



Randomized model order reduction

Alessandro Alla¹ · J. Nathan Kutz²

Received: 1 February 2017 / Accepted: 12 December 2018 /
Published online: 17 January 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

The singular value decomposition (SVD) has a crucial role in model order reduction. It is often utilized in the offline stage to compute basis functions that project the high-dimensional nonlinear problem into a low-dimensional model which is then evaluated cheaply. It constitutes a building block for many techniques such as the proper orthogonal decomposition (POD) and dynamic mode decomposition (DMD). The aim of this work is to provide an efficient computation of low-rank POD and/or DMD modes via randomized matrix decompositions. This is possible due to the randomized singular value decomposition (rSVD) which is a fast and accurate alternative of the SVD. Although this is considered an offline stage, this computation may be extremely expensive; therefore, the use of compressed techniques drastically reduce its cost. Numerical examples show the effectiveness of the method for both POD and DMD.

Keywords Nonlinear dynamical systems · Proper orthogonal decomposition · Dynamic mode decomposition · Randomized linear algebra

Mathematics Subject Classification (2010) 65L02 · 65M02 · 37M05 · 62H25

1 Introduction

Reduced-order models (ROMs) continue to play a critically enabling role in modern, large-scale scientific computing applications [3]. The ROM architecture is

Communicated by: Anthony Patera

✉ Alessandro Alla
alla@mat.puc-rio.br

J. Nathan Kutz
kutz@uw.edu

¹ Department of Mathematics, PUC-Rio, Rio de Janeiro, 22451-900, Brazil

² Department of Applied Mathematics, University of Washington, Seattle, WA 98195-3925, USA

being exploited in many simulation-based physics and engineering systems in order to render tractable many high-dimensional simulations. Fundamentally, the ROM algorithmic structure is designed to construct low-dimensional subspaces, typically computed with SVD, where the evolution dynamics can be projected using a Galerkin method. Thus, instead of solving a high-dimensional system of differential equations (e.g., millions or billions of degrees of freedom), a rank ℓ model can be constructed in a principled way. Three steps are required for this low-rank approximation: (i) numerical solutions of the original high-dimensional system; (ii) dimensionality reduction of this solution data typically produced with an SVD; and (iii) Galerkin projection of the dynamics on the low-rank subspace using sparse and/or greedy sampling methods, i.e., gappy POD [26]. The first two steps are often called the *offline* stage of the ROM architecture, whereas the third step is known as the *online* stage. Offline stages are exceptionally expensive but enable the (cheap) online stage to potentially run in real time. In this manuscript, we integrate recent innovations in randomized linear algebra methods [21], particularly as it relates to the SVD, and compressive sampling in order to (i) improve the computational efficiency of the second step of the ROM architecture, namely the building of low-rank subspaces used for Galerkin projection, and (ii) provide a rapid evaluation of the nonlinear terms in the ROM model using compressive sampling of the DMD.

Randomized methods for matrix computations provide an efficient computation of low-rank structures in data matrices, which is a foundational aspect of machine learning and big data applications. Such algorithms exploit the fact that the target rank of interest, ℓ , is significantly smaller than the high-dimensional data under consideration, thus allowing for a compressed representation of the original data matrix with only a minor loss in precision. Indeed, the randomized singular value decomposition (rSVD) exploits this compressive representation and provides a low-rank approximation of the SVD while using only slightly more than ℓ random measurements. The randomized SVD algorithm comes with strong theoretical error bounds and the advantage that the error can be controlled by oversampling and subspace iterations [9, 18, 21]. The rSVD can produce a low-rank approximation to the full high-dimensional state vector by taking advantage of the QR algorithm with random sampling. The QR approximation gives a mapping from high- to low-dimensional subspaces and back again. The rSVD algorithm additionally provides an estimate of the target rank and oversampling required [9, 18, 21]. In the case of ROMs, there may only be a couple hundred modes of interest ($\ell \approx 200$), whereas the numerical solution of the original high-dimensional system may have millions or billions of degrees of freedom. ROMs allow one to simulate this system with a differential equation of dimension ℓ , thus greatly reducing computational time. Randomized techniques circumvent the challenge of traditional (deterministic) SVD reduction which requires significant memory and processing resources for the high-dimensional data generated from full-state simulations. Randomized techniques are robust, reliable, and computationally efficient, and can be used to construct the low-rank POD modes which accurately approximate the POD modes of the high-dimensional data matrix [6, 18]. There exist several strategies for obtaining the compressed matrix, and using random projections is certainly the most robust off-the-shelf approach. Randomized

algorithms have been in particular studied for computing the near-optimal low-rank singular value decomposition by [11, 17, 20, 31]. The seminal contribution [13] extends and surveys this work.

The potential of randomized sampling techniques has already been realized in the ROMs community where it has been used recently to construct preconditioners of parameter-dependent matrices for the solution of large systems of parameter-dependent equations [32] and to construct a low-rank approximation of the linearized parameter-to-observable map in modeling Antarctic ice sheets [14]. In addition to randomized techniques, compressive sampling strategies are of growing interest for matrix computations as they also allow for the approximation of decompositions with few measurements. Much like randomized algorithms, compressive sampling takes advantage of the inherent sparsity of the spatiotemporal dynamics in an appropriate basis. Thus, the target rank ℓ determines the number of sample points required. Compressive sampling can be used with the dynamic mode decomposition (DMD) [16] to enact a *compressive DMD* [4, 9] approximation for the Galerkin-projected dynamics [1]. The DMD method is an attractive alternative to the standard POD–Galerkin reduction which also uses sparse sampling through the gappy POD [10] and/or (discrete) empirical interpolation method (DEIM/EIM) architecture (see, e.g., [2, 5]). DMD is a least-squares regression technique that integrates Fourier transforms and SVD. The DMD method originated in the fluid dynamics community as a method to decompose complex flows into a simple representation based on low-rank, spatiotemporal coherent structures. The growing success of DMD stems from the fact that it is an equation-free, data-driven method capable of providing an accurate decomposition of a complex system into spatiotemporal coherent structures that may be used for short-time future state prediction and control. For ROMs, only the leading-order DMD modes are necessary to reconstruct the nonlinearity in a governing PDE and predict future states of the high-dimensional dynamics. Ultimately, the low-rank structure inherent in ROMs and DMD allows the community to exploit sparse measurements to reconstruct an accurate approximation of the high-dimensional system. In this work, two new innovations are introduced that leverage our current computational capabilities, namely compressive sampling for enhancing the DMD method for ROMs and randomized singular value decompositions for constructing efficient POD basis elements.

The structure of the paper is as follows. In Section 2, we review model reduction techniques based upon the POD method, the discrete empirical interpolation method (DEIM) and the DMD applied to general nonlinear dynamical systems. In Section 3, we highlight innovations of randomized techniques for matrix computations, which is the building block for our new approach. Section 3.1 focuses on the application of compressed matrix decompositions in model order reduction. Finally, numerical tests are presented in Section 4. Throughout the paper, we use the following notation: all matrices and vectors are in bold letters. The basis functions are denoted by the matrix Ψ with different superscripts denoting how we computed the basis, e.g., Ψ^{POD} represents the basis functions from the POD method. The rank of the POD basis functions is ℓ , the rank of the nonlinear term is k , whereas p is the number of measurements utilized in the compressed techniques and \hat{U} is the number of over sampling.

2 Model order reduction techniques

We consider the general system of high-dimensional, ordinary differential equations as follows:

$$\begin{cases} \mathbf{M}\dot{\mathbf{y}}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{f}(t, \mathbf{y}(t)), & t \in (0, T], \\ \mathbf{y}(0) = \mathbf{y}_0. \end{cases} \quad (2.1)$$

where $\mathbf{y}_0 \in \mathbb{R}^n$ is a given initial data, $\mathbf{M}, \mathbf{A} \in \mathbb{R}^{n \times n}$ are given matrices and $\mathbf{f} : [0, T] \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a continuous function in both arguments and locally Lipschitz-type with respect to the second variable. It is well-known that under these assumptions there exists a unique solution for (2.1). This class of problems arises in a wide range of applications, but especially from the numerical approximation of partial differential equations. In such cases, the dimension of the problem n is the number of spatial grid points used from discretization, and it typically is very large. The numerical solution of system (2.1) may be very expensive to compute; therefore, it is often useful to simplify the complexity of the problem by means of ROMs. The model reduction approach is based on projecting the nonlinear dynamics onto a low-dimensional manifold utilizing projectors that contain information from the full, high-dimensional system. The norm, here and in the sequel of the section, can be interpreted as $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v}$ and $\| \cdot \|^2 = \langle \cdot, \cdot \rangle$.

