

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Error analysis of an incremental proper orthogonal decomposition algorithm for PDE simulation data

Hiba Fareed, John R. Singler*

Department of Mathematics and Statistics, Missouri University of Science and Technology, Rolla, MO, United States of America



ARTICLE INFO

Article history:

Received 18 March 2018

Received in revised form 14 May 2019

MSC:

65F

15A18

Keywords:

Proper orthogonal decomposition

Incremental algorithm

Singular value decomposition

Weighted norm

Error analysis

ABSTRACT

In our earlier work Fareed et al. (2018), we proposed an incremental SVD algorithm with respect to a weighted inner product to compute the proper orthogonal decomposition (POD) of a set of simulation data for a partial differential equation (PDE) without storing the data. In this work, we perform an error analysis of the incremental SVD algorithm. We also modify the algorithm to incrementally update both the SVD and an error bound when a new column of data is added. We show the algorithm produces the exact SVD of an approximate data matrix, and the operator norm error between the approximate and exact data matrices is bounded above by the computed error bound. This error bound also allows us to bound the error in the incrementally computed singular values and singular vectors. We illustrate our analysis with numerical results for three simulation data sets from a 1D FitzHugh–Nagumo PDE system with various choices of the algorithm truncation tolerances.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Proper orthogonal decomposition (POD) is a method to find an optimal low order basis to approximate a given set of data. The basis elements are called POD modes, and they are often used to create low order models of high-dimensional systems of ordinary differential equations or partial differential equations (PDEs) that can be simulated easily and even used for real-time applications. For more about the applications of POD in engineering and applied sciences and POD model order reduction, see, e.g., [1–16].

There is a close relationship between the singular value decomposition (SVD) of a set of data and the POD eigenvalues and modes of the data. Due to applications involving functional data and PDEs, many researchers discuss this relationship in weighted inner product spaces and general Hilbert spaces [17–20]. For the POD calculation, it is important to determine an inner product that is appropriate for the application [6,21–24].

Since the size of data sets continues to increase in applications, many researchers have proposed and developed more efficient algorithms for POD computations, the SVD, and other related methods [25–35]. These algorithms have been recently applied in conjunction with techniques such as POD model order reduction and the dynamic mode decomposition, which often consider simulation data from a PDE [24,36–44].

In our earlier work [45], we proposed an incremental SVD algorithm for computing POD eigenvalues and modes in a weighted inner product space. Specifically, we considered Galerkin-type PDE simulation data, initialized the SVD on a small amount of the data, and then used an incremental approach to approximately update the SVD with respect to a weighted inner product as new data arrives. Our incremental approach to update the SVD is an extension of

* Corresponding author.

E-mail addresses: hfn3@mst.edu (H. Fareed), singlerj@mst.edu (J.R. Singler).

Brand's incremental matrix SVD algorithm [25] to treat the weighted inner product without computing any matrix decompositions. The algorithm involves minimal data storage; the PDE simulation data does not need to be stored. The algorithm also involves truncation, and therefore produces approximate POD eigenvalues and modes. We proved the SVD update is exact without truncation.

In this paper, we study the effectiveness of the truncations and deduce error bounds for the SVD approximation. To handle the computational challenge raised by large data sets, we bound the error incrementally. Specifically, we extend the incremental SVD algorithm for a weighted inner product in [45] to compute an error bound incrementally without storing the data set; see Section 2, Algorithm 1. We also perform an error analysis in Section 3 that clarifies the effect of truncation at each step, and provides more insight into the accuracy of the algorithm with truncation and the choices of the two tolerances. We prove the algorithm produces the exact SVD of an approximate data set, and the operator norm error between the exact and approximate data set is bounded above by the incrementally computed error bound. This yields error bounds for the approximate POD eigenvalues and modes. To illustrate the analysis, we present numerical results in Section 4 for a set of PDE simulation data using various choices of the tolerances. Finally, we present conclusions in Section 5.

As far as the authors are aware, the error analysis approach in this work is new and the incremental SVD algorithm for a weighted inner product presented here is the only existing approach with incrementally computed rigorous error bounds.

2. Background and algorithm

We begin by setting notation, recalling background material, and discussing the algorithm.

For a matrix $A \in \mathbb{R}^{m \times n}$, let $A_{(p:q,r:s)}$ denote the submatrix of A consisting of the entries of A from rows p, \dots, q and columns r, \dots, s . Also, if p and q are omitted, then the submatrix should consist of the entries from all rows. A similar convention applies for the columns if r and s are omitted.

Let $M \in \mathbb{R}^{m \times m}$ be symmetric positive definite, and let \mathbb{R}_M^m denote the Hilbert space \mathbb{R}^m with weighted inner product $(x, y)_M = y^T M x$ and corresponding norm $\|x\|_M = (x^T M x)^{1/2}$. For a matrix $P \in \mathbb{R}^{m \times n}$, we can consider P as a linear operator $P : \mathbb{R}^n \rightarrow \mathbb{R}_M^m$. In this case, the operator norm of P is

$$\|P\|_{\mathcal{L}(\mathbb{R}^n, \mathbb{R}_M^m)} = \sup_{\|x\|=1} \|Px\|_M.$$

We note that \mathbb{R}^n without a subscript should be understood to have the standard inner product $(x, y) = y^T x$ and Euclidean norm $\|x\| = (x^T x)^{1/2}$. The Hilbert adjoint operator of the matrix $P : \mathbb{R}^n \rightarrow \mathbb{R}_M^m$ is the matrix $P^* : \mathbb{R}_M^m \rightarrow \mathbb{R}^n$ given by $P^* = P^T M$. We have $(Px, y)_M = (x, P^*y)$ for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}_M^m$.

In our earlier work [45], we discussed how the proper orthogonal decomposition of a set of PDE simulation data can be reformulated as the SVD of a matrix with respect to a weighted inner product. We do not give the details of the reformulation here, but we do briefly recall the SVD with respect to a weighted inner product since we use this concept throughout this work.

Definition 2.1. A core SVD of a matrix $P : \mathbb{R}^n \rightarrow \mathbb{R}_M^m$ is a decomposition $P = V \Sigma W^T$, where $V \in \mathbb{R}^{m \times k}$, $\Sigma \in \mathbb{R}^{k \times k}$, and $W \in \mathbb{R}^{n \times k}$ satisfy

$$V^T M V = I, \quad W^T W = I, \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_k),$$

where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k > 0$. The values $\{\sigma_j\}$ are called the (positive) singular values of P and the columns of V and W are called the corresponding singular vectors of P .

Since POD applications do not typically require the zero singular values, we do not consider the full SVD of $P : \mathbb{R}^n \rightarrow \mathbb{R}_M^m$ in this work. We do note that the SVD of $P : \mathbb{R}^n \rightarrow \mathbb{R}_M^m$ is closely related to the eigenvalue decompositions of P^*P and PP^* . See [45, Section 2.1] for more details.

Also, when we consider the SVD (or core SVD) of a matrix without weighted inner products we refer to this as the standard SVD (or standard core SVD).

We consider approximately computing the SVD of a data set U incrementally by updating the core SVD when each new column c of data is added to the data set. This incremental procedure is performed without forming or storing the original data matrix. Specifically, we focus on the incremental SVD algorithm with a weighted inner product proposed in Algorithm 4 of [45]. The algorithm is based on the following fundamental identity: if $U = V \Sigma W^T$ is a core SVD, then

$$\begin{aligned} [Uc] &= [V \Sigma W^T c] \\ &= [Vj] \begin{bmatrix} \Sigma & V^*c \\ 0 & p \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T, \end{aligned}$$

where $j = (c - VV^*c)/p$ and $p = \|c - VV^*c\|_M$ [45]. The algorithm is a modified version of Brand's incremental SVD algorithm [25] to directly treat the weighted inner product. Brand's incremental SVD algorithm without a weighted

inner product has been used for POD computations in [41,44], and our implementation strategy follows the algorithm in [44].

Below, we consider a slight modification of the algorithm from [45]; specifically, we update the algorithm to include a computable error bound e . We show in this work that the algorithm produces the exact core SVD of a matrix \tilde{U} such that $\|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}_M^m)} \leq e$, where U is the true data matrix. This error bound gives information about the approximation error for the singular values and singular vectors; see Section 3.2 for details.

We take the first step in the incremental SVD algorithm by initializing the SVD and the error bound with a single column $c \neq 0$ as follows:

$$\Sigma = \|c\|_M = (|c^T M c|)^{1/2}, \quad V = c \Sigma^{-1}, \quad W = 1, \quad e = 0.$$

Here, the error bound e is set to zero since the initial SVD is exact. Also, as mentioned in [45], even though M is positive definite it is possible for round off errors to cause $c^T M c$ to be very small and negative; we use the absolute value here and throughout the algorithm to avoid this issue.

Then we incrementally update the SVD and the error bound by applying Algorithm 1 when a new column is added. Most of the algorithm is taken directly from [45, Algorithm 4]; we refer to that work for a detailed discussion of the algorithm and details about the implementation.

