Extended dynamic mode decomposition for model reduction in fluid dynamics simulations

Published Version:
Libero, G., Chiofalo, A., Ciriello, V., Tartakovsky, D.M. (2024). Extended dynamic mode decomposition for model reduction in fluid dynamics simulations. PHYSICS OF FLUIDS, 36(6), 1-10 [10.1063/5.0207957].

Availability:
This version is available at: https://hdl.handle.net/11585/975236 since: 2024-07-22

Published:
DOI: http://doi.org/10.1063/5.0207957

Extended dynamic mode decomposition for model reduction in fluid dynamics simulations

Giulia Libero,[1] Alessia Chiofalo,[1] Valentina Ciriello,[1] and Daniel M. Tartakovsky[2, a)]

[1)] *Department of Civil, Chemical, Environmental and Materials Engineering, University of Bologna, via Zamboni, 33–40126 Bologna, Italy*

[2)] *Department of Energy Science and Engineering, Stanford University, 367 Panama St., Stanford, CA 94305, USA*

(Dated: 30 May 2024)

High computational cost and storage/memory requirements of fluid dynamics simulations constrain their usefulness as a predictive tool. Reduced-order models (ROMs) provide a viable solution to this challenge by extracting the key underlying dynamics of a complex system directly from data. We investigate the efficacy and robustness of an extended dynamic mode decomposition (xDMD) algorithm in constructing ROMs of three-dimensional cardiovascular computations. Focusing on the ROMs' accuracy in representation and interpolation, we relate these metrics to the truncation rank of singular value decomposition, which underpins xDMD and other approaches to ROM construction. Our key innovation is to relate the truncation rank to the singular values of the original flow problem. This result establishes a priori guidelines for the xDMD deployment and its likely success as a means of data compression and reconstruction of the system's dynamics from dominant spatiotemporal structures present in the data.

[a)] tartakovsky@stanford.edu

I. INTRODUCTION

High computational burden of fluid dynamics simulations has propelled the development of model reduction techniques for problems dealing with complex flow and transport processes in fields as diverse as geosciences and biomedicine[1–5]. A reduced-order model (ROM) is a computationally efficient and reasonably accurate representation of the underlying dynamics of a state variable or a quantity of interest, derived from observations and/or computer-generated data. The efficiency of a model reduction technique manifests itself in both the amount of data required for the ROM construction and the ROM approximation accuracy in the interpolation and extrapolation regimes[6].

Dynamic mode decomposition (DMD) is a data-driven technique that constructs ROMs of complex dynamical systems by employing the singular value decomposition (SVD)[7,8]. DMD aims to identify spatiotemporal structures that are dominant in the data and to reconstruct an optimal linear model from these structures. A DMD variant xDMD[9] combines salient features of the residual learning[10] and the generalized DMD with a bias term[11]. This DMD algorithm has the ability to handle dynamical systems described by inhomogeneous partial differential equations, which proved to be problematic for standard DMD. Numerical studies, dealing with problems as diverse as the Navier-Stokes equations[9] and multiphase transport in porous media[12], suggest that the xDMD is more accurate than the standard DMD algorithm (hereinafter, sDMD[13]). Since xDMD has more parameters than sDMD (the bias term), it is potentially more sensitive to noise than. However, the numerical experiments[9] indicate that the correction effects from the bias term may dominate the effects of over-fitting the noise.

These and other methods for ROM construction rely on the truncation rank of SVD to control the degree of order reduction and representation accuracy. The choice of how many singular values to keep depends on such factors as the quality and origin of the data and the dynamic importance of low-energy modes[13]. The rank selection is typically done via experimentation, rendering the method's implementation subjective. A more principled approach is to balance order reduction and approximation accuracy by utilizing a general criteria[12]. The rank choice is also linked to xDMD's data compression ability, which is given by the capability of the algorithm to preserve high accuracy for low values of the truncation rank[12,13]. By identifying dominant coherent structures from data, the method effectively reduces the dimensionality of high-dimensional datasets, thereby achieving compression-like effects. That is relevant in fluid dynamics, where DMD operates by reducing the dimensionality of the flow field data while preserving its essential characteristics. Another application is climate science, where DMD can be used to compress large-scale climate datasets into a reduced set of dominant modes, facilitating the analysis and visualization of long-term climate trends and variability[13]. In yet another setting of multiphase flow in porous media[12], xDMD demonstrated high prediction accuracy (relative interpolation error on the order of 10^{-9}) with a truncation rank of up to 35% of the dataset dimension. By way of a disclaimer, we note that, like other SVD-based techniques, DMD often struggles to honor translational and rotational invariances of low-rank objects embedded in the data[13].

Our study has three intertwined goals. The first is to analyze how the representation error of xDMD is affected by the truncation rank in SVD, which, in turn, is linked to singular values of the problem. The second is to test the xDMD-based ROM in terms of its interpolation error, for different truncation ranks. The third goal is to explore the effect of neglecting possible irrelevant/overfit-inducing information (noise) on the accuracy of the approximation. We pursue these goals in the context of three-dimensional (3D) cardiovascular simulations of blood flow in a complex geometry of a patient-specific aorta.

The reference aorta geometry is selected from the Vascular Model Repository (www.vascularmodel.com), a library of patient-specific cardiovascular models developed on volumetric image data sets and relevant physiologic data[14]. Fluid dynamics data are generated with `SimVascular` (http://simvascular.github.io/). The latter is an open-source software that provides a complete pipeline, from medical image data segmentation to patient-specific blood flow simulations based on the 3D incompressible Navier-Stokes equations[15]. We use a data set consisting of $\approx 2 \cdot 10^3$ time frames of the velocity distribution (on a mesh with $\sim 10^5$ elements) in a selected aorta.