Let us assume that we have computed some basis functions $\Psi = \{\psi_j\}_{j=1}^\ell \in \mathbb{R}^{n \times \ell}$ of rank ℓ for (2.1). We can project the dynamics onto the low-rank basis functions using the following:

$$\mathbf{y}(t) \approx \Psi \mathbf{y}^\ell(t), \quad (2.2)$$

where $\mathbf{y}^\ell(t)$ are functions on \mathbb{R}^ℓ and defined on the time interval from $[0, T]$. We note that we are working with a Galerkin-type projection where we consider only a few basis functions whose support is nonlocal, unlike finite element basis functions.

Inserting the projection assumption (2.2) into the full model (2.1), and making use of the orthogonality of the basis functions, the reduced model takes the following form as follows:

$$\begin{cases} \mathbf{M}^\ell \dot{\mathbf{y}}^\ell(t) = \mathbf{A}^\ell \mathbf{y}^\ell(t) + \Psi^T \mathbf{f}(t, \Psi \mathbf{y}^\ell(t)), & t \in (0, T], \\ \mathbf{y}^\ell(0) = \mathbf{y}_0^\ell, \end{cases} \quad (2.3)$$

where,

$$(\mathbf{M}^\ell)_{ij} = \langle \psi_i, \mathbf{M} \psi_j \rangle, (\mathbf{A}^\ell)_{ij} = \langle \psi_i, \mathbf{A} \psi_j \rangle \in \mathbb{R}^{\ell \times \ell}$$

and $\mathbf{y}_0^\ell = (\mathbf{M}^\ell)^{-1} (\Psi)^T \mathbf{M} \mathbf{y}_0 \in \mathbb{R}^\ell$. The system (2.3) is achieved following a Galerkin projection. If the dimension of the system is $\ell \ll n$, then a significant dimensionality reduction is accomplished.

This section focuses on several model order reduction techniques as they constitute the building blocks of the proposed method. In particular, we recall three key innovations for model reduction: POD, DEIM, and DMD. These techniques provide an efficient projector for the reduction of the complexity of the problem under consideration.

2.1 The POD method and reduced-order modeling

One popular method for reducing the complexity of the system is the so-called proper orthogonal decomposition (POD). The idea was proposed in [26] and is detailed here for completeness. We build an equidistant grid in time with constant step size Δt . Let $t_0 := 0 < t_1 < t_2 < \dots < t_m \leq T$ with $t_j = j\Delta t, \quad j = 0, \dots, m$. Let us assume we know the exact solution of (2.1) on the time-grid points $t_j, j \in \{0, \dots, m\}$. Our aim is to determine a POD basis of any rank $\ell \ll n$ to optimally describe the set of data collected in time by solving the following minimization problem:

$$\min_{\psi_1, \dots, \psi_\ell \in \mathbb{R}^n} \sum_{j=0}^m \alpha_j \left\| \mathbf{y}(t_j) - \sum_{i=1}^{\ell} \langle \mathbf{y}(t_j), \psi_i \rangle \psi_i \right\|^2 \quad \text{such that } \langle \psi_i, \psi_j \rangle = \delta_{ij}, \quad (2.4)$$

where the coefficients α_j are nonnegative and $\mathbf{y}(t_j)$ are the so-called *snapshots*, e.g., the solution of (2.1) at a given time t_j .

Solving (2.4), we look for an orthonormal basis $\{\psi_i\}_{i=1}^{\ell}$ which minimizes the distance between the sequence $\mathbf{y}(t_j)$ with respect to its projection onto this unknown basis. The matrix $\mathbf{Y} := \{\sqrt{\alpha_0}y(t_0) \dots, \sqrt{\alpha_m}y(t_m)\} \in \mathbb{R}^{n \times (m+1)}$ contains the collection of snapshots $\mathbf{y}(t_j)$ as columns. We note that in the numerical experiments, we will set $\alpha_j \equiv 1, j = \{0, \dots, m\}$. It is useful to look for $\ell \ll \min\{m, n\}$ in order to reduce the dimension of the problem considered. The solution of (2.4) is given by the SVD of the snapshots matrix $\mathbf{Y} = \Psi \Sigma \mathbf{V}^T$, where we consider the first ℓ - columns $\{\psi_i\}_{i=1}^{\ell}$, of the orthogonal matrix Ψ and set $\Psi^{\text{POD}} = \Psi$.

To concretely apply the POD method, the choice of the truncation parameter ℓ plays a critical role. There are no a-priori estimates which guarantee the ability to build a coherent reduced model, but one can focus on heuristic considerations, introduced in [26], so as to have the following ratio close to one as follows:

$$\mathcal{E}(\ell) = \frac{\sum_{i=1}^{\ell} \sigma_i^2}{\sum_{i=1}^r \sigma_i^2}, \quad (2.5)$$

where r is the rank of the snapshot matrix \mathbf{Y} . This indicator is motivated by the fact that the error in (2.4) is given by the singular values we neglect as follows:

$$\sum_{j=0}^m \alpha_j \left\| \mathbf{y}(t_j) - \sum_{i=1}^{\ell} \langle \mathbf{y}(t_j), \psi_i \rangle \psi_i \right\|_F^2 = \sum_{i=\ell+1}^r \sigma_i^2. \quad (2.6)$$

We note that the error (2.6) is strictly related to the computation of the snapshots and it is not related to the reduced dynamical system. More recently in [12], the authors have introduced a hard-thresholding technique for determining the truncation of the SVD when the data contains a low-rank signal with noise. This method provides a principled approach to rank selection. Selecting the target rank ℓ for the POD reduction has already been noted as a critical task. In practice, rank selection for a standard POD reduction (not randomized) is often chosen from experience with simulations of

the governing PDE. For instance, one could choose the rank ℓ such that the quantity $\mathcal{E}(\ell)$ in (2.5) is close to 1.

2.2 Discrete empirical interpolation method

For a review of DEIM, we closely follow the presentation in [5]. The ROM introduced in (2.3) is a nonlinear system where the significant challenge with the POD-Galerkin approach is the computational complexity associated with the evaluation of the nonlinearity. To illustrate this issue, we consider the nonlinearity in (2.3) as follows:

$$\mathbf{F}(t, \mathbf{y}^\ell(t)) = (\Psi^{\text{POD}})^T \mathbf{f}(t, \Psi^{\text{POD}} \mathbf{y}^\ell(t)) = \langle \Psi^{\text{POD}}, \mathbf{f}(t, \Psi^{\text{POD}} \mathbf{y}^\ell(t)) \rangle.$$

To compute this inner product, the variable $\mathbf{y}^\ell(t) \in \mathbb{R}^\ell$ is first expanded to an n -dimensional vector $\Psi^{\text{POD}} \mathbf{y}^\ell(t) \in \mathbb{R}^n$, then the nonlinearity $\mathbf{f}(t, \Psi^{\text{POD}} \mathbf{y}^\ell(t))$ is evaluated and, at the end, we return back to the reduced-order model. This is computationally expensive, since it implies that the evaluation of the nonlinear term requires computing the full, high-dimensional model; therefore, the reduced model is not independent of the full dimension n . To avoid this computationally expensive, high-dimensional evaluation, the gappy POD method was introduced [10]. In its original formulation, random and sparse sampling was proposed for computing the required nonlinear inner products. Advances in gappy methods have led to the state-of-the-art *empirical interpolation method* (EIM, [2]) and *discrete empirical interpolation method* (DEIM, [5]) methods which are now broadly used in the ROMs community.