We note the following:

- The input is an existing SVD V , Σ , and W , a new column c , the weight matrix M , two positive tolerances, and an error bound e .
- Lines 10, 15, 18, 21, and 26 are new, and are simple computations used to update the error bound e .
- In the SVD update stage (lines 1–16), e_p is the error due to p -truncation in line 3.
- In the singular value truncation stage (lines 17–22), e_{sv} is the error due to the singular value truncation in line 19.
- In the orthogonalization stage (lines 23–25), a modified Gram–Schmidt algorithm with reorthogonalization is used; see Section 4.2 in [45].
- The output is the updated SVD and error bound.
- The columns of V are the M -orthonormal POD modes, and the squares of the singular values are the POD eigenvalues.
- If only the POD eigenvalues and modes are required, then the computations involving W can be skipped; however, W is needed if an approximate reconstruction of the entire data set is desired.
- As new columns continue to be added, a user can monitor the computed error bound and lower the tolerances if desired.

3. Error analysis

In this section, we perform an error analysis of Algorithm 1. We show the algorithm produces the exact SVD of another matrix \tilde{U} , and bound the error between the matrices.

We assume all computations in the algorithm are performed in exact arithmetic. Therefore, the Gram–Schmidt orthogonalization stage (in lines 23–25) is not considered here. We note that in [45], we considered a Gram–Schmidt procedure with reorthogonalization to minimize the effect of round-off errors; see, e.g., [46–49]. We leave an analysis of round-off errors in Algorithm 1 to be considered elsewhere.

We begin our analysis in Section 3.1 by analyzing the error due to each individual truncation step in the algorithm. Then we provide error bounds for the algorithm in Section 3.2.

3.1. Individual truncation errors

We begin our analysis of the incremental SVD algorithm by recalling a result from [45]. This result shows that a single column incremental update to the SVD is exact without truncation when $p = \|c - VV^*c\|_M > 0$.

Theorem 3.1 (Theorem 4.1 in [45]). *Let $U : \mathbb{R}^n \rightarrow \mathbb{R}_M^m$, and suppose $U = V \Sigma W^T$ is an exact core SVD of U , where $V^T M V = I$ for $V \in \mathbb{R}^{m \times k}$, $W^T W = I$ for $W \in \mathbb{R}^{n \times k}$, and $\Sigma \in \mathbb{R}^{k \times k}$. Let $c \in \mathbb{R}_M^m$ and define*

$$h = c - VV^*c, \quad p = \|h\|_M, \quad Q = \begin{bmatrix} \Sigma & V^*c \\ 0 & p \end{bmatrix},$$

where $V^* = V^T M$. If $p > 0$ and a standard core SVD of $Q \in \mathbb{R}^{(k+1) \times (k+1)}$ is given by

$$Q = V_Q \Sigma_Q W_Q^T, \tag{1}$$

then a core SVD of $[U \ c] : \mathbb{R}^{n+1} \rightarrow \mathbb{R}_M^m$ is given by

$$[U \ c] = V_u \Sigma_u W_u^T,$$

Algorithm 1 Incremental SVD and error bound with weighted inner product

Input: $V \in \mathbb{R}^{m \times k}$, $\Sigma \in \mathbb{R}^{k \times k}$, $W \in \mathbb{R}^{n \times k}$, $c \in \mathbb{R}^m$, $M \in \mathbb{R}^{m \times m}$, tol , tol_{sv} , e

```

% Prepare for SVD update
1:  $d = V^T M c$ ,  $p = \text{sqrt}(|(c - Vd)^T M (c - Vd)|)$ 
2: if ( $p < \text{tol}$ ) then
3:    $Q = \begin{bmatrix} \Sigma & d \\ 0 & 0 \end{bmatrix}$ 
4: else
5:    $Q = \begin{bmatrix} \Sigma & d \\ 0 & p \end{bmatrix}$ 
6: end if
7:  $[V_Q, \Sigma_Q, W_Q] = \text{svd}(Q)$ 
   % SVD update
8: if ( $p < \text{tol}$ ) or ( $k \geq m$ ) then
9:    $V = V V_{Q(1:k, 1:k)}$ ,  $\Sigma = \Sigma_{Q(1:k, 1:k)}$ ,  $W = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_{Q(:, 1:k)}$ 
10:   $e_p = p$ 
11: else
12:   $j = (c - Vd)/p$ 
13:   $V = [V j] V_Q$ ,  $\Sigma = \Sigma_Q$ ,  $W = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_Q$ 
14:   $k = k + 1$ 
15:   $e_p = 0$ 
16: end if
   % Neglect small singular values: truncation
17: if ( $\Sigma_{(r,r)} > \text{tol}_{sv}$ ) and ( $\Sigma_{(r+1,r+1)} \leq \text{tol}_{sv}$ ) then
18:   $e_{sv} = \Sigma_{(r+1,r+1)}$ 
19:   $\Sigma = \Sigma_{(1:r, 1:r)}$ ,  $V = V_{(:, 1:r)}$ ,  $W = W_{(:, 1:r)}$ 
20: else
21:   $e_{sv} = 0$ 
22: end if
   % Orthogonalize if necessary
23: if ( $|V_{(:, \text{end})}^T M V_{(:, 1)}| > \min(\text{tol}, \text{tol} \times m)$ ) then
24:   $V = \text{modifiedGSweighted}(V, M)$ 
25: end if
26:  $e = e + e_p + e_{sv}$ 
27: return  $V, \Sigma, W, e$ 

```

where

$$V_u = [V j] V_Q, \quad j = h/p, \quad W_u = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_Q.$$

Next, we analyze the incremental SVD update in the case when the added column c satisfies $p = \|c - VV^*c\|_M = 0$.

Lemma 3.2. Let $U = V \Sigma W^T$, c , h , p , and Q be given as in [Theorem 3.1](#), and assume $p = \|c - VV^*c\|_M = 0$. If the full standard SVD of $Q \in \mathbb{R}^{k+1 \times k+1}$ is given by $Q = V_Q \Sigma_Q W_Q^T$, where $V_Q, \Sigma_Q, W_Q \in \mathbb{R}^{k+1 \times k+1}$, then

$$V_Q = \begin{bmatrix} V_{Q(1:k, 1:k)} & 0 \\ 0 & 1 \end{bmatrix}, \quad \Sigma_Q = \begin{bmatrix} \Sigma_{Q(1:k, 1:k)} & 0 \\ 0 & 0 \end{bmatrix}, \quad \Sigma_{Q(1:k, 1:k)} > 0,$$

and a standard core SVD of $R = Q_{(1:k, 1:k+1)} = [\Sigma \ V^*c] \in \mathbb{R}^{k \times k+1}$ is given by

$$R = V_{Q(1:k, 1:k)} \Sigma_{Q(1:k, 1:k)} (W_{Q(:, 1:k)})^T.$$

Proof. Let $\sigma_{Q_1} \geq \sigma_{Q_2} \geq \dots \geq \sigma_{Q_{k+1}} \geq 0$ be the singular values of Q so that $\Sigma_Q = \text{diag}(\sigma_{Q_1}, \dots, \sigma_{Q_{k+1}})$. Also, let $\{v_{Q_j}\}$ and $\{w_{Q_j}\}$ be the corresponding orthonormal singular vectors in \mathbb{R}^{k+1} , so that

$$V_Q = [v_{Q_1}, \dots, v_{Q_{k+1}}], \quad W_Q = [w_{Q_1}, \dots, w_{Q_{k+1}}],$$

with $V_Q^T V_Q = I$ and $W_Q^T W_Q = I$.

First, we show Q has exactly one zero singular value. Since we know

$$Q^T v_{Q_j} = \sigma_{Q_j} w_{Q_j}, \tag{2}$$

$$Q w_{Q_j} = \sigma_{Q_j} v_{Q_j}, \tag{3}$$

for $j = 1, \dots, k + 1$, the number of zero singular values of Q is precisely equal to the dimension of the nullspace of Q^T . Suppose $v = [v_1, \dots, v_{k+1}]^T \in \mathbb{R}^{k+1}$ satisfies $Q^T v = 0$. Recall $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k) > 0$, and let $d = V^*c = [d_1, \dots, d_k]^T$. Then $Q^T v = 0$ implies

$$\begin{bmatrix} \sigma_1 v_1 \\ \sigma_2 v_2 \\ \vdots \\ \sigma_k v_k \\ d_1 v_1 + d_2 v_2 + \dots + d_k v_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

Since $\sigma_1 \geq \dots \geq \sigma_k > 0$, we have $v_j = 0$ for $j = 1, \dots, k$. This implies the nullspace of Q^T is exactly the span of $e_{k+1} = [0, \dots, 0, 1]^T \in \mathbb{R}^{k+1}$. Therefore, the nullspace is one dimensional and Q has exactly one zero singular value, i.e., $\sigma_{Q_{k+1}} = 0$ and $\sigma_{Q_1} \geq \sigma_{Q_2} \geq \dots \geq \sigma_{Q_k} > 0$.

