Journal of Computational Physics 335 (2017) 1–20

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

LUPOD: Collocation in POD via LU decomposition

M.-L. Rapún [a], F. Terragni [b], J.M. Vega [a,*]

[a] E.T.S.I. Aeronáutica y del Espacio, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[b] Gregorio Millán Institute for Fluid Dynamics, Nanoscience and Industrial Mathematics, Universidad Carlos III de Madrid, 28911 Leganés, Spain

ARTICLE INFO

Article history:
Received 5 July 2016
Received in revised form 29 October 2016
Accepted 6 January 2017
Available online 18 January 2017

Keywords:
Proper orthogonal decomposition
Singular value decomposition
Collocation points
Reduced order models
Galerkin projection

ABSTRACT

A collocation method is developed for the (truncated) POD of a set of snapshots. In other words, POD computations are performed using only a set of collocation points, whose number is comparable to the number of retained modes, in a similar fashion as in collocation spectral methods. Intending to rely on simple ideas which, moreover, are consistent with the essence of POD, collocation points are computed via the LU decomposition with pivoting of the snapshot matrix. The new method is illustrated in simple applications in which POD is used as a data-processing method. The performance of the method is tested in the computationally efficient construction of reduced order models based on POD plus Galerkin projection for the complex Ginzburg–Landau equation in one and two space dimensions.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

Let us consider a set of real or complex vectors, which will be called below the *snapshots* and may be organized as the columns of a *snapshot matrix*. The *proper orthogonal decomposition* (POD) of the snapshot set [9] and the *singular value decomposition* (SVD) of the snapshot matrix [14] were invented as different methods by Pearson [20] in 1901 and by Beltrami and Jordan in 1873–74 (see [28]), respectively. Nevertheless, these methods are closely related to each other since, in its simplest form, the POD of a snapshot set can be calculated using the SVD of the associated snapshot matrix. Both POD and SVD are very popular and effective in applications dealing with:

- *Data processing tasks*, such as pattern recognition, error filtering, data compression [32], and missing data recovering [13] in two-dimensional databases; see also [18] and [19] for multi-dimensional extensions.
- The derivation of *reduced order models* (ROMs) for extended systems such as the Navier–Stokes equations [6], in which the high-dimensional governing equations are projected onto the lower-dimensional set of POD modes. These ROMs, in turn, may be *preprocessed ROMs*, in which the POD modes are calculated offline just once, in a preprocess, or *adaptive ROMs*, whose POD modes are first calculated and then updated online, along the integration. Preprocessed ROMs are convenient to simulate permanent dynamics many times, under strongly correlated conditions (which share the POD modes); the possibly computationally expensive preprocess is compensated by the inexpensive online operation of the ROM, which is used many times. Adaptive ROMs, instead, are convenient when the ROM is to be used either a few times or many times under strongly uncorrelated conditions.

* Corresponding author.
E-mail address: josemanuel.vega@upm.es (J.M. Vega).

http://dx.doi.org/10.1016/j.jcp.2017.01.005
0021-9991/© 2017 Elsevier Inc. All rights reserved.

Let us consider a set of snapshots, $\boldsymbol{s}_k \in \mathbb{C}^J$, for $k = 1, \ldots, K$, which to fix ideas can be seen as K spatial distributions of a state variable in a spatial mesh consisting in J grid points; the index k can be seen as discretizing the time variable (but could also discretize a relevant parameter). Truncated POD applied to these snapshots yields a set of $M < K$ (orthonormal) POD modes, $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M \in \mathbb{C}^J$, such that the snapshots can be approximated as

$$\boldsymbol{s}_k \simeq \sum_{m=1}^{M} \alpha_k^m \boldsymbol{u}_m, \tag{1}$$

where the coefficients α_k^m are calculated upon orthogonal projection of the snapshots onto the POD modes, as

$$\alpha_k^m = \langle \boldsymbol{s}_k, \boldsymbol{u}_m \rangle. \tag{2}$$

Moreover, the POD modes are optimal in the sense that they yield the best root mean square (RMS) approximation (1) retaining M modes. In principle, the inner product $\langle \cdot, \cdot \rangle$ is the usual Euclidean product in $\mathbb{C}^J$, namely

$$\langle \boldsymbol{s}_1, \boldsymbol{s}_2 \rangle := \sum_{j=1}^{J} s_1^j \overline{s}_2^j, \tag{3}$$

where the overbar stands hereinafter for the complex conjugate.

Denoting hereinafter vectors and matrices with boldface lower case and capitals, respectively, truncated POD can be calculated via truncated SVD applied to the snapshot matrix, $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K]$, which reads

$$\boldsymbol{S} \simeq \boldsymbol{S}^{\text{trunc}} := \boldsymbol{U} \boldsymbol{\Sigma} \overline{\boldsymbol{V}}^{\top}, \tag{4}$$

where the (orthonormal) POD modes are precisely the columns of the $J \times M$-matrix $\boldsymbol{U}$ and the elements of the $M \times M$-diagonal matrix $\boldsymbol{\Sigma}$ are the retained *singular values*, sorted in decreasing order. Standard SVD-formulae [14] allow for calculating the quadratic error (defined in terms of the Frobenius norm, $\| \cdot \|_{\text{Fro}}$) of the approximation (4) in terms of the strictly positive singular values of the matrix $\boldsymbol{S}$, $\sigma_1, \ldots, \sigma_R$, as

$$\text{quadratic error} := \| \boldsymbol{S} - \boldsymbol{S}^{\text{trunc}} \|_{\text{Fro}} = \sqrt{\sigma_{M+1}^2 + \ldots + \sigma_R^2}, \tag{5}$$

where R is the rank of $\boldsymbol{S}$. SVD is very effectively calculated using, e.g., the MATLAB command 'svd', option 'econ'.