Our research provides practical guidelines for the selection of low-rank truncation options for optimal order-reduction (data compression). Our findings suggest that excluding low-energy modes, which do not contribute to the elucidation of system dynamics, is beneficial to ROM accuracy. We also found the ROM accuracy to be robust to both the size of time intervals between the snapshots and low-rank truncations. This conclusion requires a flow map of the system dynamics to be sufficiently smooth in space. An optimal rank selection needs to consider the ROM's prediction reliability not only in reproducing the training data (representation error) but also in making predictions at space-time locations where the data are not available (interpolation error). Once optimized, the ROM can be used to replicate cardiac function in a low-dimensional space, reducing the simulation cost and facilitating the optimization and design of patient-specific interventions. At the same time, the DMD-based modal decomposition allows for the identification of physically interpretable patterns in the temporal and spatial evolution of the observed cardiovascular phenomena[16]. Coherent structures and dominant flow features can be analyzed to discover the underlying physics and possibly employed to detect pathologies[17–21].

The paper is organized as follows: Section II is devoted to the formulation of the problem; in Section III the xDMD algorithm is described; while in Section IV its application to the test case is presented and discussed; a set of final remarks in Section V closes the paper.

II. PROBLEM FORMULATION

Once discretized on a numerical mesh, system states are arranged into a state vector $\mathbf{u}(t) \in \mathbb{R}^N$ of length N. The temporal evolution of this discretized system is described by a system of N (nonlinear, homogeneous) ordinary differential equations,

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}, \mathbf{s}), \qquad \mathbf{u}(0) = \mathbf{u}_0, \tag{1}$$

where $\mathbf{f}(\mathbf{u}, \cdot)$ decribes the nonlinear dynamics, $\mathbf{s} \in \mathbb{R}^N$ represents the source/sinks term and boundary conditions, and $\mathbf{u}_0 \in \mathbb{R}^N$ denotes the discretized initial state of the system.

Let $\mathbf{\Phi}_{\Delta t} : \mathbb{R}^N \to \mathbb{R}^N$ be a flow map, which relates the discretized system state $\mathbf{u}(t)$ to $\mathbf{u}(t + \Delta t)$ at any time t and time step Δt.

Lemma 1 *Assume* $\mathbf{f}$ *is Lipschitz continuous with Lipschitz constant* L *on a set* $\mathcal{H} \subseteq \mathbb{R}^N$*. Define* $\mathcal{H}_{\Delta t} = \{\mathbf{y} \in \mathcal{H} : \mathbf{\Phi}_{\Delta t}(\mathbf{y}) \in \mathcal{H}\}$*. Then, the flow map* $\mathbf{\Phi}_{\Delta t}$ *is Lipschitz continuous on* $\mathcal{H}_{\Delta t}$*. Specifically, for any* $\mathbf{y}$ *and* $\tilde{\mathbf{y}} \in \mathcal{H}_{\Delta t}$*,*

$$\| \mathbf{\Phi}_{\Delta t}(\mathbf{y}; \mathbf{s}) - \mathbf{\Phi}_{\Delta t}(\tilde{\mathbf{y}}; \mathbf{s}) \| \leq e^{L\tau} \| \mathbf{y} - \tilde{\mathbf{y}} \|, \forall \tau \in [t, t + \Delta t] .$$

The proof follows from the classical result on the continuity of a dynamical system (p. 109 in Ref. 22). The local Lipschitz continuity of the flow map ensures that nearby trajectories evolve smoothly and predictably, which is critical for the validity of the DMD approximation[11]. Moreover, if the flow map is locally Lipschitz continuous, the system's behavior can be accurately represented by a finite number of modes that evolve smoothly in time, thus aiding interpretation and forecasting. Various DMD studies[23] indicate that a linear operator might not be a good approximation of the general flow map, particularly for highly nonlinear problems. In such cases, it might be necessary to map the state variables onto observables[13].

The DMD approach approximates the nonlinear dynamical system, i.e., $\mathbf{f}(\mathbf{u}, \cdot)$, with a linear model constructed from M temporal snapshots of the discretized state variable, $\mathbf{x}_k = \mathbf{u}(t_k)$ with $k = 0, \ldots, M-1$. In general, numerical simulations involve discretizing continuous processes into time steps. The continuous nature of the flow map enables interpolation between simulation time steps or extrapolation beyond them, providing a more precise representation of the system's behavior.

Let $\mathcal{L}$ be a DMD-based ROM of the dynamical system (1). At time t_k, the true solution induced by the flow map $\mathbf{\Phi}_{\Delta t}$ and its DMD approximation are

$$\mathbf{x}_k = \mathbf{\Phi}_{\Delta t}(\mathbf{x}_{k-1}) \quad \text{and} \quad \mathbf{x}_k^{\mathcal{L}} = \mathcal{L}(\mathbf{x}_{k-1}^{\mathcal{L}}), \tag{2}$$

respectively. The error of a DMD model at time t_k is

$$\delta_k^{\mathcal{L}} = \|\mathbf{x}_k^{\mathcal{L}} - \mathbf{x}_k\|, \tag{3}$$

where $\|\cdot\|$ denotes vector 2-norm. The error bounds for xDMD and sDMD, reported in Appendix A, provide a general indicator[9] for the growth of $\delta_k^{\mathcal{L}}$. The numerical experiments reported in Section IV serve to investigate this error in detail.

III. THE XDMD ALGORITHM

Consider a set of $(M+1)$ snapshots of the vector of state variables, $\mathbf{x}_k$ with $t_{k+1} = t_k + \Delta t$ and $k = 0, \ldots, M$. Let $\mathbf{X} \in \mathbb{R}^{N \times M}$ denote a matrix whose columns are the vectors $\mathbf{x}_0, \ldots, \mathbf{x}_{M-1}$. Let $\mathbf{X}' \in \mathbb{R}^{N \times M}$ denote a matrix whose columns are the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_M$. The sDMD algorithm describes the temporal evolution of $\mathbf{u}(t)$ with a linear model

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k, \qquad \mathbf{A} = \mathbf{X}'\mathbf{X}^{\dagger} \in \mathbb{R}^{N \times N}. \tag{4}$$

In a typical application, $M \ll N$ so that the rank of $\mathbf{A}$ is at most M. Even though, computing $\mathbf{A}$ (or its spectral decomposition) is generally onerous. Instead, the truncated SVD of $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$, with rank $r < M$, is used[13]:

$$\mathbf{A} \approx \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^{\top}, \tag{5}$$

where $\mathbf{U} \in \mathbb{R}^{N\times r}$, $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$, $\mathbf{V} \in \mathbb{R}^{M\times r}$. If r is smaller than the number of nonzero singular values (i.e., the rank of $\mathbf{X}$), then the truncated SVD is a proxy of $\mathbf{X}$.