Next, $Q w_{Q_j} = \sigma_j v_{Q_j}$ for $j = 1, \dots, k$ gives

$$\Rightarrow \begin{bmatrix} \sigma_1 w_{Q_{j,1}} + d_1 w_{Q_{j,k+1}} \\ \sigma_2 w_{Q_{j,2}} + d_2 w_{Q_{j,k+1}} \\ \vdots \\ \sigma_k w_{Q_{j,k}} + d_k w_{Q_{j,k+1}} \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma_j v_{Q_{j,1}} \\ \sigma_j v_{Q_{j,2}} \\ \vdots \\ \sigma_j v_{Q_{j,k}} \\ \sigma_j v_{Q_{j,k+1}} \end{bmatrix}.$$

The last equation gives $v_{Q_{j,k+1}} = 0$ since $\sigma_j > 0$ for $j = 1, \dots, k$. Therefore, for $j = 1, \dots, k$,

$$v_{Q_j} = [v_{Q_{j,1}}, v_{Q_{j,2}}, \dots, v_{Q_{j,k}}, 0]^T,$$

and

$$v_{Q_{k+1}} = [0, 0, \dots, 0, 1]^T.$$

This implies

$$V_Q = \begin{bmatrix} V_{Q(1:k, 1:k)} & 0 \\ 0 & 1 \end{bmatrix},$$

and so the SVD decomposition of Q is given by

$$Q = \begin{bmatrix} V_{Q(1:k, 1:k)} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Sigma_{Q(1:k, 1:k)} & 0 \\ 0 & 0 \end{bmatrix} W_Q^T.$$

This gives $R = Q_{(1:k, 1:k+1)} = \check{V}_Q \check{\Sigma}_Q \check{W}_Q^T$, where $\check{V}_Q = V_{Q(1:k, 1:k)}$, $\check{\Sigma}_Q = \Sigma_{Q(1:k, 1:k)}$, and $\check{W}_Q = W_{Q(1:k+1, 1:k)}$. It can be checked that $\check{V}_Q^T \check{V}_Q = I$ and $\check{W}_Q^T \check{W}_Q = I$ since $V_Q^T V_Q = I$ and $W_Q^T W_Q = I$. Therefore, a standard core SVD of $R \in \mathbb{R}^{k \times k+1}$ is given by $R = \check{V}_Q \check{\Sigma}_Q \check{W}_Q^T$. \square

The following result is nearly identical to Proposition 2.3 in [45]; the proof is also almost identical and is omitted.

Lemma 3.3 (Proposition 2.3 in [45]). Suppose $V_u \in \mathbb{R}^{m \times k}$ has M -orthonormal columns and $W_u \in \mathbb{R}^{n \times l}$ has orthonormal columns. If $R \in \mathbb{R}^{k \times l}$ has standard core SVD $R = V_R \Sigma_R W_R^T$ and $P : \mathbb{R}^n \rightarrow \mathbb{R}_M^n$ is defined by $P = VRW^T$, then

$$P = V_u \Sigma_u W_u^T, \quad V_u = V V_R, \quad \Sigma_u = \Sigma_R, \quad W_u = W W_R, \tag{4}$$

is a core SVD of P .

Next, we complete the analysis of the $p = 0$ case:

Proposition 3.4. Let $U = V \Sigma W^T$, c , h , p , and Q be given as in Theorem 3.1, and assume $p = \|c - VV^*c\|_M = 0$. If the full standard SVD of $Q \in \mathbb{R}^{k+1 \times k+1}$ is given by $Q = V_Q \Sigma_Q W_Q^T$, where $V_Q, \Sigma_Q, W_Q \in \mathbb{R}^{k+1 \times k+1}$, then a core SVD of $[U \ c] : \mathbb{R}^{n+1} \rightarrow \mathbb{R}_M^m$ is given by

$$[U \ c] = V_u \Sigma_u W_u^T,$$

where

$$V_u = VV_{Q(1:k, 1:k)}, \quad \Sigma_u = \Sigma_{Q(1:k, 1:k)}, \quad W_u = \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix} W_{Q(:, 1:k)}.$$

Proof. Since $p = 0$, we have $c = VV^*c$ and therefore

$$[U \ c] = [V \Sigma W^T \quad VV^*c] = V \begin{bmatrix} \Sigma & V^*c \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & 1 \end{bmatrix}^T.$$

The result follows from Lemmas 3.2 and 3.3 by taking $P = [U \ c]$ and $R = [\Sigma \ V^*c]$. \square

Truncation part 1. Next, we analyze the incremental SVD update in the case when the added column c satisfies $p = \|c - VV^*c\|_M < \text{tol}$. In this case, Algorithm 1 does not compute the SVD of $[U \ c]$. Instead, Algorithm 1 sets $p = 0$ and returns the exact SVD of $\tilde{U} = [U \ VV^*c]$. The approximation error in the operator norm is given in the next result.

Proposition 3.5. Let $U : \mathbb{R}^n \rightarrow \mathbb{R}_M^m$, and suppose $U = V \Sigma W^T$ is a core SVD of U . If $c \in \mathbb{R}_M^m$, $p = \|c - VV^*c\|_M$, and

$$\tilde{U} = [U \ VV^*c],$$

then

$$\|[U \ c] - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^{n+1}, \mathbb{R}_M^m)} = p.$$

Proof. For $x = [x_1, \dots, x_{n+1}]^T \in \mathbb{R}^{n+1}$, we have

$$\begin{aligned} \|[U \ c] - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^{n+1}, \mathbb{R}_M^m)} &= \sup_{\|x\|=1} \|[0 \ (c - VV^*c)]x\|_M \\ &= \sup_{\|x\|=1} \|c - VV^*c\|_M |x_{n+1}| \\ &= \|c - VV^*c\|_M, \end{aligned}$$

where the sup is clearly attained by $x = [0, \dots, 0, 1]^T \in \mathbb{R}^{n+1}$. \square

Truncation part 2. In Algorithm 1, after the SVD update due to an added column the algorithm truncates any singular values that are smaller than a given tolerance, tol_{sv} . For the matrix case with unweighted inner products, the operator norm error caused by this truncation is well-known to equal the first neglected singular value. This result is also true for a compact linear operator mapping between two Hilbert spaces; see, e.g., [50, Chapters VI–VIII], [51, Chapter 30], and [52, Sections VI.5–VI.6] for more information about the SVD for compact operators. This gives the following result:

Proposition 3.6. Let $U : \mathbb{R}^n \rightarrow \mathbb{R}_M^m$, and suppose $U = V \Sigma W^T$ is a core SVD of U . For a given $r > 0$, let \tilde{U} be the rank r truncated SVD of U , i.e.,

$$\tilde{U} = V_{(:, 1:r)} \Sigma_{(1:r, 1:r)} (W_{(:, 1:r)})^T.$$

Then

$$\|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^n, \mathbb{R}_M^m)} = \Sigma_{(r+1, r+1)}.$$

3.2. Error bounds

Next, we fully explain the computed error bound in Algorithm 1. In a typical application of the algorithm, many new columns of data are added and the POD is updated many times. In the following result, we assume we are at the k th step of this procedure and we have an existing error bound. We prove that Algorithm 1 produces a correct update of the error bound.

More specifically, let $k \in \mathbb{N}$, let $U_k, \tilde{U}_k : \mathbb{R}^k \rightarrow \mathbb{R}_M^m$, and assume

$$U_k = V_k \Sigma_k W_k^T, \quad \tilde{U}_k = \tilde{V}_k \tilde{\Sigma}_k \tilde{W}_k^T$$

are core SVDs of U and \tilde{U} . Let $c_k \in \mathbb{R}_M^m$ and define $U_{k+1} := [U_k \ c_k] : \mathbb{R}^{k+1} \rightarrow \mathbb{R}_M^m$. Furthermore, let $\tilde{U}_{k+1} : \mathbb{R}^{k+1} \rightarrow \mathbb{R}_M^m$ be the result of one step of the incremental SVD update applied to \tilde{U}_k so that

$$\tilde{U}_{k+1} = \tilde{V}_{k+1} \tilde{\Sigma}_{k+1} \tilde{W}_{k+1}^T.$$

Therefore, we consider the sequence $\{U_k\}$ to be the exact data matrices, and the sequence $\{\tilde{U}_k\}$ to be the result produced (in exact arithmetic) by Algorithm 1.

In exact arithmetic, there are two stages to Algorithm 1. The first stage is the SVD update in lines 1–16. This stage of the algorithm takes \tilde{U}_k and the added column c and produces the update \hat{U}_{k+1} . There are two possible results for \hat{U}_{k+1} depending on the value of p in line 1. The second stage is the singular value truncation applied to \hat{U}_{k+1} (lines 17–22), which produces the final update \tilde{U}_{k+1} . Again, there are two possible results for \tilde{U}_{k+1} , depending on the singular values of \hat{U}_{k+1} . We analyze the error bound for each possible outcome of the algorithm in the result below.

Let the positive tolerances tol and tol_{sv} be fixed. Below, we let p_k denote the value p in line 1 of Algorithm 1. We say that p truncation is applied if $p_k < \text{tol}$. We say the singular value truncation is applied if any of the singular values of \hat{U}_{k+1} are less than tol_{sv} . In this case, we find a value r so that the first r largest singular values of \hat{U}_{k+1} are greater than tol_{sv} , while the remaining singular values are less than or equal to tol_{sv} . We let $\hat{\sigma}_{r+1}$ denote the largest singular value of \hat{U}_{k+1} such that $\hat{\sigma}_{r+1} \leq \text{tol}_{sv}$.