In typical applications, the main advantage of POD/SVD is appreciated when the snapshots are strongly correlated among each other (due to the underlying physical laws, such as mass and momentum conservation in fluid dynamics data), which implies that

$$M \ll J \tag{6}$$

and leads to a strong *dimension reduction*. POD-based ROMs consist in sets of M equations, while the number of degrees of freedom of the original problem coincides with the size of the snapshots, J, which is usually much larger. For instance, in applications to three-dimensional aerodynamic flows, J may be of the order of 10^7, while reasonable approximations may be obtained in terms of $M \sim 100$ POD modes [1].

However, interesting and useful as it is, the combination of *POD and projection of the governing equations* may be quite computationally expensive when the projection is based on the inner product (3). For linear problems, the projection can be performed just once, at the outset, which reformulates the resulting ROM in matrix form. For general nonlinear problems, instead, the projection must be performed at each timestep and involves a computational cost that scales as $M \times J$ and may offset the dimension reduction advantages of POD-based reduced order models. This difficulty is currently overcome in various ways. For instance, in the so-called *trajectory piecewise-linear approach* [24,25], the nonlinearity is piecewise approximated along the integration by a weighted sum of linearized models.

On the other hand, the effective manipulation of the M POD modes should be possible using a reduced inner product based on a limited amount of components of the POD modes, $N \sim M$, namely

$$\langle \boldsymbol{s}_1, \boldsymbol{s}_2 \rangle := \sum_{n=1}^{N} s_1^{j_n} \overline{s}_2^{j_n}, \tag{7}$$

for some conveniently selected index set $\{j_1, \ldots, j_N\}$ that defines a set of *collocation points*. When applied to partial differential equations, appropriate formulae must be derived to calculate the spatial derivatives using information only from the collocation points. This is quite similar to what happens with *collocation* in spectral methods [16], in which the number of collocation points equals the number of retained spectral modes.

The collocation points can be selected in various ways, e.g., *equispacedly* [1] or *concentrated* in those spatial regions that are known to be most significant [29]. Such simple selections require to take N somewhat large compared to M (say, $N = 3M$, [1,29]). Of course, a more efficient selection results by using a specific *sampling method*, such as those implemented in the so-called *missing point estimation* [3], *empirical interpolation* [4], *discrete empirical interpolation* [10], and *hyper-reduction* [26]. Sampling methods improve the *quality of the snapshots* by attending to the following issues:

1. Since standard POD emphasizes those patterns exhibiting the largest 'energy' (measured with the Euclidean norm), spatially localized patterns may be masked by spatially spread patterns that exhibit a larger energy but are numerically smaller. Appropriate collocation, locating collocation points in the localized patterns, may solve this difficulty.
2. Temporally localized data (i.e., localized in a few snapshots), on the other hand, may also be masked by temporally spread patterns. As in item 1, appropriate *temporal sampling* (which is also performed in some of the above mentioned sampling methods and in the so-called *reduced basis method* [21]), selecting only the appropriate snapshots to perform POD, may also solve this difficulty.

We do not intend to compete with the above mentioned sampling methods in improving the quality of the snapshots. Instead, we focus on improving the computational efficiency of the Galerkin projection and attend to the computational efficiency of the collocation method, avoiding that its computational cost offsets in part the advantages of using the inner product (7). This may be not crucial when sampling is to be performed just once, as in the above mentioned preprocessed ROMs, but it can be crucial in the application of adaptive ROMs, in which sampling may be needed many times along the integration. Thus, the main goal of this paper is to develop a method that is as (i) simple, (ii) consistent with POD, and (iii) as computationally inexpensive as possible. As a byproduct, the quality of the snapshots will also be improved by the method derived below, which will implicitly involve sampling in both space and time; however this is not the goal of the paper.

Our collocation method first selects both a convenient set of collocation points and an appropriate subset of the snapshot set. Then POD is applied to the selected snapshots using the reduced inner product (7) based on the collocation points. The collocation points and snapshots selection is performed by applying to the snapshot matrix $\boldsymbol{S}$ (whose columns are the given snapshots) an incomplete LU decomposition (namely, Gauss elimination) with double pivoting, based on linear combinations of the columns of $\boldsymbol{S}$. If the snapshots are strongly correlated, the LU decomposition selects a (limited) number of rows of $\boldsymbol{S}$ and a limited number of columns, such that the whole set of matrix columns (i.e., the snapshots) can be reconstructed as linear combinations of the selected columns (i.e., the selected snapshots) using information only from the selected rows, which define the collocation points. The very essence of the performed LU decomposition implies that the reconstruction of the whole set of snapshots using only linear combinations of the selected snapshots is consistent with the expansions (1), while the fact that such reconstructions are based only on the collocation points is consistent with the inner product (7). This argument will be made more precise in §2.2. Since, in the end, the method consists in combining LU decomposition and POD, it will be referred to hereinafter as the *LUPOD method*.