To allow for a problem's inhomogeneity, the generalized DMD algorithm adds a bias term $\mathbf{b}_g \in \mathbb{R}^N$ to the standard formulation,

$$\mathbf{x}_{k+1} \approx \mathbf{A}_g \mathbf{x}_k + \mathbf{b}_g. \tag{6}$$

Here, $[\mathbf{A}_g\ \mathbf{b}_g] = \mathbf{X}'\tilde{\mathbf{X}}^\dagger \in \mathbb{R}^{N\times N+1}$, where $\tilde{\mathbf{X}}^\top = [\mathbf{X}\ \mathbf{1}]$ and $\tilde{\mathbf{X}} \in \mathbb{R}^{N+1\times M}$. The computational cost is reduced by obtaining the best-fit linear operator through the SVD of the matrix $\tilde{\mathbf{X}} \approx \mathbf{U}_g \mathbf{\Sigma}_g \mathbf{V}_g^\top$, such that

$$[\mathbf{A}_g\ \mathbf{b}_g] \approx \mathbf{X}'\mathbf{V}_g \mathbf{\Sigma}_g^{-1} \mathbf{U}_g^\top, \tag{7}$$

where $\mathbf{U}_g \in \mathbb{R}^{N+1\times r}$, $\mathbf{\Sigma}_g \in \mathbb{R}^{r\times r}$, and $\mathbf{V}_g \in \mathbb{R}^{M\times r}$. By construction, the error of this gDMD method is equal to or smaller than that of sDMD (Appendix A).

The extended DMD (xDMD) approach[9] endows gDMD with a residual-learning idea. It approximates the relationship between $\mathbf{Y} = \mathbf{X}' - \mathbf{X}$ and $\mathbf{X}$,

$$\mathbf{y}_{k+1} = \mathbf{B}_x \mathbf{x}_k + \mathbf{b}_x. \tag{8}$$

Here, $[\mathbf{B}_x\ \mathbf{b}_x] = \mathbf{Y}\tilde{\mathbf{X}}^\dagger \in \mathbb{R}^{N\times N+1}$, and $\tilde{\mathbf{X}}^\top \in \mathbb{R}^{N+1\times M}$ is defined as before. For computational saving, the best-fit linear operator is obtained through the SVD of the matrix $\tilde{\mathbf{X}}$ as

$$[\mathbf{B}_x\ \mathbf{b}_x] \approx \mathbf{Y}\mathbf{V}_g \mathbf{\Sigma}_g^{-1} \mathbf{U}_g^\top. \tag{9}$$

The error of xDMD equals to or is smaller than that of the residual DMD without bias (Appendix A). The impact of the bias term and residual learning on the accuracy of the DMD method is studied in Ref. 9. An efficient computational strategy to derive prediction in Eq. (8) is presented in Appendix B.

DMD can be used as a ROM of a nonlinear PDE, whose solution is confined in $\mathcal{H} \subseteq \mathbb{R}^N$ (to satisfy the assumptions in Lemma 1). We assess the performance of xDMD, both in representation and interpolation, in terms of the relative error[9,12,13]

$$\varepsilon_{\mathcal{L}}^k = \frac{\|\mathbf{x}_k^{\mathcal{L}} - \mathbf{x}_k\|^2}{\|\mathbf{x}_k\|^2}, \tag{10}$$

where $\|\cdot\|$ denotes vector 2-norm.

Several criteria can be used to select the truncation rank of a ROM[1,12,13]. One is to use the rank of the data matrix, $r = \text{rank}(\tilde{\mathbf{X}})$, i.e., to incorporate all the information contained in the data, including the noise. Another criterion is based on the cumulative energy in the SVD of $\tilde{\mathbf{X}}$; for example, one could set $r = r_{90}$, where

$$r_{90} = \min(n) : \frac{\sum_{k=0}^{n} \sigma_k}{\sum_{k=0}^{M-1} \sigma_k} \geq 0.9 \tag{11}$$

is the number of diagonal elements of $\mathbf{\Sigma}$ that accounts for 90% of the energy. Yet another criterion defines $r = r^*$ as the number of diagonal elements of $\mathbf{\Sigma}$ associated with the first singular value satisfying the inequality

$$r^* = \min(n) : \sigma_n \leq 10^{-5} \sum_{k=0}^{M-1} \sigma_k. \tag{12}$$

The latter two criteria allow one to ascertain the effect of truncation of low-energy modes, as we do below.

IV. APPLICATION

A. 3D Cardiovascular Model

We deploy the `SimVascular` software[15] to solve 3D incompressible Navier-Stokes equations describing blood flow in a patient-specific aorta. A cardiovascular model and the flow-domain geometry are selected, at random, from the Vascular Model Repository[14]; the homogeneous Dirichlet boundary conditions imposed at the aorta walls imply no-slip velocity at the rigid wall[24]. `SimVascular` relies on the 3D Delaunay triangulation to discretize the flow domain with a triangular mesh of $N = 343352$ elements. (The flow-domain geometry and the corresponding mesh are available for download from the Vascular Model Repository.) A typical 3D finite-element simulations of the unsteady Navier–Stokes of two cardiac cycles for this type of geometry takes a few hours[24]. The quantity of interest, arranged in the vector $\mathbf{u} \in \mathbb{R}^N$ (see Section II), is the velocity magnitude of which $M = 1868$ snapshots, $\mathbf{u}(t_k)$, are collected over 7.7 s, which covers about 12 pulsations. Columns of matrices $\mathbf{X}$ and $\mathbf{X}'$ are given by the snapshots of the velocity magnitude computed by `SimVascular` at a constant time interval (see Section III). We chose the number of snapshots to be sufficiently large to perform interpolation tests for different time steps.