The computation of the POD basis functions $\mathbf{U} = \{U_1, \dots, U_k\}$ for the nonlinear part are related to the set of the snapshots $\mathbf{f}(t_j, \mathbf{y}(t_j))$ where $\mathbf{y}(t_j)$ is already computed from (2.1). The DEIM approximation of $\mathbf{f}(t, \mathbf{y}(t))$ is as follows:

$$\mathbf{f}^{\text{DEIM}}(t, \mathbf{y}^{\text{DEIM}}(t)) = \mathbf{U}(\mathbf{S}^T \mathbf{U})^{-1} \mathbf{f}(t, \mathbf{y}^{\text{DEIM}}(t))$$

where $\mathbf{S} \in \mathbb{R}^{n \times k}$ and $\mathbf{y}^{\text{DEIM}}(t) = \mathbf{S}^T \Psi^{\text{POD}} \mathbf{y}^\ell(t)$. The matrix \mathbf{S} pick coordinates in \mathbb{R}^n where the nonlinearity is evaluated and the selection is made according to an LU decomposition algorithm with pivoting [5], or following the QR decomposition with pivoting [7]. The error between $\mathbf{f}(t, \mathbf{y}(t))$ and its DEIM approximation \mathbf{f}^{DEIM} is given by the following:

$$\|\mathbf{f} - \mathbf{f}^{\text{DEIM}}\|_2 \leq c \|(\mathbf{I} - \mathbf{U}\mathbf{U}^T) \mathbf{f}\|_2 \quad \text{with } c = \|(\mathbf{S}^T \mathbf{U})^{-1}\|_2$$

where different error performance is achieved depending on the selection of the interpolation points in S as shown in [7].

2.3 Dynamic mode decomposition

DMD is an *equation-free*, data-driven method capable of providing accurate assessments of the spatiotemporal coherent structures in a given complex system, or short-time future estimates of such a systems. It traces its origins to pioneering work of Bernard Koopman in 1931 [15], whose work was revived in a set of papers starting in 2004 [23–25]. The DMD provides the eigenvalues and eigenvectors of the best

fit linear system relating a snapshot matrix and a time shifter version of the snapshot matrix at some later time.

Consider the following data snapshot matrices as follows:

$$\mathbf{Y} = \left[\begin{array}{c|c|c|c} \mathbf{y}(t_0) & \mathbf{y}(t_1) & \cdots & \mathbf{y}(t_{m-1}) \end{array} \right], \quad \mathbf{Y}' = \left[\begin{array}{c|c|c|c} \mathbf{y}(t_1) & \mathbf{y}(t_2) & \cdots & \mathbf{y}(t_m) \end{array} \right] \tag{2.7}$$

with $\mathbf{y}(t_j)$ an initial condition to (2.1) and $\mathbf{y}(t_{j+1})$ its corresponding output after some prescribed evolution time $\Delta t > 0$ with there being m initial conditions considered. The DMD method computes the best linear operator \mathbf{A} relating to the matrices above:

$$\mathbf{Y}' = \mathbf{A}\mathbf{Y}, \tag{2.8}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is unknown. The exact DMD algorithm proceeds as follows [28]: First, we collect data \mathbf{Y}, \mathbf{Y}' and compute the reduced, or economy, singular value decomposition of \mathbf{Y} :

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T.$$

We note that the use of the economy SVD is suggested since the matrices $\mathbf{Y}, \mathbf{Y}' \in \mathbb{R}^{n \times m}$ with $n \gg m$. Furthermore, the economy $\mathbf{U} \in \mathbb{R}^{n \times m}$ is sufficient to provide the same approximation of the regular SVD given the limited amount of snapshots.

Then, we compute the least-squares fit \mathbf{A} that satisfies $\mathbf{Y}' = \mathbf{A}\mathbf{Y}$ and project onto the POD modes \mathbf{U} . Specifically, the Moore-Penrose pseudo-inverse of \mathbf{Y} allows us to compute $\mathbf{A} = \mathbf{Y}'\mathbf{Y}^\dagger$, where the Moore-Penrose algorithm provides the least-square fitting procedure. In terms of its low-rank projection, this yields as follows:

$$\tilde{\mathbf{A}} = \mathbf{U}^T \mathbf{A} \mathbf{U} = \mathbf{U}^T \mathbf{Y}' \mathbf{V} \mathbf{\Sigma}^{-1},$$

and compute the eigen-decomposition of $\tilde{\mathbf{A}}$:

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda},$$

where $\mathbf{\Lambda}$ are the DMD eigenvalues. Finally, the DMD modes Ψ^{DMD} are given by the following:

$$\Psi^{\text{DMD}} = \mathbf{Y}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{W}. \tag{2.9}$$

Unlike a POD approximation, the DMD modes contained in the columns of Ψ^{DMD} are not orthogonal. However, this is not a problem in practice unless there is non-normal behavior in dynamical system. The data \mathbf{Y}, \mathbf{Y}' may come from a nonlinear system $y(t_{j+1}) = f(y(t_j))$, in which case the DMD modes are related to eigenvectors of the infinite-dimensional Koopman operator. More details can be found in [16]. We may interpret DMD as a model reduction technique if data is acquired from a high-dimensional model, or a method of system identification if the data comes from measurements of an unknown system. For the purpose of this work, we consider the DMD-Galerkin method, where the assumption (2.2) holds true for Ψ given by (2.9). We note that the techniques we provide aim to speed up the computation of the offline stage.

3 Randomized linear algebra in model order reduction

Randomized linear algebra is of growing importance for the analysis of high-dimensional data [21]. Specifically, randomized techniques attempt to construct low-rank matrix decompositions that are computationally efficient and accurate approximations of the standard matrix decompositions such as QR and SVD. Randomized algorithms can be parallelized and distributed for large matrices and there are several implementations of the randomized techniques in *MATLAB* or *R* that are now available via open source [9, 27, 30]. The algorithms that result from using randomized sampling techniques are not only computationally efficient, but are also simple to implement as they rely on standard matrix-matrix multiplication and unpivoted QR factorization. In addition to computational efficiency, convergence properties of randomized methods have been well-characterized. In addition, many practical innovations, including power sampling schemes, can improve convergence performance [13]. This is beyond the scope of the current work, and the interested reader is encouraged to consult the overview work [21].

Consider a randomized algorithm to compute the low-rank matrix approximation [13] is shown as follows;

$$\mathbf{Y} \approx \mathbf{Q} \mathbf{B},$$

$$n \times m \quad n \times \ell \quad \ell \times m$$

where ℓ denotes the target rank and is assumed to be $\ell \ll \min\{m, n\}$. Random matrix theory provides a simple and elegant solution for computing the low-rank approximation by creating a random sampling matrix $\Omega \in \mathbb{R}^{m \times \ell}$ where the entries are drawn from, for example, a Gaussian distribution with mean zero and variance one. Then, a sampled matrix $\mathbf{X} \in \mathbb{R}^{n \times \ell}$ is computed as follows:

$$\mathbf{X} = \mathbf{Y}\Omega.$$

If the matrix \mathbf{Y} has exact rank ℓ , then the sampled matrix \mathbf{X} spans, with high probability, a basis for the column space. However, most data matrices in practice are only dominated by rank- ℓ features since the singular values $\{\sigma_i\}_{i=\ell+1}^n$ are nonzero. Thus, instead of just using ℓ samples, it is favorable to slightly oversample $p = \ell + \hat{U}$, where p denotes the number of measurements and \hat{U} the number of over sampling. In practice, small values of $\hat{U} \approx 20$ are sufficient to obtain a good basis that is comparable to the best possible basis [21]. An orthonormal basis $\mathbf{Q} \in \mathbb{R}^{n \times \ell}$ is then obtained via the QR decomposition $\mathbf{X} = \mathbf{QR}$, such as the following:

$$\mathbf{Y} \approx \mathbf{Q}\mathbf{Q}^T \mathbf{Y}.$$

Finally, \mathbf{Y} is projected to this low-dimensional space, as shown in the following;

$$\mathbf{B} = \mathbf{Q}^T \mathbf{Y},$$

where $\mathbf{B} \in \mathbb{R}^{\ell \times m}$. The matrix \mathbf{B} can then be used to efficiently compute the matrix decomposition of interest such as the SVD. The oversampling \hat{U} allows one to control the approximation error [13, 21]. The algorithm is summarized in Algorithm 1. In Fig. 1, we show the decay of the singular values for different level of the randomized SVD. As expected by increasing the number of sampling p , we obtain a more