Simple and natural as it is, the LUPOD method turns out to be quite computationally inexpensive, which is consistent with the above mentioned main goal of the paper. In fact, the computational cost of the LU decomposition is comparable to that of the QR decomposition, which is usually performed, as a previous step, in standard POD. In other words, the computational cost of LUPOD is comparable to that of standard POD.

With the above ideas in mind, the remaining of the paper is organized as follows. The LUPOD method is described in §2, where (keeping in mind applications to partial differential equations) the computation of the spatial derivatives of the resulting POD modes is also considered. The method is illustrated in several simple snapshot sets in §3. The performance of the LUPOD method in improving the computational efficiency of POD-based ROMs will be tested in §4, considering the complex Ginzburg–Landau equation in 1D and 2D. The outcomes of the paper are discussed in §5.

2. The LUPOD method

The method is described in §2.1, justified in §2.2, and commented in §2.3. As above, we consider a set of snapshots, $\boldsymbol{s}_k \in \mathbb{C}^J$, for $k = 1, \ldots, K$, and the associated snapshot matrix $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K]$. In the following, the errors of the various approximations will be measured using the *relative RMS error* E_{RMS}, defined as the ratio of the RMS error to the RMS norm of the data, where the RMS norm is defined in terms of the Frobenius norm as

$$\|\cdot\|_{\mathrm{RMS}} = \|\cdot\|_{\mathrm{Fro}}/\sqrt{J\,K}. \tag{8}$$

For instance, invoking (5), for the truncated approximation (4), we have

$$E_{\mathrm{RMS}} := \frac{\|\boldsymbol{S} - \boldsymbol{S}^{\mathrm{trunc}}\|_{\mathrm{RMS}}}{\|\boldsymbol{S}\|_{\mathrm{RMS}}} = \frac{\|\boldsymbol{S} - \boldsymbol{S}^{\mathrm{trunc}}\|_{\mathrm{Fro}}}{\|\boldsymbol{S}\|_{\mathrm{Fro}}} = \frac{\sqrt{\sigma_{M+1}^2 + \ldots + \sigma_R^2}}{\sqrt{\sigma_1^2 + \ldots + \sigma_R^2}}. \tag{9}$$

Similarly, the *relative maximum error* is defined as

$$E_{\max} := \frac{\|\boldsymbol{S} - \boldsymbol{S}^{\mathrm{trunc}}\|_{\infty}}{\|\boldsymbol{S}\|_{\mathrm{RMS}}}. \tag{10}$$

2.1. The method

A set of collocation points, $j_1, \ldots, j_N$, and a set of snapshots, $k_1, \ldots, k_N$, are selected by applying Gaussian elimination to the snapshot matrix $\boldsymbol{S}$ in 4 steps, as follows:

1. That element, $S_{j_1k_1}$, of the matrix $\boldsymbol{S} = [S_{jk}]$ exhibiting the largest absolute value is identified. Using $S_{j_1k_1}$ as pivot, convenient linear combinations of the columns of $\boldsymbol{S}$ are performed to set to zero the remaining elements of the j_1-th row of $\boldsymbol{S}$.
2. The indexes j_1 and k_1 define the *first collocation point* and the *first selected snapshot*, respectively. The k_1-th snapshot is removed from the matrix $\boldsymbol{S}$, obtaining a first modified snapshot matrix $\boldsymbol{S}_1$. Note that the j_1-th row of $\boldsymbol{S}_1$ identically vanishes.
3. Steps 1 and 2 are applied to the matrix $\boldsymbol{S}_1$, which defines a second collocation point, $j_2 \neq j_1$, and a second snapshot, $k_2 \neq k_1$, which is removed from $\boldsymbol{S}_1$, yielding a new modified matrix $\boldsymbol{S}_2$ whose j_1-th and j_2-th rows identically vanish.
4. The procedure iteratively continues, yielding new collocation points, $j_3, \ldots$, new snapshots, $k_3, \ldots$, and new modified matrices, $\boldsymbol{S}_3, \ldots$, until the first iteration step at which

$$\|\boldsymbol{S}_N\|_{\text{Fro}} / \|\boldsymbol{S}\|_{\text{Fro}} < \varepsilon_N, \tag{11}$$

for some tunable small threshold ε_N, which must be somewhat small compared to the tolerance allowed in the approximations below. This defines the (common) number of collocation points and selected snapshots, N.

Condition (11) is consistent with (9) and means that the *relative RMS error* of the approximation of the non-selected snapshots as linear combinations of the selected snapshots is smaller than ε_N. Steps 1–4 define an *incomplete LU decomposition*, in which only the first N steps of the standard LU decomposition with *double pivoting* are accomplished.

In order to avoid misleading interpretations of the good functioning of the method, it is important to note here that, as defined, the collocation points are related to extrema of the modified snapshots, which generally have nothing to do with the extrema of the original snapshots. Instead, the extrema of the modified snapshots occur at those spatial regions where the original snapshots are more uncorrelated (namely, less linearly dependent) among each other. This is obvious noting that (as further recalled in the next subsection) Gauss elimination with double pivoting somehow maximizes linear independency, in connection with the selection of both the retained snapshots and the pivots (namely, the collocation points). This important issue will be illustrated and further commented in §3.1.