The `SimVascular` predictions are used to perform multiple tests, both in representation and interpolation regimes, with datasets of reduced (in space and/or time) size to verify the generality of our results. In the representation regime, these tests start with an analysis of the representation error performed on the entire dataset of $M = 1868$ snapshots, each consisting of $N = 343352$ grid elements. Next, ROMs are trained on randomly selected data sets in which N is reduced by a tenth and a hundredth. Finally, ROMs are trained on randomly selected data sets in which M is reduced to 200 snapshots associated with different time intervals. In the interpolation-error analysis, we perform several tests for different interpolation rates. Results and analysis of these tests are presented in the following section.

B. Results and Discussion

1. *Representation Error and Data Compression*

We use xDMD to construct ROMs from the entire collection of snapshots of the velocity magnitude and testing these ROMs' ability to reproduce these training data. This exercise quantifies the representation error of xDMD. A sequence of ROMs differ from each other in the truncation rank applied to the SVD. We explore the xDMD accuracy at low-rank truncations, which are relied upon to identify dominant spatiotemporal structures in the computer-generated data. This analysis is also relevant for the exploration of xDMD's effectiveness for data compression and storage.

Figure 1a shows the ROMs' representation error, computed with Eq. (10) for different truncation ranks r and averaged over all the time steps. As expected, the representation error decreases with the truncation rank r. High accuracy is reached for relatively low r: when $r = \text{rank}(\tilde{X}) = 1868$, i.e., in the absence of truncation, the representation error is $3.6 \cdot 10^{-16}$; setting $r = r^* = 357$ or $r = r_{90} = 24$ leads to errors of $2.8 \cdot 10^{-5}$ or $1.5 \cdot 10^{-1}$, respectively. By considering only 20% of the modes, with $r = r^*$, the result is remarkably accurate. Additionally, the cumulative energy associated with r^* is approximately equal to 1 (Fig. 1c). That is linked to the rate at which the singular values decrease to 0 (Fig. 1b), indicating that the limited number of modes captured by r^* are dominant in the dynamics. The remaining features ($n > 357$) are low-energy modes that do not affect the ROM accuracy; as such, they can be interpreted as noise and, for the purpose of data compression, can be neglected.

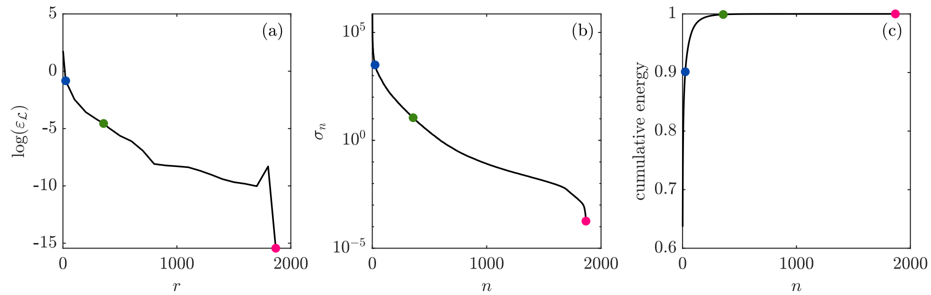

FIG. 1. (a) Representation error (averaged over the time instants) as function of the truncation rank r of the SVD of $\tilde{\mathbf{X}}$ when all data ($N = 343352$, $M = 1868$) are used to train the ROMs. (b) Singular values and (c) cumulative energy associated with the SVD, both plotted as function of the singular values number n. In all panels, the blue, green, and red dots correspond to $r = r_{90}$, $r = r^*$, and $r = \text{rank}(\tilde{\mathbf{X}})$, respectively. In this example, $r_{90} = 24$, $r^* = 357$, and $\text{rank}(\tilde{\mathbf{X}}) = M = 1868$ resulting in no truncation.

To elucidate further the effects of the truncation rank on the prediction accuracy of xDMD, we compare the original data with the corresponding reconstructed snapshots provided by the ROMs truncated at r_{90} and r^* (Fig. 2). Both ROMs reproduce the general velocity patterns, although the r_{90} truncation rank returns a slightly worse approximation. This comparison demonstrates the ROM ability to capture the salient features of the flow, which suggests that xDMD is suitable for the interpretation and reproduction of 3D cardiovascular simulations. Depending on the accuracy required by the application, one can select an appropriate truncation criteria and employ the xDMD-based ROM to replace the onerous numerical simulations with compressed reconstructions.

To test the method's robustness, we train ROMs on data sets with missing spatial data. Specifically, elements of the original mesh of size N are randomly selected to obtain two reduced-size data sets of dimensions $N/10$ and $N/100$. Representation accuracy of the resulting ROMs, trained on all $M = 1868$ temporal snapshots, is shown in Figure 3a, for the same values of $r = r_{90}$, $r = r^*$, and $r = \text{rank}(\tilde{\mathbf{X}})$. When only the dominant spatiotemporal structures of the underlying flow are considered, the accuracy close to locations where the training data are sampled is not affected by the data loss. The error increases with r,