Theorem 3.7. *If*

$$\|U_k - \tilde{U}_k\|_{\mathcal{L}(\mathbb{R}^k, \mathbb{R}_M^m)} \leq e_k, \quad p_k = \|c_k - \tilde{V}_k \tilde{V}_k^* c_k\|_M,$$

then

$$\|U_{k+1} - \tilde{U}_{k+1}\|_{\mathcal{L}(\mathbb{R}^{k+1}, \mathbb{R}_M^m)} \leq e_{k+1},$$

where

$$e_{k+1} = \begin{cases} e_k, & \text{if no truncation is applied,} \\ e_k + p_k, & \text{if only } p \text{ truncation is applied,} \\ e_k + \hat{\sigma}_{r+1}, & \text{if only the singular value truncation is applied,} \\ e_k + p_k + \hat{\sigma}_{r+1}, & \text{if both truncations are applied.} \end{cases}$$

Proof. Stage 1 of Algorithm 1 (lines 1–16) takes \tilde{U}_k and produces \hat{U}_{k+1} . If $p_k \geq \text{tol}$, then Theorem 3.1 gives that the core SVD is updated exactly, i.e.,

$$\hat{U}_{k+1} = [\tilde{U}_k \ c_k] \quad \text{if } p_k \geq \text{tol}.$$

Otherwise, if $p_k < \text{tol}$, then Proposition 3.5 implies

$$\hat{U}_{k+1} = [\tilde{U}_k \ \tilde{V} \tilde{V}^* c_k] \quad \text{if } p_k < \text{tol},$$

and the error is given by

$$\|[\tilde{U}_k \ c_k] - \hat{U}_{k+1}\|_{\mathcal{L}(\mathbb{R}^{k+1}, \mathbb{R}_M^m)} = p_k.$$

Stage 2 of Algorithm 1 (lines 17–22) takes \hat{U}_{k+1} and produces \tilde{U}_{k+1} . If all of the singular values of \hat{U}_{k+1} are greater than tol_{sv} , then $\tilde{U}_{k+1} = \hat{U}_{k+1}$ and there is no error in this stage. Otherwise, let $\hat{\sigma}_{r+1}$ denote the largest singular value of \hat{U}_{k+1} such that $\hat{\sigma}_{r+1} \leq \text{tol}_{sv}$. In this case, \tilde{U}_{k+1} is simply the r th order truncated SVD of \hat{U}_{k+1} , and the error is given by Proposition 3.6:

$$\|\tilde{U}_{k+1} - \hat{U}_{k+1}\|_{\mathcal{L}(\mathbb{R}^{k+1}, \mathbb{R}_M^m)} = \hat{\sigma}_{r+1}.$$

Below, for ease of notation, let $\|\cdot\|$ denote the $\mathcal{L}(\mathbb{R}^{k+1}, \mathbb{R}_M^m)$ operator norm. The error between U_{k+1} and \tilde{U}_{k+1} in the operator norm can be bounded as follows:

$$\|U_{k+1} - \tilde{U}_{k+1}\| \leq \|U_{k+1} - [\tilde{U}_k \ c_k]\| + \|[\tilde{U}_k \ c_k] - \hat{U}_{k+1}\| + \|\hat{U}_{k+1} - \tilde{U}_{k+1}\|.$$

As noted above, the second error term is either zero if p truncation is not applied or p_k otherwise. Also, the third error term is either zero if the singular values truncation is not applied or $\hat{\sigma}_{r+1}$ otherwise. For the first term, we have

$$\begin{aligned} \|U_{k+1} - [\tilde{U}_k \ c_k]\| &= \|[U_k \ c_k] - [\tilde{U}_k \ c_k]\| \\ &= \|(U_k - \tilde{U}_k) \mathbf{0}\| \\ &= \sup_{\|x\|=1} \|[(U_k - \tilde{U}_k) \mathbf{0}] x\|_M \\ &\leq \|U_k - \tilde{U}_k\|_{\mathcal{L}(\mathbb{R}^k, \mathbb{R}_M^m)} \leq e_k. \end{aligned}$$

This completes the proof. \square

The result above explains the update of the error bound in one step of Algorithm 1. Now we assume the SVD is initialized exactly when $k = 1$, and then the algorithm is applied for a sequence of added columns $\{c_k\} \subset \mathbb{R}_M^m$, for $k = 2, \dots, s$.

Corollary 3.8. Let tol and tol_{sv} be fixed positive constants, and let $\{c_k\} \subset \mathbb{R}_M^m$, for $k = 1, \dots, s$, be the columns of a matrix U . For $k = 1$, assume the SVD $\tilde{U}_1 = \tilde{V}_1 \tilde{\Sigma}_1 \tilde{W}_1^T$ and error bound $e_1 = 0$ are initialized exactly as described in Section 2. For $k = 1, \dots, s - 1$, let $\tilde{U}_{k+1} = \tilde{V}_{k+1} \tilde{\Sigma}_{k+1} \tilde{W}_{k+1}^T$ and e_{k+1} be the output of Algorithm 1 applied to the input $\tilde{U}_k = \tilde{V}_k \tilde{\Sigma}_k \tilde{W}_k^T$ and e_k . If T_p represents the total number of times p truncation is applied and T_{sv} represents the total number of times the singular value truncation is applied, then

$$\|U - \tilde{V}_s \tilde{\Sigma}_s \tilde{W}_s\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}_M^m)} \leq T_p \text{tol} + T_{\text{sv}} \text{tol}_{\text{sv}}.$$

Proof. The proof follows immediately from the previous result, using $p_k \leq \text{tol}$ and $\hat{\sigma}_{r+1} \leq \text{tol}_{\text{sv}}$. \square

The error bound in the result above is not as precise as the error bound computed using Algorithm 1 since the tolerances are only upper bounds on the errors in each step. However, this result does provide some insight into the choice of the tolerances for the algorithm. Specifically, in general there is no reason to expect one of T_p or T_{sv} to be significantly larger than the other; therefore, it seems reasonable to choose equal values for the tolerances. Furthermore, for a very large number of added columns, it is possible that T_p and T_{sv} can be large; therefore, small tolerances should be chosen to preserve accuracy.

Algorithm 1 computes an upper bound on the operator norm error between the exact data matrix U and the approximate truncated SVD $\tilde{U} = \tilde{V} \tilde{\Sigma} \tilde{W}^T$ of the data matrix. (The above corollary also provides another upper bound on the error.) This error bound allows us to bound the error in the incrementally computed singular values and singular vectors. Let $\{\sigma_k, v_k, w_k\}_{k \geq 1}$ and $\{\tilde{\sigma}_k, \tilde{v}_k, \tilde{w}_k\}_{k \geq 1}$ denote the ordered singular values and corresponding orthonormal singular vectors of $U, \tilde{U} : \mathbb{R}^s \rightarrow \mathbb{R}_M^m$ in the result below. The following result follows directly from general results about error bounds for singular values and singular vectors of compact linear operators in Appendix.

Theorem 3.9. Let $k \geq 1$, and let $\varepsilon > 0$ such that $\|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}_M^m)} \leq \varepsilon$. Then

$$|\sigma_\ell - \tilde{\sigma}_\ell| \leq \varepsilon \quad \text{for all } \ell \geq 1.$$

Also, for $j = 1, \dots, k$, define

$$\varepsilon_j = j\varepsilon + 2 \sum_{i=1}^{j-1} (\varepsilon_i + \sigma_i E_i^{1/2}), \quad E_j = 2 \left(1 - \sqrt{\frac{(\sigma_j - 2\varepsilon_j)^2 - \sigma_{j+1}^2}{\sigma_j^2 - \sigma_{j+1}^2}} \right).$$

If the first $k + 1$ singular values of U are distinct and positive, the singular vector pairs $\{\tilde{v}_j, \tilde{w}_j\}_{j=1}^k$ are suitably normalized, and

$$\varepsilon_j \leq \frac{\sigma_j - \sigma_{j+1}}{2} \quad \text{for } j = 1, \dots, k,$$

then

$$\|v_j - \tilde{v}_j\|_M \leq E_j^{1/2}, \quad \|w_j - \tilde{w}_j\| \leq E_j^{1/2} + 2\sigma_j^{-1} \varepsilon_j, \quad \text{for } j = 1, \dots, k. \quad (5)$$

This result indicates we should expect accurate approximate singular values and also accurate approximate singular vectors if ε is small and there is not a small gap in the singular values. We note that POD singular values often decay to zero quickly, and therefore we expect to see lower accuracy in the computed POD modes for smaller singular values due to the small gap. The examples in our first work [45] and the new examples below show both of these expected behaviors for the errors in the approximate singular vectors.

4. Numerical results

We consider a nerve impulse transmission model, the 1D FitzHugh–Nagumo system

$$\begin{aligned} \frac{\partial v(t, x)}{\partial t} &= \mu \frac{\partial^2 v(t, x)}{\partial x^2} - \frac{1}{\mu} w(t, x) + \frac{1}{\mu} f(v) + \frac{c}{\mu}, \quad 0 < x < 1, \\ \frac{\partial w(t, x)}{\partial t} &= bv(t, x) - \gamma w(t, x) + c, \quad 0 < x < 1, \end{aligned}$$

where $f(v) = v(v-0.1)(1-v)$, $\mu = 0.015$, $b = 0.5$, $\gamma = 2$, $c = 0.05$, the boundary conditions are $v_x(t, 0) = -50000t^3 e^{-15t}$, $v_x(t, 1) = 0$, and the initial conditions are zero. This example problem was considered in [53], and we used the interpolated coefficient finite element method from that work to discretize the problem in space. For the finite element method we used continuous piecewise linear basis functions with equally spaced nodes, and we used Matlab's ode23s to approximate the solution of the resulting nonlinear ODE system on different time intervals.