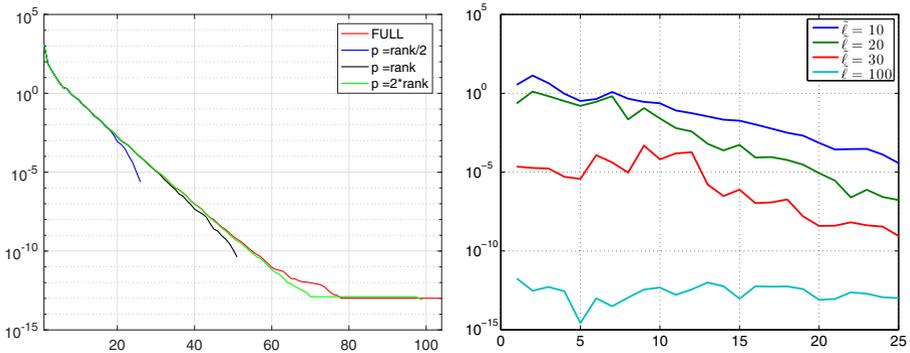


Fig. 1 Convergence of the singular values for randomized SVD with different number of measurements ℓ . We consider the full matrix (red), $p = r/2$ (blue), $p = r$ (black), $p = 2r$ (green) where r is the rank of the snapshots matrix in test 1. (left) Absolute difference for singular values computed by standard and randomized method with different numbers of over sampling $\tilde{\ell}$

accurate approximation. For the sake of completeness, in this example, we relate the number of sampling p to the rank of the matrix, which is in general not known in advance. However, we note that the target rank ℓ can be estimated by a variety of *rank revealing* algorithms [8, 13, 19, 30]. A complete description of these methods goes beyond the scopes of this paper. In the right panel, we compute the absolute difference between the true and approximated singular values. As one can see with an over sample of $\tilde{\ell} = 20$, we already get accurate results. Clearly, the more oversampling the better quality.

Algorithm 1 Randomized SVD (rSVD).

Require: Matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$

- 1: Draw a Gaussian random matrix $\Omega \in \mathbb{R}^{m \times p}$ with mean 0 and variance 1,
 - 2: Form the sample matrix $\mathbf{X} = \mathbf{Y}\Omega$,
 - 3: Compute the QR decomposition of $\mathbf{X} : \mathbf{X} = \mathbf{Q}\mathbf{R}$,
 - 4: Set $\mathbf{B} = \mathbf{Q}^T \mathbf{Y}$,
 - 5: Compute SVD of $\mathbf{B} = \hat{\mathbf{U}}\Sigma\mathbf{V}^T$,
 - 6: Set $\mathbf{U} = \mathbf{Q}\hat{\mathbf{U}}$.
-

Finally, we note that the computational cost for the randomized SVD is $O(mn\ell)$, whereas the deterministic SVD is $O(mn^2)$. We can easily see that if $\ell \ll n$, we have an impressive speed up of the method. Our next result follows closely ([13, Theorem 10.5]) and provide an error for the randomized SVD:

Theorem 3.1 Suppose that $\mathbf{Y} \in \mathbb{R}^{n \times m}$ is a matrix with singular values $\sigma_1 \geq \sigma_2 \geq \dots$. Choose a target rank $\ell \geq 2$ and an oversampling parameter $\tilde{\ell} \geq 2$, where

$p = \ell + \tilde{\ell} \leq \min\{m, n\}$. Draw an $m \times p$ standard Gaussian matrix Ω , and construct the sample matrix $\mathbf{X} = \mathbf{Y}\Omega$. Then, the expected approximation error is as follows:

$$\mathbb{E}\|(\mathbf{I} - \mathbf{P}_{\mathbf{X}})\mathbf{Y}\|_F \leq \sqrt{\left(1 + \frac{\ell}{\tilde{\ell} - 1}\right)} \sqrt{\sum_{j>\ell} \sigma_j^2}. \quad (3.10)$$

The interested reader can find in [13] the proof of the theorem.

3.1 Compressed model order reduction techniques

Model order reduction techniques are usually based on snapshots that collect data on the underlying dynamical system. The SVD decomposition of the data matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$ provides a low-dimension projector operator that allows one to obtain surrogate models. However, the SVD may be computationally expensive and, for this reason, we propose the use of Algorithm 1 to reduce the offline cost of the method. The main idea is to consider basis functions not from the full set of measurements but from a few spatially incoherent measurements. We introduce the measurement matrix $\mathbf{C} \in \mathbb{R}^{p \times n}$ which produces the compressed matrix $\mathbf{X} \in \mathbb{R}^{p \times m}$ such as the following:

$$\mathbf{X} = \mathbf{C}\mathbf{Y}.$$

Here, we consider sparse measurements of the snapshots matrix in order to compute POD and DMD from this new compressed snapshot matrix. In this paper, we assume that snapshots matrix is almost square, e.g., $n \approx m$, and one can imagine this is a realistic situation working with an explicit time scheme or in a many-query context. In the following subsections, we provide further details about compressed POD, compressed POD-DEIM, and compressed DMD.

3.1.1 Compressed POD

The compressed POD method works, as does POD, starting with a snapshot matrix with the aim to compute solutions of the problem (2.4) in a fast and reliable way. As discussed before, the solution of the minimization problem leads to an expensive singular value decomposition problem. Here, the idea is to apply the randomized SVD technique in Section 2 for the approximation of (2.4). The method works as follows: (i) we collect the snapshot set and (ii) we produce the SVD modes which satisfy the optimization problem (2.4). We make use of the optimality conditions in [29] in order to take advantage of the randomized SVD. In this way, we are able to approximate the high-dimensional POD modes using compressed POD modes which are computed significantly faster. Clearly, the number of samples points plays a crucial role. The algorithm is summarized in Algorithm 2.

The error in the minimization problem is now associated with subsampling of the randomized SVD and follows directly from Theorem 3.1 since the optimality conditions are derived as the standard method and we replace the computation of the

Algorithm 2 Compressed POD (cPOD).

Require: Snapshot matrix $\mathbf{Y} \in \mathbb{R}^{n \times m}$, ℓ number of basis functions, p number of measurements.

- 1: Compute the randomized SVD (see Algorithm 1), $[\mathbf{U}, \Sigma, \mathbf{V}] = rsvd(\mathbf{Y})$
 - 2: Set $\Psi_i = \mathbf{U}_i$ for $i = 1, \dots, \ell$.
-

SVD with its randomized algorithm. The expectation error for problem (2.4) reads as follows:

$$\mathbb{E} \left(\sum_{j=0}^m \alpha_j \left\| \mathbf{y}(t_j) - \sum_{i=1}^{\ell+\tilde{\ell}} \langle \mathbf{y}(t_j), \psi_i \rangle \psi_i \right\|_F^2 \right) \leq \left(1 + \frac{\ell}{\tilde{\ell} - 1} \right) \sum_{j>\ell} \sigma_j^2. \tag{3.1}$$

where $\tilde{\ell}$ is the number of additional samples in the compressed technique. We note that we consider the expectation value of the error due to the random measurements we consider. The error (3.1) is now related to the computation of the set of snapshots and the number of samples $p = \ell + \tilde{\ell}$. We note that if the singular values of the snapshot matrix decay rapidly a minimal amount of samples drives the error close to the theoretically minimum value. However, if the singular values do not decay rapidly, we can lose accuracy. As explained in [13], the error bound always exceeds this reference error, but it may be polynomially larger, depending on the ratio between the target rank ℓ and the oversampling parameter $\tilde{\ell}$. For $\tilde{\ell}$ small (say, less than five), the error is somewhat variable because the small singular values of a nearly square Gaussian matrix are very unstable. As the oversampling increases, the performance improves quickly. We refer to [13] and the reference therein for more details about the error of the randomized SVD. We also note that both errors (3.1) and (2.6) depend on the sum of the singular values neglected. Finally, we note that the POD basis functions for the snapshot matrix $\mathbf{Y} \in \mathbb{R}^{n \times (m+1)}$ can be also computed from the eigenvalue problem for the matrix $\mathbf{Y}\mathbf{Y}^T$ or $\mathbf{Y}^T\mathbf{Y}$. Interested readers can see, e.g., [29] for a more comprehensive description. Regardless, if the dimension of the matrix \mathbf{Y} is such that $n \approx m$, then the computation of the eigenvalue problem will not lead to a faster approximation than the SVD. This further motivates our approach through the rSVD.