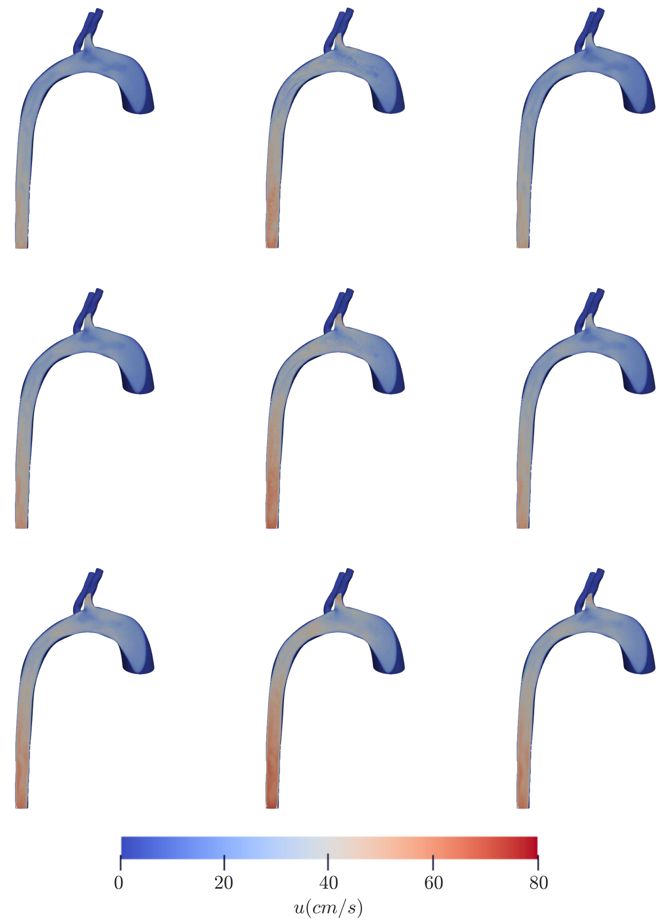

FIG. 2. Magnitude of the flow velocity u in the aorta, as predicted by (left column) direct numerical simulations, (middle column) xDMD with truncation ranks r_{90}, and (right column) xDMD with r^*. The velocity is plotted at times $k = M/3$, $k = 2M/3$ and $k = M$ in the first, second and third rows, respectively.

reaching tens of orders of magnitude for $r = \text{rank}(\tilde{\mathbf{X}})$ when all the features contained in the data are accounted for. This finding suggests that when the data are not sufficiently rich to cover the solution space of interest, considering low-energy modes does not increase the ROM accuracy.

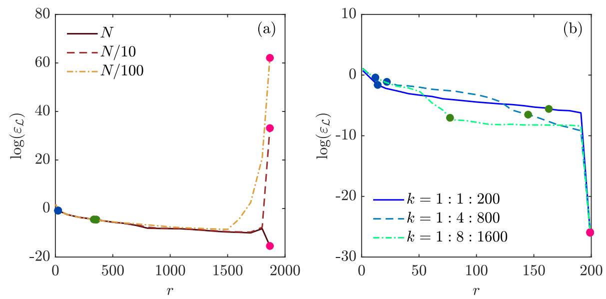

FIG. 3. Dependence of time-averaged representation error of ROMs on the SVD truncation rank r. In (a), the ROMs are alternatively trained on the data in all N pixels and on the data at randomly selected $N/10$ and $N/100$ pixels; in all three cases, using M snapshots. In (b), the ROMs are alternatively trained on the first 200 snapshots and on the 200 snapshots selected with time intervals 4 or 8; in all three cases, using $N/100$ pixels. The blue, green and red dots correspond to $r = r_{90}$, $r = r^*$ and $r = \text{rank}(\tilde{\mathbf{X}})$, respectively.

Another facet of xDMD's robustness is its sensitivity to the number of temporal snapshots available for training. Figure 3b shows the representation error of the xDMD trained on $N/100$ velocity measurements and 200 snapshots. (These snapshots are selected from the full data set ($M = 1868$) using either the first 200 images or every fourth or every eighth image.) This experiment reveals that the ROM's accuracy is not affected by either the reduction of the number of snapshots or the time step between the snapshots. Hence, xDMD is robust and provides a good approximation of nonlinear flow phenomena.

2. *Interpolation Error*

ROMs are typically employed to make predictions at space-time points wherein the output of fluid dynamic simulations is not available. We test our ROMs' performance in the interpolation regime for several values of the interpolation rate η. The data-matrix dimensions and truncation ranks for all the cases considered are reported in Table I.

We start by constructing three ROMs associated with the truncation rank $r = \text{rank}(\tilde{\mathbf{X}})$, $r = r^*$ and $r = r_{90}$, and trained on half of the snapshots, i.e., $\eta = 0.5$ (Case 1 in Table I).

TABLE I. Cases considered for interpolation tests.

Case	η	Train set	$r = r_{90}$	$r = r^*$	rank($\tilde{\mathbf{X}}$)
1	0.5	$k = 1 : 2 : M$	23	252	934
2	0.67	$k = 1 : 3 : M$	23	261	622
3	0.8	$k = 1 : 5 : M$	16	119	373
4	0.9	$k = 1 : 10 : M$	10	62	186

Interpolation errors of these ROMs are shown in Figure 4a; the errors are defined in Eq. (10) and predictions are carried out for the missing half of time steps. The ROM truncated at $r = r^*$ assures high accuracy and stability (the error varies between 10^{-5} and 10^{-4} at all times), while the truncation at $r = \mathrm{rank}(\tilde{\mathbf{X}})$ results in the error that increases with time; if $r = r_{90}$ the error is stable in time but about three orders of magnitude higher than in the case of $r = r^*$ (it varies between 10^{-2} and $10^{-0.5}$). Reducing the size of the training set, i.e., setting $\eta = 0.67$ (Case 2 in Table I), yields the two different ROMs truncated at r^* and rank($\tilde{\mathbf{X}}$) with similar interpolation errors, while $r = r_{90}$ produces a significantly higher error (Figure 4b); for all r considered, the respective ROMs' error peaks are aligned and the periodicity is similar, with $r = r^*$ providing a smaller error. In the cases of $\eta = 0.8$ and $\eta = 0.9$ (Cases 3 and 4 in Table I, respectively) the interpolation errors of all the ROMs increase with time and the difference when truncating at r^* and rank($\tilde{\mathbf{X}}$) relative to $r = r_{90}$ decreases till about one order of magnitude in the case of $\eta = 0.9$ (Figure 4c-d).