Table 1
Example 1 – error between true and incremental SVD.

tol	tol _{sv}	Rank	Exact error	Incr. error bound
10 ⁻⁸	10 ⁻⁸	36	3.6924e-07	2.8029e-06
10 ⁻⁸	10 ⁻¹⁰	66	3.1932e-07	1.1826e-06
10 ⁻⁸	10 ⁻¹²	61	8.5938e-07	9.0495e-07
10 ⁻¹⁰	10 ⁻⁸	30	3.9090e-08	1.4908e-06
10 ⁻¹⁰	10 ⁻¹⁰	44	4.4893e-10	2.7417e-08
10 ⁻¹⁰	10 ⁻¹²	71	3.9349e-10	8.9680e-09
10 ⁻¹²	10 ⁻⁸	30	3.9090e-08	1.4908e-06
10 ⁻¹²	10 ⁻¹⁰	41	4.5256e-10	1.5511e-08
10 ⁻¹²	10 ⁻¹²	55	4.4334e-12	2.8596e-10

Table 2
Example 2 – error between true and incremental SVD.

tol	tol _{sv}	Rank	Exact error	Incr. error bound
10 ⁻⁸	10 ⁻⁸	35	3.0859e-07	3.6931e-06
10 ⁻⁸	10 ⁻¹⁰	66	1.3881e-07	1.1429e-06
10 ⁻⁸	10 ⁻¹²	64	3.4657e-07	1.5321e-06
10 ⁻¹⁰	10 ⁻⁸	31	4.1497e-08	1.7368e-06
10 ⁻¹⁰	10 ⁻¹⁰	45	5.3142e-10	3.6491e-08
10 ⁻¹⁰	10 ⁻¹²	74	7.7348e-10	1.1523e-08
10 ⁻¹²	10 ⁻⁸	30	4.1497e-08	1.7368e-06
10 ⁻¹²	10 ⁻¹⁰	41	4.6086e-10	1.8671e-08
10 ⁻¹²	10 ⁻¹²	59	4.8658e-12	3.4880e-10

We emphasize that the proposed incremental POD algorithm can be applied to simulation data arising from many types of PDE approximation schemes; see [45] for more discussion of this point. For our numerical tests, we chose the interpolated coefficient finite element method from [53] since it is simple to implement, computationally efficient, and second order accurate in space. We chose Matlab’s second order accurate time stepping method ode23s so that we could consider variable time steps and also accurately and efficiently approximate the solution over longer time intervals. It would be interesting to compare the output of the incremental POD algorithm on simulation data sets arising from different discretizations of the same PDE; we leave this to be explored elsewhere.

For the POD computations, we consider the data $z(t, x) = [v(t, x), w(t, x)]$ in the Hilbert space $L^2(0, 1) \times L^2(0, 1)$ with standard inner product. Now we follow the procedure in our first work [45] to arrive at the weighted SVD problem. At each time step, we rescale the approximate solution data by the square root of the time step; see [45, Section 5.1]. We expand the approximate solution in the finite element basis to obtain the weight matrix M as in [45, Section 5.2]. To compute the POD of the approximate solution data, we compute the SVD of the finite element solution coefficient matrix $U : \mathbb{R}^s \rightarrow \mathbb{R}_M^m$, where s is the number of time steps (snapshots) and m is two times the number of finite element nodes.

To illustrate our analysis of the incremental SVD algorithm, we consider three examples:

Example 1 5000 finite element nodes and $s = 491$ snapshots in the time interval $[0, 10]$

Example 2 10 000 finite element nodes and $s = 710$ snapshots in the time interval $[0, 15]$

Example 3 50 000 finite element nodes and $s = 1275$ snapshots in the time interval $[0, 28]$

We consider relatively small values of $m = 2 \times \text{nodes}$ and s in order to test the incremental algorithm against exact SVD computations.

Let U denote the finite element solution coefficient matrix, and let $\tilde{U} = \tilde{V}\tilde{\Sigma}\tilde{W}^T$ denote the incrementally computed approximate SVD of $U : \mathbb{R}^s \rightarrow \mathbb{R}_M^m$ produced by Algorithm 1. For each example, we choose various tolerances and compute:

$$\text{Rank} = \text{rank}(\tilde{U}), \quad \text{Exact error} = \|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^s, \mathbb{R}_M^m)},$$

$$\text{Incr. error bound} = e \text{ computed by Algorithm 1 at the final snapshot.}$$

The exact SVD of $U : \mathbb{R}^s \rightarrow \mathbb{R}_M^m$ and the exact error are both computed using a Cholesky factorization of the weight matrix M following Algorithm 1 in [45]. The exact computations are for testing only since they require storing all of the data.

Tables 1–3 display the computed quantities listed above for the three examples with various choices of the p truncation tolerance, tol , and the singular value truncation tolerance, tol_{sv} . We set each tolerance to 10^{-8} , 10^{-10} , or 10^{-12} , for a total of nine tests for each example. In all of the tests, the incrementally computed error bound is larger than the exact error and the error bound is small. Also, the tests indicate that there is no benefit from choosing one tolerance different than the other.

Fig. 1 shows the exact and incrementally computed POD singular values and also the weighted norm error between the exact and incrementally computed POD modes with tol and tol_{sv} both equal to 10^{-12} . The errors for the POD modes

Table 3
Example 3 – error between true and incremental SVD.

tol	tol _{sv}	Rank	Exact error	Incr. error bound
10 ⁻⁸	10 ⁻⁸	38	6.5705e-08	4.3271e-06
10 ⁻⁸	10 ⁻¹⁰	72	6.8271e-07	1.1523e-06
10 ⁻⁸	10 ⁻¹²	67	3.6916e-07	2.3847e-06
10 ⁻¹⁰	10 ⁻⁸	31	4.7018e-08	2.2388e-06
10 ⁻¹⁰	10 ⁻¹⁰	49	4.8302e-10	4.3655e-08
10 ⁻¹⁰	10 ⁻¹²	78	2.4473e-08	2.6825e-08
10 ⁻¹²	10 ⁻⁸	31	4.7018e-08	2.2388e-06
10 ⁻¹²	10 ⁻¹⁰	41	4.9660e-10	2.5022e-08
10 ⁻¹²	10 ⁻¹²	60	6.3200e-12	5.7438e-10

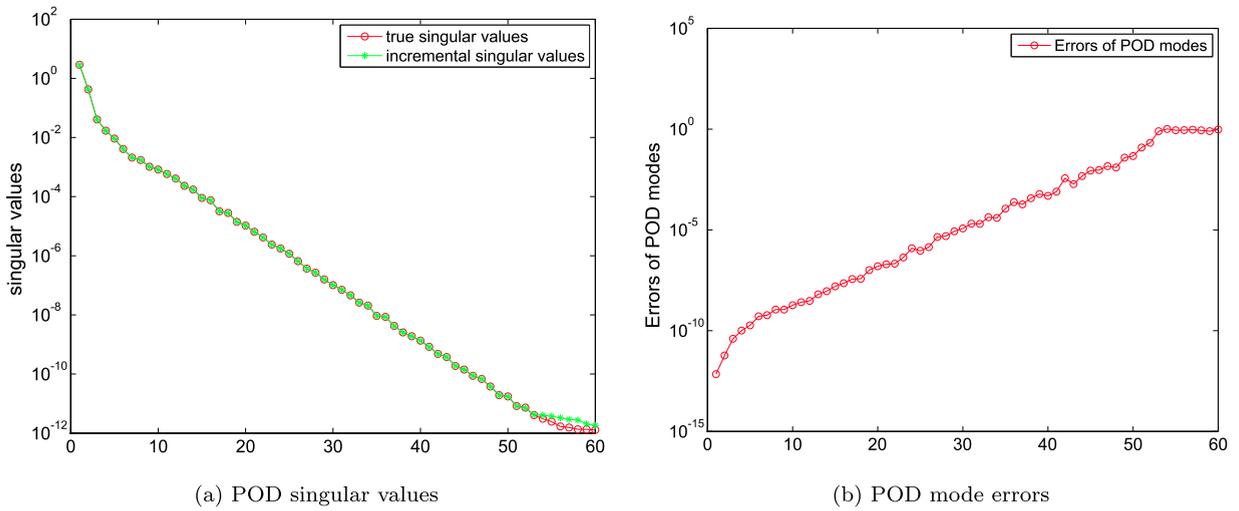


Fig. 1. Example 3 – exact versus incremental POD computations with tol = tol_{sv} = 10⁻¹².

corresponding to the largest singular values are extremely small (approximately 10⁻¹²). The errors in the POD modes increase slowly as the corresponding singular values approach zero. There are many accurate POD modes; the first 30 modes are computed to an accuracy level of at least 10⁻⁵. The POD singular value and mode errors behaved similarly for other cases.

5. Conclusion

In our earlier work [45], we proposed computing the SVD with respect to a weighted inner product incrementally to obtain the POD eigenvalues and modes of a set of PDE simulation data. In this work, we extended the algorithm to update the SVD and an error bound incrementally when a new column is added. We also performed an error analysis of this algorithm by analyzing the error due to each individual truncation. We showed that the algorithm produces the exact SVD of a matrix \tilde{U} such that $\|U - \tilde{U}\|_{\mathcal{L}(\mathbb{R}^S, \mathbb{R}_M^m)} \leq e$, where U is the true data matrix, M is the weight matrix, and e is computed error bound. We also proved error bounds for the incrementally computed singular values and singular vectors. We tested our approach on three example data sets from a 1D FitzHugh–Nagumo PDE system with various choices of the two truncation tolerances. In all of the tests, the incrementally computed error bound was larger than the exact error and the error bound was small. Furthermore, the approximate singular values and dominant singular vectors were accurate. Also, our analysis and the numerical tests suggest that there is no benefit from choosing one algorithm tolerance different than the other.