For the randomized model reduction advocated here, as in standard model reduction, the choice of the target rank ℓ is critical. One can also use past experience with a given PDE as a guide to select it. In the absence of prior knowledge, *rank revealing* algorithms have also been developed for the randomized SVD which estimates the target rank ℓ of interest [8, 13, 19, 22, 30], including data sets with slowly decaying singular values. Thus, the target rank does not need to be known in advance. Importantly, the blocked randUTV [22] provides accuracy very close to that of the SVD for problems such as low-rank approximation, solving ill-conditioned linear systems, determining bases for various subspaces associated with the matrix, etc. Further-

more, randUTV produces highly accurate approximations to the singular values of the desired matrix. Unlike the SVD, the randomized algorithm of randUTV builds a UTV factorization in an incremental, single-stage, and non-iterative way, making it possible to halt the factorization process once a specified tolerance has been met. Thus, an advantage of randomized algorithms is that one can build the reduced space in an adaptive manner until a certain accuracy is satisfied. The code to perform this computation is available as open source [22]. Ultimately, this gives a computationally cheap way to estimate the rank, and then the standard randomized SVD method can be used on ensuing computations. Although a complete description of randomized rank revealing algorithms goes beyond the scopes of this work, we would like to mention that it already constructs the reduced space which is desired for the approximation of the range of the snapshot matrix \mathbf{Y} . For this reason, we show in Fig. 2 a comparison of the k -rank projection error $\|\mathbf{Y} - \mathbf{U}_k \mathbf{U}_k^T \mathbf{Y}\|$ where the projection operators \mathbf{U}_k are computed with randomized SVD and blocked randUTV techniques. We further note that the randUTV is a blocked randomized technique and we need to choose a-priori the number of blocks.

3.1.2 Compressed POD-DEIM

Similarly to the compressed POD method, we aim to apply the rSVD to the DEIM approach. The DEIM method considers the computation of the SVD for both the snapshots of the solution and snapshots of the nonlinear term. We note that, although the online stage benefits from a sparse evaluation of the nonlinearity, the offline stage is even more expensive than POD itself. The goal is to substitute the full dimensional SVD with the much smaller randomized SVD. In this way, we can highly reduce the cost of the computational costs and, at the same time, obtain accurate results.

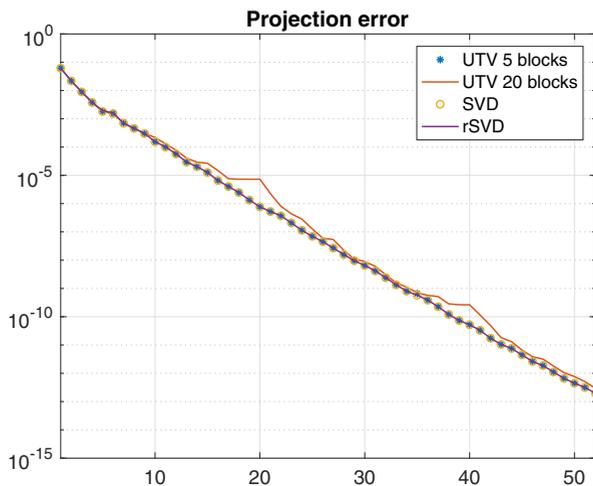


Fig. 2 Comparison of the projection operator computed by SVD (yellow circles), rSVD (purple line, $\tilde{\ell} = 5$), randUTV with 5 blocks (blue stars) and randUTV with 20 blocks (red line)

3.2 Compressive DMD

We can also combine ideas from compressive sampling to compute the dynamic mode decomposition from a few measurements of the data. This method was already introduced in [4]. Here, it is applied in as Galerkin projection method. It is possible to either collect data the full amount of data \mathbf{Y}, \mathbf{Y}' as in Section 2.3 or projected data \mathbf{X}, \mathbf{X}' , where $\mathbf{X} = \mathbf{C}\mathbf{Y}, \mathbf{X}' = \mathbf{C}\mathbf{Y}'$ and $\mathbf{C} \in \mathbb{R}^{p \times n}$ is the measurement matrix. We will call the matrices \mathbf{X}, \mathbf{X}' the output-projected snapshot matrices. Similar to equation above, \mathbf{X} and \mathbf{X}' are related by the following:

$$\mathbf{X}' = \mathbf{A}_\mathbf{X}\mathbf{X}. \tag{3.2}$$

The goal, as in DMD, is to compute eigenvalues and eigenvectors of the unknown matrix $\mathbf{A}_\mathbf{X}$. The method differs from the standard DMD since we are using sparse measurements. Under general assumptions, it is possible to prove the convergence of the method when the number of measurements p increases [18, 21], but there is an important empirical observation that $p = \ell + 20$ is sufficient to given an accurate approximation of the low-rank subspace in practice [9, 21]. Clearly the cDMD method is computationally more efficient, and the method is summarized in Algorithm 3.

Algorithm 3 Compressive DMD (cDMD).

- Require:** Snapshots $\{\mathbf{y}(t_0), \dots, \mathbf{y}(t_m)\}, \mathbf{C} \in \mathbb{R}^{p \times n}$
 Set $\mathbf{Y} = [\mathbf{y}(t_0), \dots, \mathbf{y}(t_{m-1})]$ and $\mathbf{Y}' = [\mathbf{y}(t_1), \dots, \mathbf{y}(t_m)]$,
 $\mathbf{X} = \mathbf{C}\mathbf{Y}, \mathbf{X}' = \mathbf{C}\mathbf{Y}'$,
 3: Compute the SVD of $\mathbf{X}, \mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$,
 Define $\tilde{\mathbf{A}}_\mathbf{X} := \mathbf{U}^T\mathbf{Y}'\mathbf{V}\mathbf{\Sigma}^{-1}$,
 Compute eigenvalues and eigenvectors of $\tilde{\mathbf{A}}_\mathbf{X}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$,
 6: Set $\Psi^{\text{DMD}} = \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{W}$.
-

Once the DMD basis functions Ψ^{DMD} are computed, we utilize assumption (2.2) and obtained a surrogate model of the form (2.3).

4 Numerical tests

In this section, we present our numerical tests using our three proposed compressed/randomized SVD strategies of the last section. In our numerical computations, we use the finite difference method to reduce a partial differential equation into the form (2.1) and integrate the system with a semi-implicit Euler scheme in the first example where the linear part is treated implicitly and the nonlinear term explicitly. In the second example, we integrate the system with an implicit Euler scheme which involves the Newton method. All the numerical simulations reported in this paper are performed on a MacBook Pro with an Intel Core i5, 2.2Ghz and 8GB RAM using MATLAB R2013a.