Once the sets of collocation points and selected snapshots have been identified, truncated POD is applied to the set of selected snapshots, using the inner product (7), based on the collocation points. This is performed by applying standard SVD to the *reduced snapshot matrix* $\widehat{\boldsymbol{S}}$, which is defined by considering in the original snapshot matrix $\boldsymbol{S}$ only those rows and columns corresponding to the collocation points and the selected snapshots, respectively. It follows that

$$\widehat{\boldsymbol{S}} = \widehat{\boldsymbol{U}}\,\widehat{\boldsymbol{\Sigma}}\,\overline{\widehat{\boldsymbol{V}}}^{\top}, \quad \text{with } \overline{\widehat{\boldsymbol{V}}}^{\top}\widehat{\boldsymbol{V}} = \overline{\widehat{\boldsymbol{U}}}^{\top}\widehat{\boldsymbol{U}} = M \times M\text{-unit matrix}, \tag{12}$$

where for convenience we are recalling that the columns of the matrices $\widehat{\boldsymbol{U}}$ and $\widehat{\boldsymbol{V}}$ are orthonormal. The number of retained modes, M, is defined in terms of a tunable threshold ε_M, as

$$\frac{\sqrt{\widehat{\sigma}_{M+1}^2 + \ldots + \widehat{\sigma}_R^2}}{\sqrt{\widehat{\sigma}_1^2 + \ldots + \widehat{\sigma}_R^2}} < \varepsilon_M, \tag{13}$$

where, invoking (9), the left-hand side is the relative RMS error of the reconstruction of the reduced snapshot matrix retaining M modes.

In order to recover the POD modes at all mesh points, we first note that, using (12), the reduced mode matrix $\widehat{\boldsymbol{U}}$ can also be written as

$$\widehat{\boldsymbol{U}} = \widehat{\boldsymbol{S}}\,\widehat{\boldsymbol{V}}\,\widehat{\boldsymbol{\Sigma}}^{-1}. \tag{14}$$

This formula produces slightly non-orthonormal reduced modes due to round-off errors when very small singular values are retained, which makes the matrix $\widehat{\boldsymbol{\Sigma}}$ ill-conditioned. Thus, the reduced modes are re-orthonormalized using the QR decomposition, as $\widehat{\boldsymbol{U}} = \widehat{\boldsymbol{Q}}\,\widehat{\boldsymbol{R}}$, which corrects (14) as

$$\widehat{\boldsymbol{U}} = \widehat{\boldsymbol{S}}\,\widehat{\boldsymbol{P}}, \tag{15}$$

with the $N \times M$-projection matrix $\widehat{\boldsymbol{P}} = \widehat{\boldsymbol{V}}\,\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{R}}^{-1}$. Now, taking into account the meaning of the various matrices appearing in (15), this equation yields the components of the POD modes corresponding to the collocation points (namely, the columns of $\widehat{\boldsymbol{U}}$) as *linear combinations* of their counterparts for the selected snapshots (namely, the columns of $\widehat{\boldsymbol{S}}$). Assuming that the remaining components of the modes (not corresponding to the collocation points) are slaved to the 'collocation points components', the modes are extended to all spatial points by using the same linear combinations of snapshots appearing in (15). In other words, the full mode matrix $\boldsymbol{U}$, which is a $J \times M$-matrix, is calculated as

$$\boldsymbol{U} = \widehat{\boldsymbol{S}}^{\text{SR}}\,\widehat{\boldsymbol{P}}, \tag{16}$$

where the semi-reduced snapshot matrix $\widehat{\boldsymbol{S}}^{\text{SR}}$ is the $J \times N$-matrix that is obtained from the full snapshot matrix $\boldsymbol{S}$ by considering only the N selected snapshots (but all mesh points). By construction, the columns of $\widehat{\boldsymbol{S}}^{\text{SR}}$ are orthonormal with

the inner product (7). Note how simply the reduced modes (which only contain the values of the modes at the collocation points) have been extrapolated to the full modes (which contain information at all mesh points). The same idea is used to calculate the spatial derivatives of the modes in terms of the derivatives of the snapshots: if $\mathcal{D}$ is a *derivative matrix*, giving a specific derivative of the snapshots in the whole computational mesh as $\mathcal{D}\boldsymbol{S}$, the corresponding derivative of the modes is calculated as

$$\mathcal{D}\boldsymbol{U} = (\mathcal{D}\widehat{\boldsymbol{S}}^{\mathrm{SR}})\,\widehat{\boldsymbol{P}}, \tag{17}$$

where the semi-reduced snapshot matrix $\widehat{\boldsymbol{S}}^{\mathrm{SR}}$ is as defined after (16). Again, this method is much simpler than finite differences, which would require reconstructions at some neighbors of the collocation points. More general differential operators, such as the Laplacian, are similarly treated: eq. (17) still applies if $\mathcal{D}$ is the matrix associated with the differential operator.