To provide a local view on the ROMs' accuracy, Figure 5 compares the reference and reconstructed velocity time series at two points in a cross-section of the aorta for Cases 3 and 4 in Table I in panels (a) and (c) and (b) and (d), respectively. As expected, the ROM truncated at rank $r = r^*$ (panels (c) and (d)) has high accuracy both in representation and interpolation for all the points considered; instead, the ROM truncated at $r = r_{90}$ (panels (a) and (b)) fails to adequately reproduce the overall system state and loses accuracy when η or t increases. The ROM's performance is not affected by the selection of the points near the wall or in the middle of the aorta.

These results provide actionable indicators for the rank choice and the role played by the non-dominant modes. When all the modes are included in the training phase, $r = \mathrm{rank}(\tilde{\mathbf{X}})$,

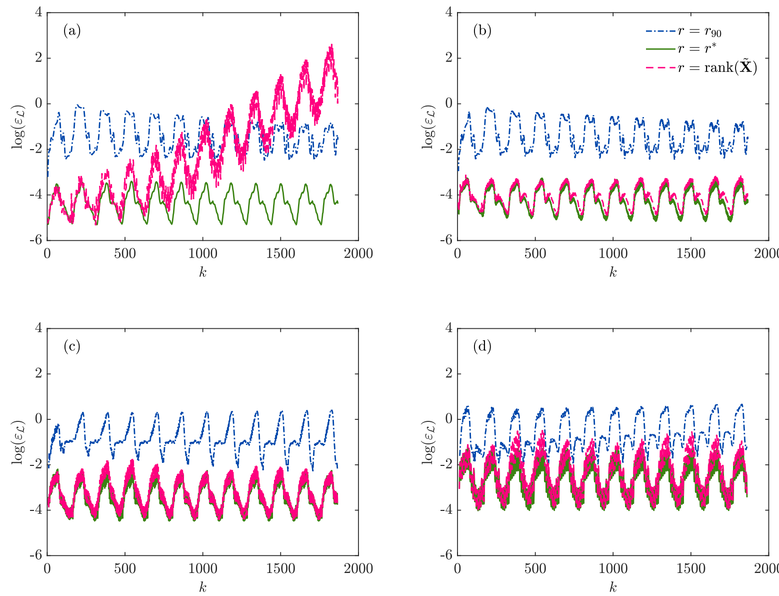

FIG. 4. Interpolation errors for (a) Case 1, (b) Case 2, (c) Case 3, and (d) Case 4 in Table I. In each plot different lines correspond to the ROMs with different truncation ranks $r = r_{90}$, $r = r^*$ and $r = \mathrm{rank}(\tilde{\mathbf{X}})$.

the ROM suffers from noise overfitting and loses its interpolation accuracy, especially when the training set is larger ($\eta = 0.5$). The loss in accuracy is difficult to predict given the lack of a priori error estimators. Hence, the use of a low-rank truncation not only aligns with a ROM's purpose (identification of the dominant modes and data compression) but also increases the ROM's prediction reliability at space-time locations where data are not available.

V. CONCLUSION

We analyzed the performance of an extended dynamic mode decomposition (xDMD)[9] on the task of ROM construction to approximate the fluid dynamics simulations of 3D blood

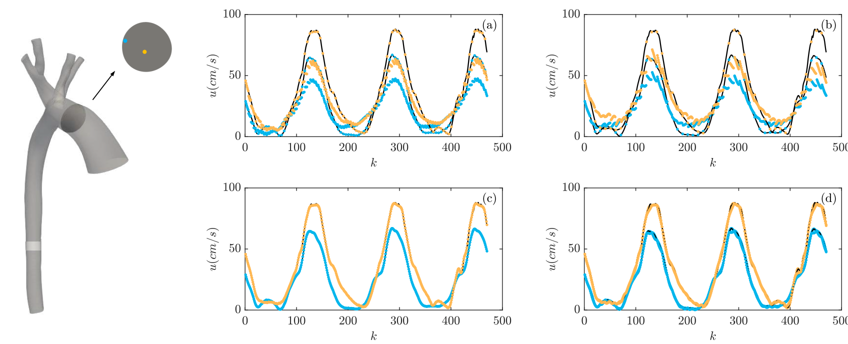

FIG. 5. Flow velocity u at kth time step, provided by `SimVascular` (continuous line) and estimated by the ROMs (dots) in the interpolation regime. The ROMs are trained for Case 3 in Table I in panels (a) and (c), and for Case 4 in Table I in panels (b) and (d). The data are reported for two points in one aorta's cross-section, as shown on the left. Panels (a) and (b) refer to the ROMs truncated at $r = r_{90}$, while (c) and (d) refer to the ROMs truncated at $r = r^*$.

flow in a patient-specific aorta. Our results show that xDMD is able to identify dominant spatiotemporal structures in the simulated data set and to provide an accurate approximation of numerical simulations. We explored relevant indicators of a ROM's performance in both representation and interpolation. These indicators are related to the choice of the truncation rank and linked to the number of retained singular values corresponding to the most relevant spatiotemporal structures. We found that a low-rank truncation, which preserves almost all the cumulative energy in the data, avoids overfitting and yields high accuracy and error stability. The xDMD-based ROMs demonstrate a remarkable robustness to the number of space-time training data. Finally, we verified the local accuracy of xDMD when used to predict time series at selected points in the flow domain. Overall, our study suggests that the use of xDMD is beneficial for time-dependent data compression and for computational saving when used in place of onerous numerical simulations.

The use of DMD for order reduction offers other benefits as well. By identifying the dominant spatially correlated structures (modes) in a given dataset and analyzing their temporal evolution (time dynamics), we can gain insight into the main features of the physical process, facilitating both data interpretation and reconstruction. DMD not only enables data compression, which is beneficial in many fields, but also allows us to reconstruct the system's behavior where data is unavailable (in interpolation or extrapolation regimes) with a single linear model providing predictions everywhere in space at any given time. This linear model is readily interpretable and is cleansed of noise, which would otherwise impede the reconstruction.

