\hat{\beta}_{i,j}|\,|\hat{v}_1| + \gamma_2 \sum_{k=2}^{j}\sum_{i=k-1}^{j} |\hat{h}_{k,i}|\,|\hat{\beta}_{i,j}|\,|\hat{v}_k| + \epsilon_m |\hat{h}_{j+1,j}|\,|\hat{\beta}_{j,j}|\,|\hat{v}_{j+1}| + |e_j^{AR}|\,|\hat{\beta}_{j,j}| \\ &+ \left(|\psi_{j+1}| + \sum_{i=1}^{j-1} |\hat{h}_{1,i}|\,|\hat{\beta}_{i,j}|\,|\hat{v}_1| + \sum_{k=2}^{j}\sum_{i=k-1}^{j-1} |\hat{h}_{k,i}|\,|\hat{\beta}_{i,j}|\,|\hat{v}_k| \right) \gamma_{j+2} + O\left(\epsilon_m^2\right); \\ & j = 1, \ldots, N-1.\end{aligned} \tag{B.12}$$

In matrix form, we then have,

$$|X_2^N - \hat{V}_1^N \hat{\beta}_{1:N,2:N}| \leq \gamma_2 |\hat{V}_1^N|\,|\hat{\hat{H}}_N|\,|\hat{\beta}_{N-1}| + |E^{AR}|\,|\hat{\beta}_{N-1}^d| + \gamma_{N+1}\left(|X_2^N| + |\hat{V}_1^{N-1}|\,|\hat{H}_{N-1}|\,|\hat{\beta}_{N-1}^U|\right) + O\,(\epsilon_m^2) \tag{B.13}$$

and

$$\begin{aligned}|X_1^{N-1} - \hat{V}_1^{N-1}\hat{\beta}_{N-1}| \leq{}& \left[\gamma_{3M+4}|\psi_1|, \gamma_2 |\hat{V}_1^{N-1}|\,|\hat{\hat{H}}_{N-1}|\,|\hat{\beta}_{N-2}| + |E^{AR}_{:,1:N-2}|\,|\hat{\beta}_{N-2}^d| \right. \\ & \left. + \gamma_{N+1}\left(|X_2^{N-1}| + |\hat{V}_1^{N-2}|\,|\hat{H}_{N-2}|\,|\hat{\beta}_{N-2}^U|\right)\right] + O\,(\epsilon_m^2),\end{aligned} \tag{B.14}$$

where $\hat{\beta}_j^d$ is the diagonal matrix formed by using the diagonal entries of $\hat{\beta}_j$ and $\hat{\beta}_j^U$ is the strictly upper triangular matrix formed by using the strictly upper triangular entries of $\hat{\beta}_j$.

From Equation (B.4) and using the assumption that $c_1\epsilon_m < 1$, we can bound the backward error E^{AR} using the computed matrix $\hat{\hat{H}}_N$ as

$$\|E^{AR}\|_F \leq \frac{c_1\epsilon_m}{1 - c_1\epsilon_m}\sqrt{N}\|\hat{\hat{H}}_N\|_F. \tag{B.15}$$

Taking Frobenius norm of Equation (B.13) and (B.14), and using Equation (B.15) we get,

$$\begin{aligned}\|X_2^N - \hat{V}_1^N\hat{\beta}_{1:N,2:N}\|_F \leq{}& \gamma_2\sqrt{N}\|\hat{\hat{H}}_N\|_F\|\hat{\beta}_{N-1}\|_F + \frac{c_1\epsilon_m}{1 - c_1\epsilon_m}\sqrt{N}\|\hat{\hat{H}}_N\|_F\|\hat{\beta}_{N-1}\|_F + \\ & \gamma_{N+1}\left(\|A\|_F\|X_1^{N-1}\|_F + \sqrt{N-1}\|\hat{H}_{N-1}\|_F\|\hat{\beta}_{N-1}\|_F\right) + O\left(\epsilon_m^2\right)\end{aligned} \tag{B.16}$$