2.2. Justification of the LUPOD method

After performing the steps 1–4 described in §2.1 and appropriately re-ordering both the columns and rows of the $J \times K$-snapshot matrix $\boldsymbol{S}$, the new snapshot matrix, denoted again as $\boldsymbol{S}$, is such that $\boldsymbol{S}^\top$ has been (incompletely) LU-decomposed as

$$\boldsymbol{S}^\top \equiv \begin{bmatrix} \boldsymbol{S}_1^\top \\ \boldsymbol{S}_2^\top \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{12} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{D}\boldsymbol{U}_1 \\ d_{N+1}\boldsymbol{U}_2 \end{bmatrix} \equiv \mathbf{L}\boldsymbol{U}, \tag{18}$$

where the rows of the $N \times J$-matrix $\boldsymbol{S}_1^\top$ and the $(K-N) \times J$-matrix $\boldsymbol{S}_2^\top$ are the selected snapshots and the remaining snapshots, respectively, and the blocks appearing in the $\mathbf{L}$ and $\boldsymbol{U}$ factors of the decomposition are as follows. The $N \times N$-matrix $\mathbf{L}_{11}$ is well conditioned and lower triangular, with its diagonal elements all equal to one, $\mathbf{L}_{12}$ exhibits bounded norm, and $\mathbf{0}$ and $\mathbf{I}$ are the zero and unit matrices of appropriate order; $\mathbf{D}$ is the $N \times N$-diagonal matrix whose N diagonal elements are the pivots used in the performed incomplete LU decomposition, $\boldsymbol{U}_1$ is upper triangular, with the diagonal elements equal to one and its maximum norm equal to one, namely

$$\boldsymbol{U}_1 = [\boldsymbol{U}_{11}\,|\,\boldsymbol{U}_{12}] = \left[\begin{array}{ccccc|ccc} 1 & u_{12} & u_{13} & \dots & u_{1N} & u_{1,N+1} & \dots & u_{1J} \\ 0 & 1 & u_{23} & \dots & u_{2N} & u_{2,N+1} & \dots & u_{2J} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & u_{N,N+1} & \dots & u_{NJ} \end{array}\right], \tag{19}$$

with $|u_{jk}| \le 1$; d_{N+1} is the pivot in the $(N+1)$-th LU-step (which is not performed), and the $(K-N) \times J$-matrix $\boldsymbol{U}_2$ is such that its maximum norm equals one.

Now, according to (11), the pivot d_{N+1} is negligible in the present context. If the associated term in (18) is neglected, we obtain

$$\boldsymbol{S}_1^\top = \mathbf{L}_{11}\mathbf{D}\boldsymbol{U}_1, \qquad \boldsymbol{S}_2^\top \simeq \mathbf{L}_{12}\mathbf{D}\boldsymbol{U}_1 = \mathbf{L}_{12}\mathbf{L}_{11}^{-1}\boldsymbol{S}_1^\top, \tag{20}$$

where the last equality in the second equation is obtained by solving the first equation for $\mathbf{D}\boldsymbol{U}_1$ and substituting into the second equation. These two equations show that:

- Using well-known results on perturbation of the spectra of symmetric matrices [12,27], the N largest singular values of the snapshot matrix $\boldsymbol{S} = [\boldsymbol{S}_1|\boldsymbol{S}_2]$ can be identified (to the approximation relevant here) with the N singular values of $\boldsymbol{S}_1$, and the remaining singular values can be neglected.
- Since $\mathbf{L}_{11}$ is well conditioned, the neglected snapshots (namely, the rows of $\boldsymbol{S}_2^\top$) are well approximated through the second equation (20) in terms of the retained snapshots (namely, the rows of $\boldsymbol{S}_1^\top$). Thus, the neglected snapshots are slaved to the retained ones and retaining these snapshots is justified.
- If a good reconstruction (by whatever means, such as a POD expansion in terms of orthonormal POD modes) of the rows of $\mathbf{D}\boldsymbol{U}_1$ is available, then the first equation in (20) gives an approximate reconstruction of the rows of $\boldsymbol{S}_1^\top$ (i.e., the retained snapshots). On the other hand, the shape of the matrix $\boldsymbol{U}_1$ displayed in (19) indicates that a good reconstruction of the rows of $\boldsymbol{U}_1$ readily follows using information from only the first N columns, which correspond to the selected collocation points, both in $\mathbf{D}\boldsymbol{U}_1$ and in $\boldsymbol{S}_1^\top$. In other words, the collocation points can be used as master points to perform POD, with the remaining points slaved to them. This is the idea behind using incomplete LU decomposition to efficiently solve linear systems [5].
- The argument in the last item will be checked in the remaining of the paper using a variety of applications. Moreover, this argument has been checked in several additional tests, not included in the paper for the sake of brevity. In fact, in these tests we have repeatedly obtained that the singular values of the matrices $\boldsymbol{S}_1^\top$, $\boldsymbol{S}_{11}^\top$ (namely, the matrix that contains the first N columns of $\boldsymbol{S}_1^\top$, which corresponds to the reduced snapshot matrix defined in the last subsection), $\mathbf{D}\boldsymbol{U}_1$, and $\mathbf{D}\boldsymbol{U}_{11}$ all 'scale' with the diagonal elements of $\mathbf{D}$ (i.e., the pivots, $d_1, \dots, d_N$, in the incomplete LU decomposition performed above). By 'similar scaling' we mean that these sets of singular values give approximately parallel

patterns in the usual semi-logarithmic plot of the singular values σ_n vs. n. In other words, for a given (small) accuracy, the truncated SVD of these matrices requires approximately the same number of modes. This property is quite natural in view of the shape of the several involved matrices. However, proving this property is well beyond the scope of this paper. It must be noted that relating the singular values of the product of two matrices to the singular values of the factors is a very subtle issue ([15] and references therein).