Our study demonstrates that the identification of an optimal DMD structure requires the selection of a low-rank approximation able to guarantee the ROM's accuracy in both representation and interpolation. This instill trust in the ROM's predictions, paving the way for their use in clinical practice. For example, DMD can be employed to predict blood flow beyond the available data to study variations in the flow waveform[17], to provide reliable real-time forecasting of tumor ablation treatment[25], and to facilitate spectral analysis in dynamic MRI acquisitions to advance the diagnostic potential[20].

Since DMD is formulated entirely in terms of (observational and/or simulated) data, it can be readily deployed in a wide range of applications, including in real-time simulation environments. In this context, newly available data can be absorbed in the training phase while updating the future state prediction.

ACKNOWLEDGMENTS

The authors would like to acknowledge Alison Marsden and Luca Pegolotti for providing the HFM data. The data that support the findings of this study are openly available in the repository www.vascularmodel.com.[14] This study has received funding from the Research Project of National Interest PRIN 2017 "Fluid dynamics of hearts at risk of failure: towards methods for the prediction of disease progression", funded by the Italian Ministry of Education, Universities and Research (MIUR); MIUR code: 2017A889FP_006; CUP code: J34I19001480001. GL has also received funding from the Marco Polo mobility scholarship for research 2021 of the University of Bologna. DMT was funded, in part, by the Air Force Office of Scientific Research under grant FA9550-21-1-0381, by the Office of Advanced Scientific Computing Research (ASCR) within the Department of Energy Office of Science under award number DE-SC0023163, and by the Strategic Environmental Research and Development Program (SERDP) of the Department of Defense under award RC22-3278.

Appendix A: Error bounds

In addition to the assumptions in Lemma 1, we assume that $\|\mathcal{L} - \mathbf{\Phi}_{\Delta t}\|_{L^\infty(\mathcal{H}_{\Delta t})} < +\infty$ and that $\mathbf{x}_k$ and $\mathbf{x}_k^{\mathcal{L}} \in \mathcal{H}_{\Delta t}$ for $k = 0, \ldots, M-1$. If $\mathcal{L}$ is sDMD, then the error $\delta_{\mathcal{L}}^M$ at time t_M, defined in (3), satisfies the inequality

$$\delta_{\mathcal{L}}^M \leq \mathrm{e}^{ML\Delta t}\delta_{\mathcal{L}}^0 + \| \mathcal{L} - \mathbf{\Phi}_{\Delta t} \|_{L^\infty(\mathcal{H}_{\Delta t})} \sum_{k=0}^{M-1} \mathrm{e}^{kL\Delta t}.$$

The proof, based on the triangle inequality, follows that for Theorem 4.3 in Ref. 11. Moreover, the gDMD is proven to have a tighter error bound than sDMD.[9]

Similarly, if $\mathcal{L}$ is the xDMD, then the error $\delta_{\mathcal{L}}^M$ at time t_M satisfies the inequality

$$\delta_{\mathcal{L}}^M \leq (1 + \mathrm{e}^{L\Delta t})^M \delta_{\mathcal{L}}^0 + \| \mathcal{L} - \mathbf{\Phi}_{\Delta t} \|_{L^\infty(\mathcal{H}_{\Delta t})} \sum_{k=0}^{M-1} (1 + \mathrm{e}^{L\Delta t})^k.$$

The xDMD is proven to have a tighter error bound than rDMD.[9] The error bounds provide a general guideline for the growth of errors.

Appendix B: Strategy to increase the xDMD efficiency

Direct evaluation of (9) requires the computation of $[\mathbf{B}_x\ \mathbf{b}_x] \in \mathbb{R}^{N\times N+1}$. Since N is large in any application of practical significance, this computation decreases the efficiency and accuracy of the algorithm. To avoid this bottleneck, we decompose the computation into two parts. First, we multiply only the first three terms of (9) thus leading to the matrix

$$\mathbf{C}_x = \mathbf{Y}\mathbf{V}_g\mathbf{\Sigma}_g^{-1} \in \mathbb{R}^{N\times r}. \tag{B1}$$

Second, we multiply the last term in (9) by $\tilde{\mathbf{x}}_k$, which gives a vector

$$\mathbf{d}_x = \mathbf{U}_g^\top \tilde{\mathbf{x}}_k \in \mathbb{R}^{r\times 1}. \tag{B2}$$

This procedure leads to

$$\mathbf{y}_{k+1} = \mathbf{C}_x\mathbf{d}_x, \tag{B3}$$

which is equivalent to (8).

An overall step-by-step implementation of xDMD with the efficient computational strategy described in this Section, is illustrated in Algorithm 1.

Algorithm 1: xDMD implementation based on the efficient computational strategy.

1. Compute the residual matrix $\mathbf{Y}$: $\mathbf{Y} = \mathbf{X}' - \mathbf{X}$, where $\mathbf{Y} \in \mathbb{R}^{N\times M}$
2. Introduce the matrix $\tilde{\mathbf{X}}$: $\tilde{\mathbf{X}}^\top = [\mathbf{X}\ \mathbf{1}]$, where $\tilde{\mathbf{X}}^\dagger \in \mathbb{R}^{N\times N+1}$
3. Compute the truncated SVD of $\tilde{\mathbf{X}}$: $\tilde{\mathbf{X}} \approx \mathbf{U}_g \mathbf{\Sigma}_g \mathbf{V}_g^\top$, where $\mathbf{U}_g \in \mathbb{R}^{N+1\times r}, \mathbf{\Sigma}_g \in \mathbb{R}^{r\times r}, \mathbf{V}_g \in \mathbb{R}^{M\times r}$
4. Compute the matrix $\mathbf{C}_x$: $\mathbf{C}_x = \mathbf{Y}\mathbf{V}_g \mathbf{\Sigma}_g^{-1}$, where $\mathbf{C}_x \in \mathbb{R}^{N\times r}$
5. Compute the vector $\mathbf{d}_x$: $\mathbf{d}_x = \mathbf{U}_g^\top \tilde{\mathbf{x}}_k$, where $\mathbf{d}_x \in \mathbb{R}^{r\times 1}$
6. Compute the residual at $k+1$: $\mathbf{y}_{k+1} = \mathbf{C}_x \mathbf{d}_x$, where $\mathbf{y}_{k+1} \in \mathbb{R}^{N\times 1}$
7. Compute the state at $k+1$: $\mathbf{x}_{k+1} = \mathbf{y}_{k+1} + \mathbf{x}_k$, where $\mathbf{x}_{k+1} \in \mathbb{R}^{N\times 1}$