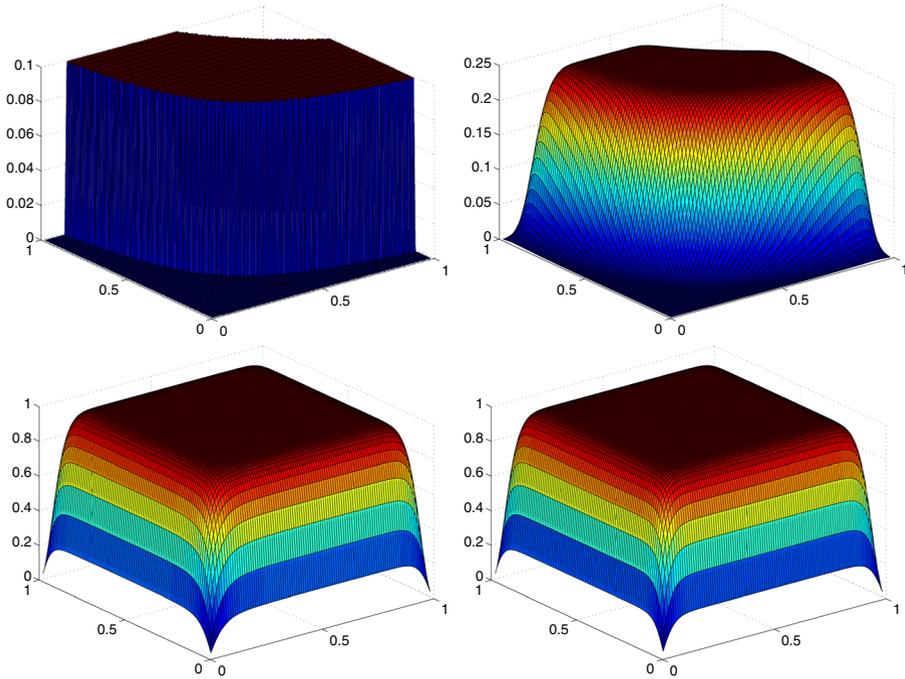


Fig. 3 Test 1: solution of equation (4.3) at time $t = \{0, 0.1\}$ (top) and $t = \{1.5, 3\}$ (bottom)

In the following numerical examples, we build different surrogate models, such as POD, compressed POD (cPOD), POD-DEIM, compressed POD-DEIM (cPOD-cDEIM), DMD and compressed DMD (cDMD), and compare their performance in terms of CPU time and the error with respect to a reference solution computed by a high-fidelity, finite-difference approximation. We select two numerical examples, the first one considers a time-dependent semilinear PDEs, whereas the second studies a semilinear elliptic parametric equations. Both examples lead to the same conclusions. In the numerical tests, the number of samples utilized for the compression of the snapshot matrix is $p = \ell + 20$, where ℓ is the number of desired basis functions and $\tilde{\ell} = 20$ the number of oversampling. The randomized techniques provides also accurate algorithms to detect the rank of a matrix as mentioned in [13]. As shown in Fig. 1, this turns out to be very efficient for both accuracy and computational cost.

4.1 Test 1: semilinear equation

Let us consider the following semilinear parabolic equations:

$$\begin{cases} y_t(x, t) - \theta \Delta y(x, t) + \mu (y(x - t) - y^3(x, t)) = 0, & (x, t) \in \Omega \times [0, T], \\ y(x, 0) = y_0(x), & x \in \Omega, \\ y(\cdot, t) = 0, & x \in \partial\Omega, t \in [0, T], \end{cases} \quad (4.3)$$

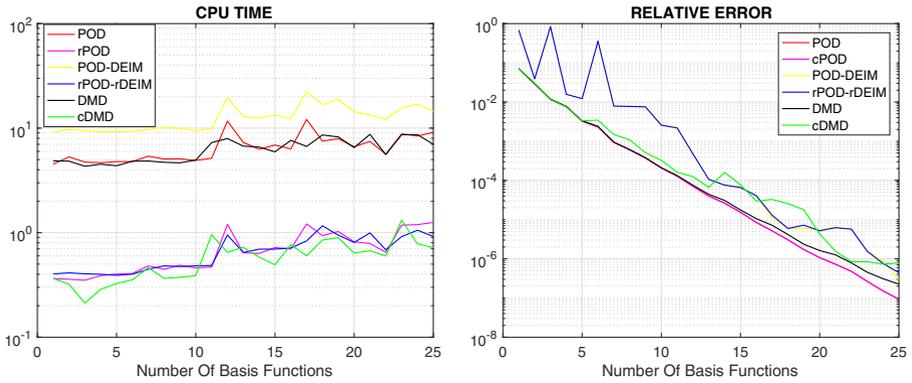


Fig. 4 Test 1: CPU time of the offline-online stages (left) and relative error in Frobenius norm (right). We compare the following methods: POD (red), cPOD (magenta), POD-DEIM (yellow), cPOD-cDEIM (blue, DMD (black), cDMD (green). Number of model are always the same for all the methods

where $\Omega = [0, 1] \times [0, 1]$, $T = 5$, $\theta = 0.1$, $\mu = 11$, $x = (x_1, x_2)$, $y_0(x) = 0.1$ if $0.1 \leq x_1 x_2 \leq 0.6$ and 0 elsewhere. The POD basis vectors are built upon 10000 equidistant snapshots. The FD discretization yields a system of ODEs of the same form as (2.1) with $n = 10000$. The solution of this equation generates a stationary solution $y(x, t) \equiv 1$ for large t as shown in Fig. 3.

The complexity of problem (4.3) is reduced by model order reduction. When dealing with model order reduction, it is relevant to consider the CPU time of the simulation and the error. In general, it is important to have a trade-off between the two quantities. Figure 4 considers the CPU time on the left panel. As we can see, the compressed techniques are faster than the standard reduction techniques. We note

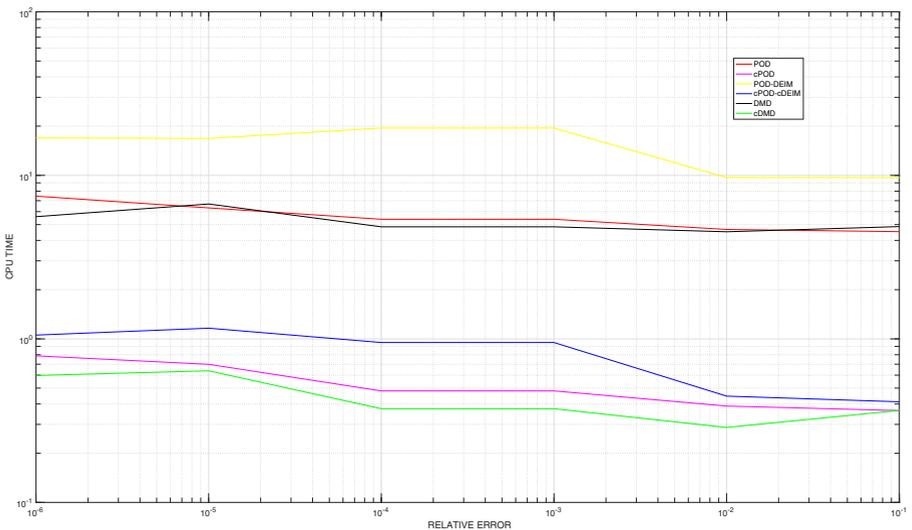


Fig. 5 Test 1: CPU time for a given error tolerance

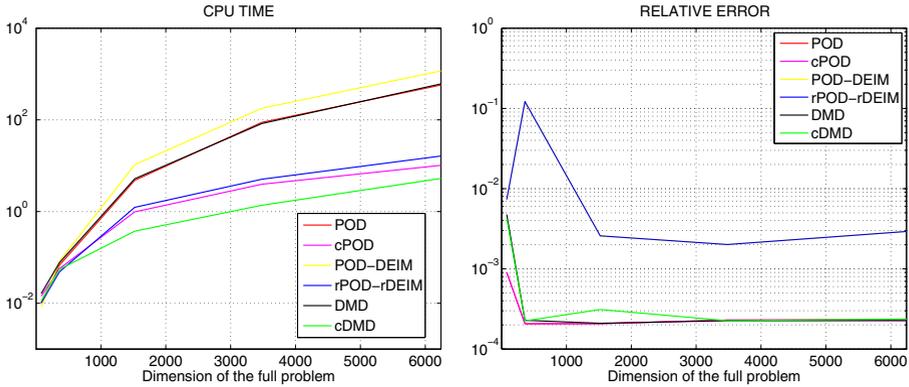


Fig. 6 Test 1: scaling of the CPU time with increasing dimension of the snapshot set (left), relative error for ten modes and different snapshot set

here that for the CPU time we consider both offline and online stages. Although we do not aim at an improvement of the online stage, in this work, one might also consider a further speed up as suggested in [1]. It is somehow clear that the compressed DMD provides the fastest approximation since it does not require the computation of the randomized SVD. However, we show in Fig. 4 the relative error computed with respect to the Frobenius norm. As we can see, POD and cPOD, such as POD-DEIM and cPOD-cDEIM, perform exactly the same results. Slightly different are the results from DMD and cDMD. All these techniques perform with very high accuracy. As expected, the POD-DEIM and its related compressed technique is less accurate since we do not evaluate the nonlinearity for the full state. We also note that the error computed using the POD-DEIM and the cPOD-cDEIM methods matches perfectly.