Summarizing the above, the POD modes and singular values of the reduced snapshot matrix, which only bears information of the selected snapshots at the selected collocation points, are good means to reconstruct the whole snapshot matrix. This is the requirement for the good functioning of the method as a data processing tool, as in the applications that will be considered in §3. Also, in the end, this is the requirement for the good functioning of the method in constructing ROMs (the object of §4 below) via POD applied to a representative set of snapshots. By representative snapshots we mean that the whole dynamics of the underlying system is approximately contained in the vector space spanned by the snapshots.

2.3. Some remarks on the LUPOD method

Some remarks are now in order:

1. *Maximum norm vs. Frobenius norm.* The numerator in (11) could be replaced by the maximum norm of $\boldsymbol{S}_N$, which improves the quality of the resulting collocation points/selected snapshots, but decreases robustness against noise and increases N. However, in all applications below, using the maximum norm in the numerator in (11) gives essentially the same results.
2. *Tunable parameters.* As described in §2.1, the LUPOD method involves the tunable parameters ε_N and ε_M, appearing in (11) and (13), respectively, which define the number of collocation points/selected snapshots, N, and the number of retained modes, M, respectively. These parameters must be selected as appropriately small. For simplicity, they will be selected such that $M = N$ in some of the applications below, but the robustness of the method will be increased in other applications by selecting $N > M$.
3. *Noise.* Noisy snapshots are to be expected in industrial applications. Noise could be dealt with in the LUPOD method by, e.g., increasing the number N of collocation points and, moreover, selecting an even larger number of snapshots. However, for the sake of simplicity, noisy snapshots are assumed to have been treated with a convenient data-processing tool before applying the LUPOD method. The level of noise is thus small and noise is filtered by just selecting N somewhat larger than M.
4. *Simultaneous approximation of various snapshot sets.* In some cases, it is convenient that the collocation points be shared in the POD approximation of various snapshot sets. For instance, in the reduced order modeling for partial differential equations, it may be convenient that the POD modes give good approximations not only of the snapshots, but also of some of their partial derivatives, collected as derivative snapshots, $\boldsymbol{s}_k^{\mathrm{D}}$. In the simultaneous approximation of the snapshot and derivative snapshot matrices, $\boldsymbol{S}$ and $\boldsymbol{S}^{\mathrm{D}}$, respectively, the standard LUPOD method is applied to the *enlarged* $2J \times K$*-snapshot matrix*

$$\widetilde{\boldsymbol{S}} = \sqrt{2JK}\begin{bmatrix} w\boldsymbol{S}/\|\boldsymbol{S}\|_{\mathrm{Fro}} \\ (1-w)\boldsymbol{S}^{\mathrm{D}}/\|\boldsymbol{S}^{\mathrm{D}}\|_{\mathrm{Fro}} \end{bmatrix}, \tag{21}$$

for some tunable weight w, such that $0 < w < 1$, which balances the relative importance of the snapshots and derivative snapshots. Regarding the applications in §4, results turned out to slightly improve after calibrating the value of w for some specific cases. However, for the sake of robustness and generality, $w = 0.5$ has been set as a reasonable value for all test problems. On the other hand, a larger number of derivatives could be similarly treated.
Note that the collocation points, distributed along the rows associated with the first and second groups of J rows of $\widetilde{\boldsymbol{S}}$ in (21), all correspond to the same spatial mesh, meaning that each selected collocation point may appear twice upon application of LUPOD to $\widetilde{\boldsymbol{S}}$. Repeated points, if any, may be treated by either considering these points just once or using a weighted inner product, assigning the weights as proportional to the number of times each collocation point is repeated. Nevertheless, it is worth remarking that no repetitions occurred in the applications below.

3. Illustration of the method

To begin with, the selection of the LUPOD collocation points is compared in §3.1 with the distribution of standard collocation points in classical collocation methods. In addition, although, as anticipated, this is not the main goal of the method, the LUPOD performance when applied to snapshot sets exhibiting concentrated complexity is illustrated in §3.2. By regions with concentrated complexity we mean hereinafter spatial regions where the snapshots are poorly correlated, which for smooth functions occurs in regions where some of the snapshots show steep gradients, or large higher order derivatives. The simplest localized complexity, which is easily visualized (and thus used in §3.2 to illustrate the method), consists in localized maxima or minima, which obviously involve steep gradients nearby. However, localized steep higher order derivatives (which cannot be generally visualized by the naked eye in plots of the snapshots themselves) also produce