Acknowledgment

The authors thank Mark Opmeer for a helpful discussion.

Appendix

Let X and Y be two separable Hilbert spaces, with inner products $(\cdot, \cdot)_X$ and $(\cdot, \cdot)_Y$ and corresponding norms $\|\cdot\|_X$ and $\|\cdot\|_Y$. Below, we drop the subscripts on the inner products and the norms since the space will be clear from the context.

Assume $H, H_\varepsilon : X \rightarrow Y$ are compact linear operators. In this section, we prove bounds on the error between the singular vectors of H and H_ε assuming the singular values are distinct. Our results rely on techniques from [54,55].

Let $\{\sigma_k, v_k, w_k\}_{k \geq 1}$ and $\{\sigma_k^\varepsilon, v_k^\varepsilon, w_k^\varepsilon\}_{k \geq 1}$ be the ordered singular values and corresponding orthonormal singular vectors of H and H_ε . They satisfy

$$Hv_k = \sigma_k w_k, \quad H^* w_k = \sigma_k v_k, \quad H_\varepsilon v_k^\varepsilon = \sigma_k^\varepsilon w_k^\varepsilon, \quad H_\varepsilon^* w_k^\varepsilon = \sigma_k^\varepsilon v_k^\varepsilon, \tag{6}$$

where the star denotes the Hilbert adjoint operator. Also, if $\sigma_k > 0$, then σ_k^2 is the k th ordered eigenvalue of the self-adjoint nonnegative compact operators HH^* and H^*H . First, we recall a well-known bound on the singular values; see, e.g., [56, page 30] and [50, page 99].

Proposition A.1. *Let $\varepsilon > 0$ such that $\|H - H_\varepsilon\|_{\mathcal{L}(X,Y)} \leq \varepsilon$. Then for all $k \geq 1$ we have*

$$|\sigma_k - \sigma_k^\varepsilon| < \varepsilon. \tag{7}$$

In the results below, we require the singular vectors $\{v_k^\varepsilon, w_k^\varepsilon\}$ are suitably normalized. We note that any pair $\{v_k^\varepsilon, w_k^\varepsilon\}$ of singular vectors for a fixed value of k can be rescaled by a constant of unit magnitude and remain a pair of singular vectors. However, due to the relationship (6), we note that both vectors in the pair must be rescaled by the same constant.

The proof of the following result is largely contained in [54, Appendix 2], but we include the proof here to be complete.

Lemma A.2. *Let $\varepsilon > 0$ such that $\|H - H_\varepsilon\|_{\mathcal{L}(X,Y)} \leq \varepsilon$. If $\sigma_1 > \sigma_2 > 0$, v_1^ε and w_1^ε are suitably normalized, and*

$$\varepsilon \leq \frac{\sigma_1 - \sigma_2}{2}, \tag{8}$$

then

$$\begin{aligned} \|v_1 - v_1^\varepsilon\| &\leq E_1^{1/2}, \quad \|w_1 - w_1^\varepsilon\| \leq E_1^{1/2} + 2\sigma_1^{-1}\varepsilon, \\ E_1 &= 2 \left(1 - \sqrt{\frac{(\sigma_1 - 2\varepsilon)^2 - \sigma_2^2}{\sigma_1^2 - \sigma_2^2}} \right). \end{aligned} \tag{9}$$

Remark A.3. The larger error bound for $\|w_1 - w_1^\varepsilon\|$ is due to the way we assume the singular vectors are normalized in the proof. It is possible to use a different normalization and make the error bound larger for $\|v_1 - v_1^\varepsilon\|$ instead. We comment on the normalization in the proof.

Proof. Define $V_1 = \text{span}\{v_1\} \subset X$. We have $X = V_1 \oplus V_1^\perp$, and therefore $v_1^\varepsilon = r_\varepsilon v_1 + x_\varepsilon$ for some constant r_ε and $x_\varepsilon \in X$ satisfies $(x_\varepsilon, v_1) = 0$. This gives $\|x_\varepsilon\|^2 = 1 - |r_\varepsilon|^2$ and also $|r_\varepsilon| \leq 1$. Then

$$\begin{aligned} \|v_1 - v_1^\varepsilon\|^2 &= \|v_1 - r_\varepsilon v_1 - x_\varepsilon\|^2 \\ &= |1 - r_\varepsilon|^2 \|v_1\|^2 + \|x_\varepsilon\|^2 \\ &= 2(1 - \text{Re}(r_\varepsilon)). \end{aligned} \tag{10}$$

Note $\|\sigma_1^\varepsilon w_1^\varepsilon\| = \|H_\varepsilon v_1^\varepsilon\|$ implies

$$\begin{aligned} \sigma_1^\varepsilon &= \|H_\varepsilon v_1^\varepsilon + H v_1^\varepsilon - H v_1^\varepsilon\| \\ &\leq \|H v_1^\varepsilon\| + \|H - H_\varepsilon\| \|v_1^\varepsilon\| \\ &\leq \|H(r_\varepsilon v_1 + x_\varepsilon)\| + \varepsilon \\ &= \|r_\varepsilon \sigma_1 w_1 + H x_\varepsilon\| + \varepsilon. \end{aligned}$$

To estimate this norm, we use $(Hx_\varepsilon, w_1) = (x_\varepsilon, H^* w_1) = \sigma_1(x_\varepsilon, v_1) = 0$ and also

$$\|Hx_\varepsilon\|^2 = \frac{(H^* H x_\varepsilon, x_\varepsilon)}{\|x_\varepsilon\|^2} \|x_\varepsilon\|^2 \leq \sup_{x \in V_1^\perp, x \neq 0} \frac{(H^* H x, x)}{\|x\|^2} \|x_\varepsilon\|^2 = \sigma_2^2 \|x_\varepsilon\|^2,$$

where we used the variational characterization of the second eigenvalue σ_2^2 of the self-adjoint compact nonnegative operator H^*H [51, Chapter 28]. These results give

$$\begin{aligned} \|r_\varepsilon \sigma_1 w_1 + Hx_\varepsilon\|^2 &= |r_\varepsilon|^2 \sigma_1^2 + \|Hx_\varepsilon\|^2 \\ &\leq |r_\varepsilon|^2 \sigma_1^2 + \sigma_2^2 \|x_\varepsilon\|^2 \\ &= (\sigma_1^2 - \sigma_2^2) |r_\varepsilon|^2 + \sigma_2^2. \end{aligned}$$

Next, the assumption (8) for ε gives $\varepsilon \leq (\sigma_1 - \sigma_2)/2 \leq \sigma_1/2$, and therefore $\sigma_1 - 2\varepsilon \geq 0$. Also, (7) gives $-\varepsilon \leq \sigma_1^\varepsilon - \sigma_1$, or $\sigma_1^\varepsilon - \varepsilon \geq \sigma_1 - 2\varepsilon \geq 0$. This gives $(\sigma_1^\varepsilon - \varepsilon)^2 \geq (\sigma_1 - 2\varepsilon)^2$, and therefore

$$|r_\varepsilon|^2 \geq \frac{(\sigma_1^\varepsilon - \varepsilon)^2 - \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \geq \frac{(\sigma_1 - 2\varepsilon)^2 - \sigma_2^2}{\sigma_1^2 - \sigma_2^2}.$$

Note that the assumption (8) for ε guarantees that we can take a square root of this estimate.

If v_1^ε is normalized so that r_ε is a nonnegative real number, then (10), $1 - \text{Re}(r_\varepsilon) = 1 - |r_\varepsilon|$, and the above inequality give the desired estimate (9) for $\|v_1 - v_1^\varepsilon\|$. If r_ε is not a nonnegative real number, then rescale the singular vector pair $\{v_1^\varepsilon, w_1^\varepsilon\}$ by $\bar{r}_\varepsilon/|r_\varepsilon|$ to obtain the proper normalization and the bound (9) for $\|v_1 - v_1^\varepsilon\|$.

For w_1 and w_1^ε , it does not appear that we can use a similar proof strategy since we have already rescaled the singular vector pair $\{v_1^\varepsilon, w_1^\varepsilon\}$. Specifically, we can obtain $w_1^\varepsilon = s_\varepsilon w_1 + y_\varepsilon$, but it is not clear that s_ε will be a nonnegative real number and we are unable to rescale again. Therefore, we use $\|H\| = \sigma_1$, $\|H - H_\varepsilon\| \leq \varepsilon$, and $|\sigma_1 - \sigma_1^\varepsilon| \leq \varepsilon$ to directly estimate:

$$\begin{aligned} \|w_1 - w_1^\varepsilon\| &= \|\sigma_1^{-1}Hv_1 - (\sigma_1^\varepsilon)^{-1}H_\varepsilon v_1^\varepsilon\| \\ &\leq \|\sigma_1^{-1}Hv_1 - \sigma_1^{-1}Hv_1^\varepsilon\| + \|\sigma_1^{-1}Hv_1^\varepsilon - \sigma_1^{-1}H_\varepsilon v_1^\varepsilon\| \\ &\quad + \|\sigma_1^{-1}H_\varepsilon v_1^\varepsilon - (\sigma_1^\varepsilon)^{-1}H_\varepsilon v_1^\varepsilon\| \\ &\leq \|v_1 - v_1^\varepsilon\| + \sigma_1^{-1}\varepsilon + |\sigma_1^\varepsilon \sigma_1^{-1} - 1| \\ &\leq \|v_1 - v_1^\varepsilon\| + 2\sigma_1^{-1}\varepsilon. \quad \square \end{aligned}$$

In the result below, note that $\varepsilon_1 = \varepsilon$ and E_1 is defined as in (9) in Lemma A.2.