REFERENCES

[1] H. Lu and D. M. Tartakovsky, “Lagrangian dynamic mode decomposition for construction of reduced-order models of advection-dominated phenomena,” Journal of Computational Physics **407**, 109229 (2020).

[2] S. Dutta, M. W. Farthing, E. Perracchione, G. Savant, and M. Putti, “A greedy non-intrusive reduced order model for shallow water equations,” Journal of Computational Physics **439**, 110378 (2021).

[3] L. Pegolotti, M. R. Pfaller, A. L. Marsden, and S. Deparis, “Model order reduction of flow based on a modular geometrical approximation of blood vessels,” Computer Methods in Applied Mechanics and Engineering **380**, 113762 (2021).

[4] D. Collia, G. Libero, G. Pedrizzetti, and V. Ciriello, “Surrogate models provide new insights on metrics based on blood flow for the assessment of left ventricular function,” Scientific Reports **12** (2022), 10.1038/s41598-022-12560-3.

[5] A. Marzadri, V. Ciriello, and F. P. J. de Barros, “Hyporheic flows in stratified sediments: Implications on residence time distributions,” Water Resources Research **60**, e2023WR035625 (2024).

[6] V. Ciriello, I. Lauriola, S. Bonvicini, V. Cozzani, V. Di Federico, and D. M. Tartakovsky, “Impact of hydrogeological uncertainty on estimation of environmental risks posed by hydrocarbon transportation networks,” Water Resources Research **53**, 8686–8697 (2017).

[7] P. J. Schmid, “Dynamic mode decomposition of numerical and experimental data,” Journal

of Fluid Mechanics **656**, 5–28 (2010).

[8] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," Journal of Computational Dynamics **1**, 391–421 (2014).

[9] H. Lu and D. M. Tartakovsky, "Extended dynamic mode decomposition for inhomogeneous problems," Journal of Computational Physics **444**, 110550 (2021).

[10] Z. Chen and D. Xiu, "On generalized residual network for deep learning of unknown dynamical systems," Journal of Computational Physics **438**, 110362 (2021).

[11] T. Qin, K. Wu, and D. Xiu, "Data driven governing equations approximation using deep neural networks," Journal of Computational Physics **395**, 620–635 (2019).

[12] G. Libero, D. Tartakovsky, and V. Ciriello, "Polynomial chaos enhanced by dynamic mode decomposition for order-reduction of dynamic models," Advances in Water Resources **186**, 104677 (2024).

[13] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition* (Society for Industrial and Applied Mathematics, 2016).

[14] N. M. Wilson, A. K. Ortiz, and A. B. Johnson, "The vascular model repository: A public resource of medical imaging data and blood flow simulation results," Journal of Medical Devices **7** (2013), 10.1115/1.4025983.

[15] A. Updegrove, N. M. Wilson, J. Merkow, H. Lan, A. L. Marsden, and S. C. Shadden, "SimVascular: An open source pipeline for cardiovascular simulation," Annals of Biomedical Engineering **45**, 525–541 (2016).

[16] A. Arzani and S. T. M. Dawson, "Data-driven cardiovascular flow modelling: examples and opportunities," **18** (2021), 10.1098/rsif.2020.0802.

[17] M. Habibi, S. T. M. Dawson, and A. Arzani, "Data-driven pulsatile blood flow physics with dynamic mode decomposition," **5**, 111 (2020).

[18] Y. T. Delorme, A.-E. M. Kerlo, K. Anupindi, M. D. Rodefeld, and S. H. Frankel, "Dynamic mode decomposition of fontan hemodynamics in an idealized total cavopulmonary connection," **46**, 041425 (2014).

[19] N. Groun, M. Villalba-Orero, E. Lara-Pezzi, E. Valero, J. Garicano-Mena, and S. Le Clainche, "Higher order dynamic mode decomposition: From fluid dynamics to heart disease analysis," **144**, 105384 (2022).

[20] E. Ilicak, S. Ozdemir, J. Zapp, L. R. Schad, and F. G. Zöllner, "Dynamic mode decom-

position of dynamic MRI for assessment of pulmonary ventilation and perfusion," **90**, 761–769 (2023).

[21]G. Di Labbio and L. Kadem, "Reduced-order modeling of left ventricular flow subject to aortic valve regurgitation," **31** (2019), 10.1063/1.5083054.

[22]A. Stuart and A. R. Humphries, *Dynamical Systems and Numerical Analysis* (Cambridge University Press, 1998).

[23]H. Lu and D. M. Tartakovsky, "Prediction accuracy of dynamic mode decomposition," SIAM Journal on Scientific Computing **42**, A1639–A1662 (2020).

[24]L. Pegolotti, M. R. Pfaller, N. L. Rubio, K. Ding, R. Brugarolas Brufau, E. Darve, and A. L. Marsden, "Learning reduced-order models for cardiovascular simulations with graph neural networks," Computers in Biology and Medicine **168**, 107676 (2024).

[25]G. C. Bourantas, M. Ghommem, G. C. Kagadis, K. Katsanos, V. C. Loukopoulos, V. N. Burganos, and G. C. Nikiforidis, "Real-time tumor ablation simulation based on the dynamic mode decomposition method: Dynamic mode decomposition in tumor ablation simulation," **41**, 053301 (2014).