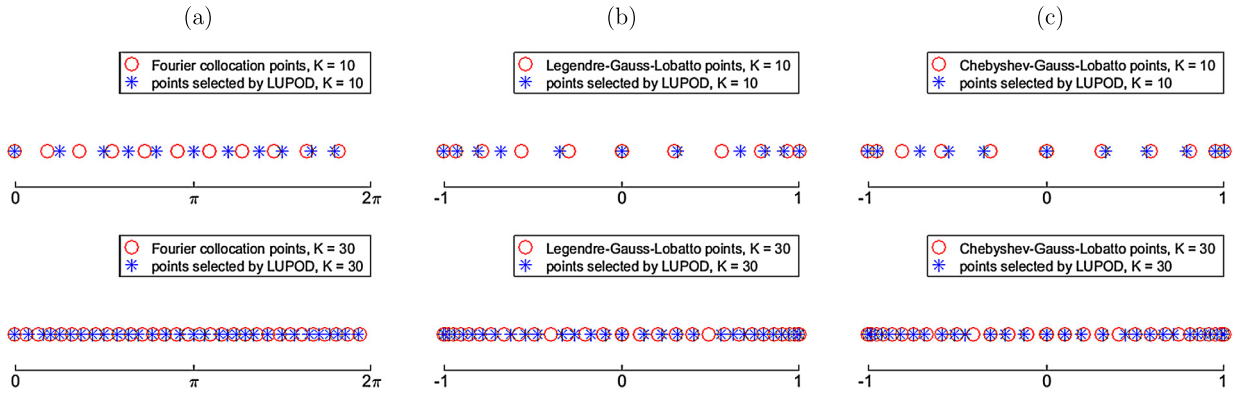

Fig. 1. Comparison of classical Fourier (a), Legendre (b), and Chebyshev (c) collocation points with the LUPOD collocation points.

snapshots that are poorly correlated locally and, thus, produce concentrated complexity; this is illustrated at the end of §3.1 considering snapshots whose concentrated complexity is not associated just with local maxima or minima of any of the snapshots. Finally, the method is applied in §3.3 to a less academic set of snapshots, resulting from computational fluid dynamics (CFD).

3.1. Comparison with standard Fourier, Legendre and Chebyshev collocation

The very simple and general collocation strategy of LUPOD is tested against classical collocation methods. To begin with, we consider the first $K+1$ *trigonometric functions* on the interval $[0, 2\pi]$, $F_k(x) = \cos(kx + k\pi/2)$ for $k = 0, \ldots, K$, and the associated uniformly distributed Fourier collocation points, $x_k = 2\pi k/(K+1)$ for $k = 0, \ldots, K$. The LUPOD method is applied to the matrix $\boldsymbol{S}^{\mathrm{F}}$, whose columns are the values of these functions at the uniformly distributed points $x^j = 2\pi j/2001$, with $j = 0, \ldots, 2000 = J$. As can be seen in Fig. 1(a) the LUPOD selected points are distributed more or less equispacedly, similarly to the Fourier collocation points.

Standard *Gauss–Lobatto* collocation points for the Legendre and Chebyshev polynomials on the interval $[-1, 1]$, $L_k(x)$ and $T_k(x)$ for $k = 0, \ldots, K$, respectively, occur at the local extrema of the last polynomial $L_K(x)$ or $T_K(x)$, respectively [7], which concentrate near the end-points of the interval. Applying the LUPOD method to the matrices $\boldsymbol{S}^{\mathrm{L}}$ and $\boldsymbol{S}^{\mathrm{T}}$, whose columns are the values of these polynomials (for $K = 10$ and 30) at the equispaced points $x^j = j/1000 - 1$, with $j = 0, \ldots, 2000$, yields the LUPOD collocation points. These are seen in Fig. 1(b,c) to show similar patterns as the Gauss–Lobatto collocation points, namely they concentrate near the end-walls of the interval for both Legendre and Chebyshev polynomials. In this plot, the LUPOD points appear to be somewhat close to the Gauss–Lobatto points, which are related to the extrema of the snapshots (in fact, they are the extrema of the last snapsphot). Thus, this example might (wrongly) suggest that the LUPOD collocation points are related to the extrema of the selected snapshots. Instead, as anticipated, they concentrate at those regions where the snapshots are more uncorrelated, which is generally associated with large values of either the snapshots or their higher order derivatives. To illustrate this, we consider the spatio-temporal behavior defined in the square region $-1 \le x \le 1, 0 \le t \le 1$, in terms of the Chebyshev polynomials as

$$s(x,t) = (1-x^2)\sum_{p=0}^{P}(p+1)^{-4}\sin[3(p+2)\sqrt{p+1}\,(t+\pi/4)]\,T_p(x). \tag{22}$$

As above, we discretize this function using 2001 equispaced points in the interval $-1 \le x \le 1$, and consider 1001 equispaced snapshots in the interval $0 \le t \le 1$. Noting that the l-th derivative of the p-th order Chebyshev polynomial behaves near the end-points of the interval as p^{2l} and that the coefficient of T_p in (22) is $\sim p^{-4}$, the third and higher order derivatives all peak near the end-points, which is where complexity (as defined above) concentrates.

Fig. 2(a,b) shows the snapshots for $P = 10$ and 30; the crosses are the collocation points, which also indicate the selected snapshots (namely, the vertical lines passing through the crosses). Note in these two plots that the snapshots themselves do not show a localized complexity near the end-points. In fact, as more clearly seen in plot (c), the snapshots are qualitatively similar among each other and all peak near the center of the interval. The LUPOD collocation points, instead, plotted in Fig. 3 for $P = 10$ and 30, do capture very well the above mentioned localized complexity associated with higher order derivatives since they concentrate near the end-points, as it happened with their counterparts in Fig. 1(c).