and

$$\begin{aligned}\|X_1^{N-1} - \hat{V}_1^{N-1}\hat{\beta}_{N-1}\|_F \leq{}& \gamma_{3M+4}\|X_1^{N-1}\|_F + \gamma_2\sqrt{N-1}\|\hat{\hat{H}}_{N-1}\|_F\|\hat{\beta}_{N-2}\|_F + \\ & \frac{c_1\epsilon_m}{1 - c_1\epsilon_m}\sqrt{N-1}\|\hat{\hat{H}}_{N-1}\|_F\|\hat{\beta}_{N-2}\|_F + \gamma_{N+1}\sqrt{N-2}\|\hat{H}_{N-2}\|_F\|\hat{\beta}_{N-2}\|_F + O\left(\epsilon_m^2\right)\end{aligned} \tag{B.17}$$

From Equation (B.17), we see that $\|\beta_{N-1}\|_F \leq \|X_1^{N-1}\|_2\sqrt{N-1} + O\,(\epsilon_m)$ as $\hat{V}_1^{N-1}$ is orthonormal upto machine precision and since $\|\hat{\beta}_{N-2}\|_F \leq \|\hat{\beta}_{N-1}\|_F$,

$$\begin{aligned}\|\hat{\beta}_{1:N,2:N} - \hat{\hat{H}}_N\hat{\beta}_{N-1}\|_F \leq{}& \gamma_{N-1}\sqrt{N-1}\|\hat{\hat{H}}_N\|_F\|X_1^{N-1}\|_2 + O\,(\epsilon_m^2), \\ \|X_2^N - \hat{V}_1^N\hat{\beta}_{1:N,2:N}\|_F \leq{}& \gamma_2 N\|\hat{\hat{H}}_N\|_F\|X_1^{N-1}\|_2 + \frac{c_1\epsilon_m}{1 - c_1\epsilon_m}N\|\hat{\hat{H}}_N\|_F\|X_1^{N-1}\|_2 + \\ & \gamma_{N+1}\left((N-1)\,\|A\|_F\|X_1^{N-1}\|_2 + (N-1)\,\|\hat{H}_{N-1}\|_F\|X_1^{N-1}\|_2\right) + O\left(\epsilon_m^2\right), \\ \|X_1^{N-1} - \hat{V}_1^{N-1}\hat{\beta}_{N-1}\|_F \leq{}& \gamma_{3M+4}\|X_1^{N-1}\|_F + \gamma_2\,(N-1)\,\|\hat{\hat{H}}_{N-1}\|_F\|X_1^{N-1}\|_2 + \\ & \frac{c_1\epsilon_m}{1 - c_1\epsilon_m}\,(N-1)\,\|\hat{\hat{H}}_{N-1}\|_F\|X_1^{N-1}\|_2 + \gamma_{N+1}\,(N-2)\,\|\hat{H}_{N-2}\|_F\|X_1^{N-1}\|_2 + O\left(\epsilon_m^2\right).\end{aligned} \tag{B.18}$$

Using the above equation in Equation (B.1) we get,

$$\|A\hat{V}_1^{N-1}\hat{\beta}_{N-1} - \hat{V}_1^N\hat{\hat{H}}_N\hat{\beta}_{N-1}\|_F \leq C_1\left(\|A\|_2, \|\hat{\hat{H}}_N\|_2, M, N\right)\epsilon_m\|X_1^{N-1}\|_2 + O\left(\epsilon_m^2\right), \tag{B.19}$$

where $C_1\left(\|A\|_2, \|\hat{\hat{H}}_N\|_2, M, N\right)$ is a constant which is a function of $\|A\|_2$, $\|\hat{\hat{H}}_N\|_2$, M and N.

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, arXiv:1702.01453, 2017.

[2] C.W. Rowley, S.T. Dawson, Model reduction for flow analysis and control, Annu. Rev. Fluid Mech. 49 (2017) 387–417.

[3] C.W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, D.S. Henningson, Spectral analysis of nonlinear flows, J. Fluid Mech. 641 (2009) 115–127.