Theorem A.4. Let $k \geq 1$, and let $\varepsilon > 0$ such that $\|H - H_\varepsilon\|_{\mathcal{L}(X,Y)} \leq \varepsilon$. For $j = 1, \dots, k$, define

$$\varepsilon_j = j\varepsilon + 2 \sum_{i=1}^{j-1} (\varepsilon_i + \sigma_i E_i^{1/2}), \quad E_j = 2 \left(1 - \sqrt{\frac{(\sigma_j - 2\varepsilon_j)^2 - \sigma_{j+1}^2}{\sigma_j^2 - \sigma_{j+1}^2}} \right).$$

If the first $k + 1$ singular values of H are distinct and positive, the singular vector pairs $\{v_j^\varepsilon, w_j^\varepsilon\}_{j=1}^k$ are suitably normalized, and

$$\varepsilon_j \leq \frac{\sigma_j - \sigma_{j+1}}{2} \quad \text{for } j = 1, \dots, k,$$

then

$$\|v_j - v_j^\varepsilon\| \leq E_j^{1/2}, \quad \|w_j - w_j^\varepsilon\| \leq E_j^{1/2} + 2\sigma_j^{-1}\varepsilon_j, \quad \text{for } j = 1, \dots, k. \tag{11}$$

Proof. The proof is by induction. First, the result is true for $k = 1$ by Lemma A.2. Next, assume the result is true for all $j = 1, \dots, k - 1$. Define compact linear operators for $j = 2, \dots, k$ by

$$H^j x = Hx - \sum_{i=1}^{j-1} \sigma_i(x, v_i)w_i, \quad H_\varepsilon^j x = H_\varepsilon x - \sum_{i=1}^{j-1} \sigma_i^\varepsilon(x, v_i^\varepsilon)w_i^\varepsilon,$$

for all $x \in X$. Then the ordered singular values and corresponding singular vectors of H^j and H_ε^j are $\{\sigma_i, v_i, w_i\}_{i \geq j}$ and $\{\sigma_i^\varepsilon, v_i^\varepsilon, w_i^\varepsilon\}_{i \geq j}$.

Note that

$$\begin{aligned} \|H^k x - H_\varepsilon^k x\| &\leq \|(H - H_\varepsilon)x\| + \sum_{i=1}^{k-1} \|\sigma_i^\varepsilon(x, v_i^\varepsilon)w_i^\varepsilon - \sigma_i(x, v_i)w_i\| \\ &\leq \varepsilon\|x\| + \|x\| \sum_{i=1}^{k-1} (|\sigma_i^\varepsilon - \sigma_i| + \sigma_i\|v_i^\varepsilon - v_i\| + \sigma_i\|w_i^\varepsilon - w_i\|). \end{aligned}$$

Then since the result (11) is true for all $j = 1, \dots, k - 1$, we have $\|H^k - H_\varepsilon^k\| \leq \varepsilon_k$, where

$$\begin{aligned} \varepsilon_k &= \varepsilon + \sum_{i=1}^{k-1} (\varepsilon + \sigma_i E_i^{1/2} + \sigma_i(E_i^{1/2} + 2\sigma_i^{-1}\varepsilon_i)) \\ &= k\varepsilon + 2 \sum_{i=1}^{k-1} (\varepsilon_i + \sigma_i E_i^{1/2}). \end{aligned}$$

Applying Lemma A.2 to H^k and H_ε^k with $\|H^k - H_\varepsilon^k\| \leq \varepsilon_k$ completes the proof. \square

References

- [1] T. Colonius, J. Freund, POD analysis of sound generation by a turbulent jet, in: 40th AIAA Aerospace Sciences Meeting & Exhibit, 2002, p. 72.
- [2] R. Zimmermann, S. Görtz, Non-linear reduced order models for steady aerodynamics, *Procedia Comput. Sci.* 1 (1) (2010) 165–174, URL <https://doi.org/10.1016/j.procs.2010.04.019>.
- [3] R. Zimmermann, A locally parametrized reduced-order model for the linear frequency domain approach to time-accurate computational fluid dynamics, *SIAM J. Sci. Comput.* 36 (3) (2014) B508–B537, URL <https://doi.org/10.1137/130942462>.
- [4] L. Peng, K. Mohseni, Nonlinear model reduction via a locally weighted POD method, *Int. J. Numer. Methods Eng.* 106 (5) (2016) 372–396, URL <https://doi.org/10.1002/nme.5124>.
- [5] E.A. Christensen, M. Brøns, J.N. Sørensen, Evaluation of proper orthogonal decomposition-based decomposition techniques applied to parameter-dependent nonturbulent flows, *SIAM J. Sci. Comput.* 21 (4) (1999/00) 1419–1434, URL <https://doi.org/10.1137/S1064827598333181>.
- [6] I. Kalashnikova, M.F. Barone, S. Arunajatesan, B.G. van Bloemen Waanders, Construction of energy-stable projection-based reduced order models, *Appl. Math. Comput.* 249 (2014) 569–596, URL <https://doi.org/10.1016/j.amc.2014.10.073>.
- [7] D. Amsallem, J. Nordström, Energy stable model reduction of neurons by nonnegative discrete empirical interpolation, *SIAM J. Sci. Comput.* 38 (2) (2016) B297–B326, URL <https://doi.org/10.1137/15M1013870>.
- [8] D.N. Daescu, I.M. Navon, A dual-weighted approach to order reduction in 4DVAR data assimilation, *Mon. Weather Rev.* 136 (3) (2008) 1026–1041, URL <https://doi.org/10.1175/2007MWR2102.1>.
- [9] P. Holmes, J.L. Lumley, G. Berkooz, C.W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, second ed., Cambridge Monographs on Mechanics, Cambridge University Press, Cambridge, 2012, p. xvi+386, URL <https://doi.org/10.1017/CBO9780511919701>.
- [10] M.F. Barone, I. Kalashnikova, D.J. Segalman, H.K. Thornquist, Stable Galerkin reduced order models for linearized compressible flow, *J. Comput. Phys.* 228 (6) (2009) 1932–1946, URL <https://doi.org/10.1016/j.jcp.2008.11.015>.
- [11] V.M. Calo, Y. Efendiev, J. Galvis, M. Ghommem, Multiscale empirical interpolation for solving nonlinear PDEs, *J. Comput. Phys.* 278 (2014) 204–220, URL <https://doi.org/10.1016/j.jcp.2014.07.052>.
- [12] C. Farhat, T. Chapman, P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, *Int. J. Numer. Methods Eng.* 102 (5) (2015) 1077–1110, URL <https://doi.org/10.1002/nme.4820>.
- [13] X. Xie, D. Wells, Z. Wang, T. Iliescu, Numerical analysis of the Leray reduced order model, *J. Comput. Appl. Math.* 328 (2018) 12–29, URL <https://doi.org/10.1016/j.cam.2017.06.026>.
- [14] M. Mohebujjaman, L.G. Rebholz, X. Xie, T. Iliescu, Energy balance and mass conservation in reduced order models of fluid flows, *J. Comput. Phys.* 346 (2017) 262–277, URL <https://doi.org/10.1016/j.jcp.2017.06.019>.
- [15] M. Gunzburger, N. Jiang, M. Schneier, An ensemble-proper orthogonal decomposition method for the nonstationary Navier-Stokes equations, *SIAM J. Numer. Anal.* 55 (1) (2017) 286–304, URL <https://doi.org/10.1137/16M1056444>.
- [16] T. Kostova-Vassilevska, G.M. Oxberry, Model reduction of dynamical systems by proper orthogonal decomposition: error bounds and comparison of methods using snapshots from the solution and the time derivatives, *J. Comput. Appl. Math.* 330 (2018) 553–573, URL <https://doi.org/10.1016/j.cam.2017.09.001>.
- [17] A. Quarteroni, A. Manzoni, F. Negri, *Reduced Basis Methods for Partial Differential Equations*, Springer, Cham, 2016, URL <https://doi.org/10.1007/978-3-319-15431-2>.
- [18] J.R. Singler, New POD error expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs, *SIAM J. Numer. Anal.* 52 (2) (2014) 852–876, URL <https://doi.org/10.1137/120886947>.
- [19] K. Kunisch, S. Volkwein, Galerkin Proper orthogonal decomposition methods for a general equation in fluid dynamics, *SIAM J. Numer. Anal.* 40 (2) (2002) 492–515, URL <https://doi.org/10.1137/S0036142900382612>.
- [20] M. Gubisch, S. Volkwein, Proper orthogonal decomposition for linear-quadratic optimal control, in: *Model Reduction and Approximation*, in: *Comput. Sci. Eng.*, vol. 15, SIAM, Philadelphia, PA, 2017, pp. 3–63, URL <https://doi.org/10.1137/1.9781611974829.ch1>.
- [21] Z. Drmač, A.K. Saibaba, The discrete empirical interpolation method: canonical structure and formulation in weighted inner product spaces, *SIAM J. Matrix Anal. Appl.* 39 (3) (2018) 1152–1180, URL <https://doi.org/10.1137/17M1129635>.
- [22] M. Tabandeh, M. Wei, J.P. Collins, On the Symmetrization in POD-Galerkin Model for Linearized Compressible Flows, in: 54th AIAA Aerospace Sciences Meeting, 2016, p. 1106, URL <https://doi.org/10.2514/6.2016-1106>.
- [23] G. Serre, P. Lafon, X. Gloerfelt, C. Bailly, Reliable reduced-order models for time-dependent linearized Euler equations, *J. Comput. Phys.* 231 (15) (2012) 5176–5194, URL <https://doi.org/10.1016/j.jcp.2012.04.019>.
- [24] D. Amsallem, M.J. Zahr, K. Washabaugh, Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction, *Adv. Comput. Math.* 41 (5) (2015) 1187–1230, URL <https://doi.org/10.1007/s10444-015-9409-0>.
- [25] M. Brand, Incremental singular value decomposition of uncertain data with missing values, in: A. Heyden, G. Sparr, M. Nielsen, P. Johansen (Eds.), *Computer Vision – ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 707–720, URL https://doi.org/10.1007/3-540-47969-4_47.
- [26] M. Brand, Fast low-rank modifications of the thin singular value decomposition, *Linear Algebra Appl.* 415 (1) (2006) 20–30, URL <https://doi.org/10.1016/j.laa.2005.07.021>.
- [27] C.G. Baker, K.A. Gallivan, P. Van Dooren, Low-rank incremental methods for computing dominant singular subspaces, *Linear Algebra Appl.* 436 (8) (2012) 2866–2888, URL <https://doi.org/10.1016/j.laa.2011.07.018>.
- [28] Y. Chahlaoui, K. Gallivan, P. Van Dooren, Recursive calculation of dominant singular subspaces, *SIAM J. Matrix Anal. Appl.* 25 (2) (2003) 445–463, URL <https://doi.org/10.1137/S0895479803374657>.
- [29] M.A. Iwen, B.W. Ong, A distributed and incremental SVD algorithm for agglomerative data analysis on large networks, *SIAM J. Matrix Anal. Appl.* 37 (4) (2016) 1699–1718, URL <https://doi.org/10.1137/16M1058467>.
- [30] N. Mastronardi, M. Van Barel, R. Vandebril, A note on the recursive calculation of dominant singular subspaces, *Numer. Algorithms* 38 (4) (2005) 237–242, URL <https://doi.org/10.1007/s11075-004-4338-x>.
- [31] N. Mastronardi, M. Van Barel, R. Vandebril, A fast algorithm for the recursive calculation of dominant singular subspaces, *J. Comput. Appl. Math.* 218 (2) (2008) 238–246, URL <https://doi.org/10.1016/j.cam.2006.12.032>.
- [32] M. Fahl, Computation of POD basis functions for fluid flows with Lanczos methods, *Math. Comput. Modelling* 34 (1–2) (2001) 91–107, URL [https://doi.org/10.1016/S0895-7177\(01\)00051-6](https://doi.org/10.1016/S0895-7177(01)00051-6).
- [33] C.A. Beattie, J. Borggaard, S. Gugercin, T. Iliescu, A domain decomposition approach to POD, in: *Proceedings of the IEEE Conference on Decision and Control*, 2006, pp. 6750–6756, URL <https://doi.org/10.1109/CDC.2006.377642>.
- [34] Z. Wang, B. McBee, T. Iliescu, Approximate partitioned method of snapshots for POD, *J. Comput. Appl. Math.* 307 (2016) 374–384, URL <https://doi.org/10.1016/j.cam.2015.11.023>.
- [35] C. Himpe, T. Leibner, S. Rave, Hierarchical approximate proper orthogonal decomposition, *SIAM J. Sci. Comput.* 40 (5) (2018) A3267–A3292, URL <https://doi.org/10.1137/16M1085413>.