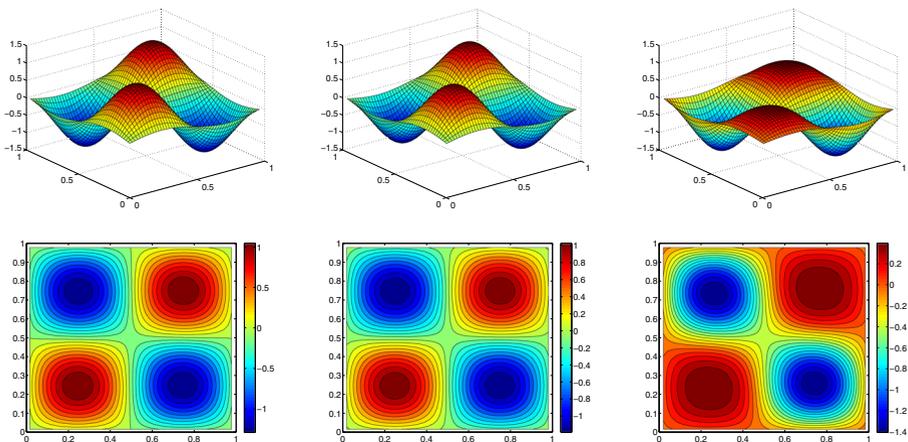


Fig. 7 Test 2: solution of problem (4.4) for different parametric configurations (top) and contour lines (bottom). We consider $(\mu_1, \mu_2) = (0.2, 5)$ (left), $(\mu_1, \mu_2) = (7, 0.4)$ (middle), $(\mu_1, \mu_2) = (9, 9)$ (right)

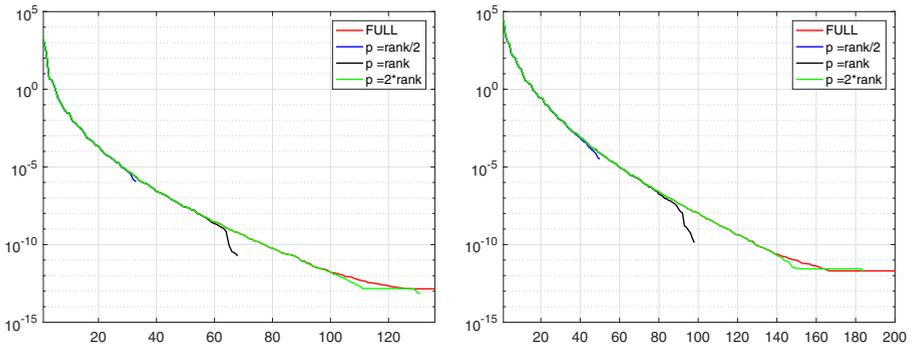


Fig. 8 Test 2: decay of the singular values for the snapshot set with different number of measurements (left) and for the nonlinear term (right)

Furthermore, once can also see the instability of the POD-DEIM method due to an insufficient approximation of the nonlinear term.

A more appropriate comparison is given in Fig. 5. There, we look at the CPU time for a given fixed error tolerance. As it is shown, the compressed techniques reach a desired error faster than standard method gaining at least 1 order of magnitude.

Another important feature to investigate when dealing with compressed techniques is how the CPU time scales with different dimensions of the snapshot matrix. The computation of the SVD is, computationally, the most expensive part of the method and its cost varies according to the dimension of the snapshot set. Here, we consider a square matrix. As we can see in Fig. 6 in the left panel, the CPU time scale shows that we gain more than two orders of magnitude in speed up as the dimension increases. Thus, it provides a powerful technique that allows one to significantly reduce the computational costs in the offline stage. In the right panel, we can see the relative error for ten basis functions.

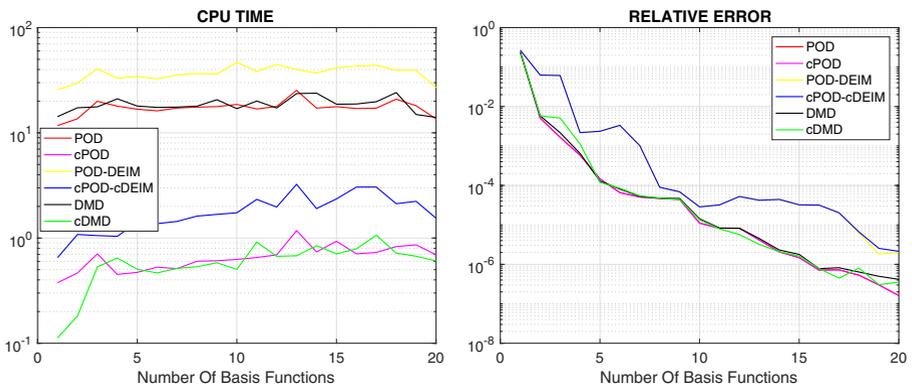


Fig. 9 Test 2: CPU time of the offline-online stages (left) and Relative Error in Frobenius norm (right). We compare the following methods: POD (red), cPOD (magenta), POD-DEIM (yellow), cPOD-cDEIM (blue, DMD (black), cDMD (green). Number of model are always the same for all the methods

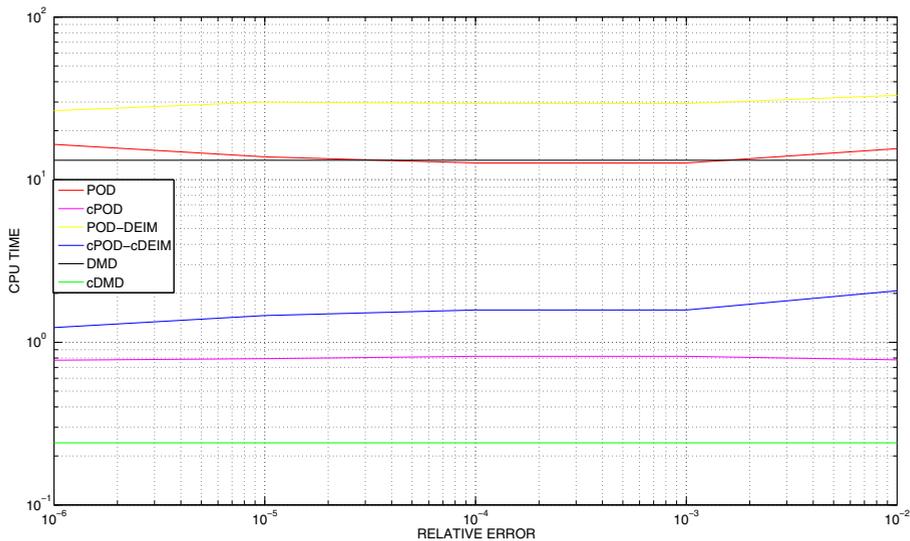


Fig. 10 Test 2: CPU time for a given error tolerance

4.2 Test 2: parametric example

The second numerical example concerns a parametric elliptic equation. This example follows closely from [5]. Let the dynamics given by the following:

$$\begin{cases} -\Delta u(x, y) + s(u(x, y); \mu) = f(x, y) & (x, y) \in \Omega \\ u(x, y) = 0 & (x, y) \in \partial\Omega \end{cases} \quad (4.4)$$

where the spatial variable $(x, y) \in \Omega = (0, 1)^2$ and the parameters are $\mu = (\mu_1, \mu_2) \in \mathcal{D} = [0, 01, 10]^2 \subset \mathbb{R}^2$ with a homogeneous Dirichlet boundary conditions. The nonlinearity and the source term are given, respectively, by the following:

$$s(u, \mu) = \frac{\mu_1}{\mu_2} (e^{\mu_2 u} - 1), \quad f(x, y) = 100 \sin(2\pi x) \sin(2\pi y).$$

We numerically solve the system applying Newton’s method to the nonlinear equations resulting from a FD discretization. The full dimension of the discretized problem is $n = 2500$ and the parameter training set contains 2500 equidistributed. The solution of (4.4) is shown in Fig. 7. Note that different choice of the parameter configuration leads different solutions.