[4] I. Mezić, Analysis of fluid flows via spectral properties of the Koopman operator, Annu. Rev. Fluid Mech. 45 (2013) 357–378.

[5] P. Schmid, J. Sesterhenn, Dynamic mode decomposition of numerical and experimental data, in: APS Meeting Abstracts, 2008.

[6] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28.

[7] J.H. Tu, C.W. Rowley, D.M. Luchtenburg, S.L. Brunton, J.N. Kutz, On dynamic mode decomposition: theory and applications, J. Comput. Dyn. 1 (2014) 391–421.

[8] M.S. Hemati, M.O. Williams, C.W. Rowley, Dynamic mode decomposition for large and streaming datasets, Phys. Fluids 26 (2014) 111701.

[9] T. Sayadi, P.J. Schmid, Parallel data-driven decomposition algorithm for large-scale datasets: with application to transitional boundary layers, Theor. Comput. Fluid Dyn. 30 (2016) 415–428.

[10] J. Demmel, L. Grigori, M. Hoemmen, J. Langou, Communication-optimal parallel and sequential qr and lu factorizations, SIAM J. Sci. Comput. 34 (2012) A206–A239.

[11] K.K. Chen, J.H. Tu, C.W. Rowley, Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses, J. Nonlinear Sci. 22 (2012) 887–915.

[12] M.O. Williams, I.G. Kevrekidis, C.W. Rowley, A data-driven approximation of the Koopman operator: extending dynamic mode decomposition, J. Nonlinear Sci. 25 (2015) 1307–1346.

[13] M.O. Williams, C.W. Rowley, I.G. Kevrekidis, A kernel-based approach to data-driven Koopman spectral analysis, arXiv:1411.2260, 2014.

[14] S.T. Dawson, M.S. Hemati, M.O. Williams, C.W. Rowley, Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition, Exp. Fluids 57 (2016) 42.

[15] B.R. Noack, W. Stankiewicz, M. Morzyński, P.J. Schmid, Recursive dynamic mode decomposition of transient and post-transient wake flows, J. Fluid Mech. 809 (2016) 843–872.

[16] M.R. Jovanović, P.J. Schmid, J.W. Nichols, Sparsity-promoting dynamic mode decomposition, Phys. Fluids 26 (2014) 024103.

[17] A. Wynn, D. Pearson, B. Ganapathisubramani, P.J. Goulart, Optimal mode decomposition for unsteady flows, J. Fluid Mech. 733 (2013) 473–503.

[18] D. Duke, J. Soria, D. Honnery, An error analysis of the dynamic mode decomposition, Exp. Fluids 52 (2012) 529–542.

[19] H. Zhang, S. Dawson, C.W. Rowley, E.A. Deem, L.N. Cattafesta, Evaluating the accuracy of the dynamic mode decomposition, arXiv:1710.00745, 2017.

[20] Y. Saad, Numerical Methods for Large Eigenvalue Problems, revised edition, SIAM, 2011.

[21] N.J. Higham, Accuracy and Stability of Numerical Algorithms, vol. 80, SIAM, 2002.

[22] L. Giraud, J. Langou, M. Rozložník, J. van den Eshof, Rounding error analysis of the classical Gram–Schmidt orthogonalization process, Numer. Math. 101 (2005) 87–100.

[23] L.N. Trefethen, D. Bau III, Numerical Linear Algebra, vol. 50, SIAM, 1997.

[24] Å. Björck, C.C. Paige, Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm, SIAM J. Matrix Anal. Appl. 13 (1992) 176–190.

[25] G.H. Golub, C.F. Van Loan, Matrix Computations, vol. 3, JHU Press, 2012.

[26] J.N. Kutz, S.L. Brunton, B.W. Brunton, J.L. Proctor, Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems, 2016.

[27] P.S. Iyer, K. Mahesh, A numerical study of shear layer characteristics of low-speed transverse jets, J. Fluid Mech. 790 (2016) 275–307.