- [36] A. Placzek, D.-M. Tran, R. Ohayon, A nonlinear POD-Galerkin reduced-order model for compressible flows taking into account rigid body motions, *Comput. Methods Appl. Mech. Engrg.* 200 (49–52) (2011) 3497–3514, URL <https://doi.org/10.1016/j.cma.2011.08.017>.
- [37] A. Corigliano, M. Dossi, S. Mariani, Model order reduction and domain decomposition strategies for the solution of the dynamic elastic-plastic structural problem, *Comput. Methods Appl. Mech. Engrg.* 290 (2015) 127–155, URL <https://doi.org/10.1016/j.cma.2015.02.021>.
- [38] B. Peherstorfer, K. Willcox, Dynamic data-driven reduced-order models, *Comput. Methods Appl. Mech. Engrg.* 291 (2015) 21–41, URL <https://doi.org/10.1016/j.cma.2015.03.018>.
- [39] B. Peherstorfer, K. Willcox, Dynamic data-driven model reduction: adapting reduced models from incomplete data, *Adv. Model. Simul. Eng. Sci.* 3 (1) (2016) 11, URL <https://doi.org/10.1186/s40323-016-0064-x>.
- [40] O.T. Schmidt, A. Towne, An efficient streaming algorithm for spectral proper orthogonal decomposition, *Comput. Phys. Comm.* 237 (2019) 98–109, URL <https://doi.org/10.1016/j.cpc.2018.11.009>.
- [41] M.J. Zahr, C. Farhat, Progressive construction of a parametric reduced-order model for PDE-constrained optimization, *Int. J. Numer. Methods Eng.* 102 (5) (2015) 1111–1135, URL <https://doi.org/10.1002/nme.4770>.
- [42] R. Zimmermann, A geometric note on subspace updates and orthogonal matrix decompositions under rank-one modifications, 2017. arXiv preprint [arXiv:1711.08235](https://arxiv.org/abs/1711.08235).
- [43] R. Zimmermann, B. Peherstorfer, K. Willcox, Geometric subspace updates with applications to online adaptive nonlinear model reduction, *SIAM J. Matrix Anal. Appl.* 39 (1) (2018) 234–261, URL <https://doi.org/10.1137/17M1123286>.
- [44] G.M. Oxberry, T. Kostova-Vassilevska, W. Arrighi, K. Chand, Limited-memory adaptive snapshot selection for proper orthogonal decomposition, *Int. J. Numer. Methods Eng.* 109 (2) (2017) 198–217, URL <https://doi.org/10.1002/nme.5283>.
- [45] H. Fareed, J.R. Singler, Y. Zhang, J. Shen, Incremental proper orthogonal decomposition for PDE simulation data, *Comput. Math. Appl.* 75 (6) (2018) 1942–1960, URL <https://doi.org/10.1016/j.camwa.2017.09.012>.
- [46] L. Giraud, J. Langou, When modified Gram-Schmidt generates a well-conditioned set of vectors, *IMA J. Numer. Anal.* 22 (4) (2002) 521–528, URL <https://doi.org/10.1093/imanum/22.4.521>.
- [47] L. Giraud, J. Langou, M. Rozložník, J. van den Eshof, Rounding error analysis of the classical Gram-Schmidt orthogonalization process, *Numer. Math.* 101 (1) (2005) 87–100, URL <https://doi.org/10.1007/s00211-005-0615-4>.
- [48] L. Giraud, J. Langou, M. Rozložník, The loss of orthogonality in the Gram-Schmidt orthogonalization process, *Comput. Math. Appl.* 50 (7) (2005) 1069–1075, URL <https://doi.org/10.1016/j.camwa.2005.08.009>.
- [49] M. Rozložník, M. Tůma, A. Smoktunowicz, J. Kopal, Numerical stability of orthogonalization methods with a non-standard inner product, *BIT* 52 (4) (2012) 1035–1058, URL <https://doi.org/10.1007/s10543-012-0398-9>.
- [50] I. Gohberg, S. Goldberg, M.A. Kaashoek, *Classes of Linear Operators. Vol. I*, in: *Operator Theory: Advances and Applications*, vol. 49, Birkhäuser Verlag, Basel, 1990, p. xiv+468, URL <https://doi.org/10.1007/978-3-0348-7509-7>.
- [51] P.D. Lax, *Functional Analysis*, in: *Pure and Applied Mathematics (New York)*, Wiley-Interscience [John Wiley & Sons], New York, 2002, p. xx+580.
- [52] M. Reed, B. Simon, *Methods of Modern Mathematical Physics I: Functional Analysis*, second ed., Academic Press, Inc., New York, 1980, p. xv+400.
- [53] Z. Wang, Nonlinear model reduction based on the finite element method with interpolated coefficients: semilinear parabolic equations, *Numer. Methods Partial Differential Equations* 31 (6) (2015) 1713–1741, URL <https://doi.org/10.1002/num.21961>.
- [54] K. Glover, R.F. Curtain, J.R. Partington, Realisation and approximation of linear infinite-dimensional systems with error bounds, *SIAM J. Control Optim.* 26 (4) (1988) 863–898, URL <https://doi.org/10.1137/0326049>.
- [55] P. Galán del Sastre, R. Bermejo, Error estimates of proper orthogonal decomposition eigenvectors and Galerkin projection for a general dynamical system arising in fluid models, *Numer. Math.* 110 (1) (2008) 49–81, URL <https://doi.org/10.1007/s00211-008-0155-9>.
- [56] I.C. Gohberg, M.G. Kreĭn, *Introduction to the Theory of Linear Nonselfadjoint Operators*, in: *Translated from the Russian by A. Feinstein. Translations of Mathematical Monographs*, vol. 18, American Mathematical Society, Providence, RI., 1969, p. xv+378.