For the sake of completeness, we show the decay of the singular values of the snapshot matrix on the left panel of Fig. 8. We compare the singular values computed with the standard SVD and the randomized SVD as we increase the sampling points of the original matrix. As expected, it leads to improved approximations and, at the same time, faster approximations of the problem. A similar behavior comes from the nonlinear term (right panel).

We show the CPU time for all the methods studied in the left panel of Fig. 9 and the error behavior in the right panel. One can see we obtain similar results as in the previous test, where the compressed techniques are faster than the standard methods and accuracy is of same order.

A more appropriate comparison is given in Fig. 10. There, we look at the CPU time for a given fixed error tolerance. As it is shown, the compressed techniques reach a desired error faster than standard method gaining one order of magnitude.

5 Conclusion

Model order reduction is a successful and commonly used technique that projects nonlinear high-dimensional dynamical systems and PDEs into low-dimensional surrogate models using optimal basis functions computed from information of the system. Although the solution of the surrogate model is computationally efficient, the computation of the basis functions remains computationally expensive. In this paper, we have demonstrated through several examples that compressed (randomized) techniques are a promising approach to circumventing expensive offline stages in model order reduction. In particular, when dealing with large snapshot matrices, we suggest the use of randomized singular valued decomposition for the proper orthogonal decomposition and compressed dynamic mode decomposition. They both provide very accurate solutions and promise significant computational savings in the offline stage, which turns out to be the most expensive part of the building block for the surrogate model.

Critical for enacting these computational enhancements is the advent of randomized linear algebra techniques. Randomized linear algebra methods have been recently surveyed in [21]. Indeed, the methods are continuing to mature and have many critical error bounds associated with their proposed matrix factorizations. These efficient matrix decompositions are tremendously important for analyzing high-dimensional data sets and/or for producing the low-dimensional subspaces required for ROMs. Randomized techniques have continued to experience modifications that increase their efficiency and broaden the range of applicability of the methods. More broadly, randomized methods have application to classical (nonrandomized) techniques for solving the same problems, such as, e.g., Krylov methods, subspace iteration, and rank-revealing QR factorizations.

Ultimately, ROMs are primarily concerned with producing rapid evaluation of surrogate models that represent the original high-dimensional system with a given accuracy. Given the significant computational bottleneck for evaluating the low-dimensional projection, it is surprising the randomized linear algebra techniques have yet to penetrate the ROMs community. We have explicitly demonstrated that such randomized techniques can be a significant enhancement of the ROMs architecture. It should be used whenever possible given the current maturity of the technique and the error bounds available.

Funding information This study is supported by the Department of Energy (grant no. DE-SC0009324) and the U.S. Air Force Office of Scientific Research (FA9550-15-1-0385).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

- Alla, A., Nathan Kutz, J.: Nonlinear model reduction via dynamic mode decomposition. *SIAM J. Sci. Comput.* **39**, B778–B796 (2017)
- Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations *Comptes Rendus Mathématique*, **339**, pp. 667–672 (2004)
- Benner, P., Gugercin, S., Willcox, K.: A survey of Projection-Based model reduction methods for parametric dynamical systems. *SIAM Rev.* **57**, 483–531 (2015)
- Brunton, S.L., Proctor, J.L., Kutz, J.N.: Compressive sampling and dynamic mode decomposition. *J. Comp. Dyn.* **2**, 165–191 (2015)
- Chatarantabut, S., Sorensen, D.: Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **32**, 2737–2764 (2010)
- Drineas, P., Mahoney, M.W.: RandNLA: randomized numerical linear algebra. *Communications of the ACM* **59.6**, 80–90 (2016)
- Drmac, Z., Gugercin, S.: A new selection operator for the discrete empirical interpolation method - improved a priori error bound and extensions. *SIAM J. S.i. Comput.* **38**, A631–A648 (2016)
- Duersch, J., Gu, M. (2015)
- Erichson, N.B., Voronin, S., Brunton, S.L., Kutz, J.N.: Randomized matrix decompositions using R, arXiv: [1608.02148](https://arxiv.org/abs/1608.02148) (2016)
- Everson, R., Sirovich, L.: Karhunen-loève procedure for gappy data. *J. Opt. Soc. Am. A* **12**, 1657–1664 (1995)
- Frieze, A., Ravi, K., Vempala, S.: Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)* **51.6**, 1025–1041 (2004)
- Gavish, M., Donoho, D.L.: The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Trans Inform. Theory* **60**, 5040–5053 (2014)
- Halko, N., Martinsson, P.-G., Tropp, J.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* **53**, 217–288 (2011)
- Isaac, T., Petra, N., Stadler, G., Ghattas, O.: Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet. *J. Comp. Phys.* **296**, 348–368 (2015)
- Koopman, B.O.: *Hamiltonian* systems and transformation in hilbert space. *PNAS* **17**, 315–318 (1931)
- Kutz, J.N., Brunton, S., Brunton, B., Proctor, J.: Dynamic mode decomposition: Data-driven modeling of complex systems. *SIAM Press* (2016)
- Liberty, E., Woolfe, F., Martinsson, P.-G., Rokhlin, V.: Randomized algorithms for the low-rank approximation of matrices. *Proc. Natl. Acad. Sci.* **104**, 20167–20172 (2007)
- Mahoney, M.W.: Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.* **3.2**, 123–224 (2011)
- Martinsson, P.-G.: factorizations, blocked rank-revealing QR: how randomized sampling can be used to avoid single-vector pivoting. arXiv: [1505.08115](https://arxiv.org/abs/1505.08115) (2015)
- Martinsson, P.-G., Rokhlin, V., Tygert, M.: A randomized algorithm for the decomposition of matrices. *Appl. Comput. Harmon. Anal.* **30**, 47–68 (2011)
- Martinsson, P.-G.: Randomized methods for matrix computations and analysis of high dimensional data, arXiv: [1607.01649](https://arxiv.org/abs/1607.01649) (2016)
- Martinsson, P.-G., Quintana-Orti, G., Heavner, N.: randUTV: A blocked randomized algorithm for computing a rank-revealing UTV factorization, arXiv: [1703.00998](https://arxiv.org/abs/1703.00998) (2017)
- Mezić, I., Banaszuk, A.: Comparison of systems with complex behavior. *Physica D: Nonlinear Phenomena* **197**, 101–133 (2004)
- Mezić, I.: Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dyn.* **41**, 309–325 (2005)
- Mezić, I.: Analysis of fluid flows via spectral properties of the Koopman operator. *Annu. Rev. Fluid Mech.* **45**, 357–378 (2013)

26. Sirovich, L.: Turbulence and the dynamics of coherent structures. Parts I-II Q. Appl. Math. **XVL**, 561–590 (1987)
27. Szlam, A., Kluger, Y., Tygert, M.: An implementation of a randomized algorithm for principal component analysis, arXiv: [1412.3510](https://arxiv.org/abs/1412.3510) (2014)
28. Tu, J., Rowley, C., Luchtenberg, D., Brunton, S., Kutz, J.N.: On dynamic mode decomposition theory and applications. J. Comput. Dyn. **1**, 391–421 (2014)
29. Volkwein, S.: Model Reduction Using Proper Orthogonal Decomposition. Lecture Notes, University of Konstanz (2013)
30. Voronin, S., Martinsson, P.-G.: RSVDPACK: Subroutines for computing partial singular value decompositions via randomized sampling on single core, multi core, and GPU architectures, arXiv: [1502.05366](https://arxiv.org/abs/1502.05366) (2015)
31. Woolfe, F., Liberty, E., Rokhlin, V., Tygert, M.: A fast randomized algorithm for the approximation of matrices. Appl. Comput. Harmon. Anal. **25**, 335–366 (2008)
32. Zahm, O., Nouy, A.: Interpolation of inverse operators for preconditioning parameter-dependent equations. SIAM J. Sci. Comput. **38**, 1004–1074 (2016)