Summarizing, the examples considered in Fig. 1 evidence how the LUPOD method applied to classical Fourier and orthogonal polynomials, without performing any ad hoc theoretical study, is able to capture distributions of points that are comparable to classical collocation points widely used in the literature for different numerical tasks. This is a value in addition to the properties highlighted below. On the other hand, the example considered in Figs. 2 and 3 illustrates how concentrated complexity (which promotes spatial concentration of the LUPOD collocation points) is related with steep behavior of either the snapshots, their gradients, or their higher order derivatives. Thus, this concentrated complexity is a

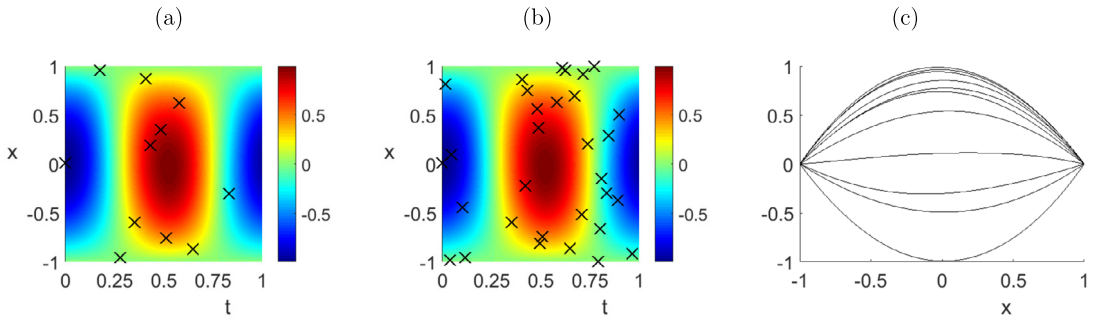

Fig. 2. (a,b) Spatio-temporal color maps of the snapshots (22), which are the vertical sections of this plot, for (a) $P = 10$ and (b) $P = 30$, indicating with crosses the LUPOD collocation points. (c) The selected snapshots in the case $P = 10$.

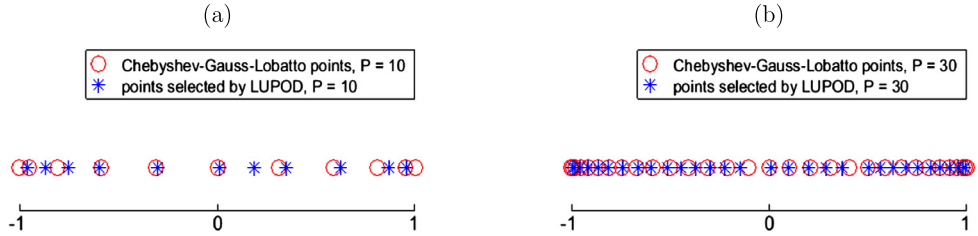

Fig. 3. Counterpart of Fig. 1 for the snapshots defined in eq. (22) with (a) $P = 10$ and (b) $P = 30$.

subtle concept that is not easily appreciated in plots of the snapshots themselves. However, in order to facilitate clear plots for the reader, the type of concentrated complexity that will be considered in the next subsection will always be associated with localized peaks in the snapshots.

3.2. Some toy model snapshot matrices with concentrated complexity

Let us now consider two academic toy model snapshot matrices with concentrated complexity. The first of these is obtained as

$$\boldsymbol{S}^1 = \boldsymbol{T}^1 + \boldsymbol{T}^2, \tag{23}$$

where the 250×250-matrices $\boldsymbol{T}^1$ and $\boldsymbol{T}^2$ are given by

$$T^1_{jk} = \phi_{\{35,165,4,12\}}(j,k) + \phi_{\{85,225,20,5\}}(j,k) + \phi_{\{125,100,9,6\}}(j,k) + \phi_{\{160,35,7,7\}}(j,k) + \phi_{\{210,85,6,25\}}(j,k), \tag{24}$$

$$T^2_{jk} = \frac{1}{30}\left[\psi(j,k)^{1/2} - \psi(j,k)^2 - 5\right], \tag{25}$$

in terms of the following functions, defined in the square $1 \le x, y \le 250$,

$$\phi_{\{x_0,y_0,a,b\}}(x,y) = -\frac{1}{2}\exp\left[-(x-x_0)^2/a^2 - (y-y_0)^2/b^2\right], \tag{26}$$

$$\psi(x,y) = \log\left[7 + \log\left(1 + g(x,y)^2 + g(x,y)^3\right)\right] - \sin\left[1 + g(x,y)^{3/2}\right], \tag{27}$$

with

$$g(x,y) = 2 + \sin[h_2(x,y)] + \cos^2[h_2(x,y)] + \cos\frac{h_1(x)}{100}\sin\frac{h_1(y)}{25}, \tag{28}$$

$$h_1(x) = \frac{2x+3}{5}, \quad h_2(x,y) = \frac{h_1(x)}{60} + \frac{h_1(y)}{50}. \tag{29}$$

The matrices $\boldsymbol{T}^1$ and $\boldsymbol{T}^2$ and the toy model snapshot matrix $\boldsymbol{S}^1$ are plotted in Fig. 4. Note that $\boldsymbol{T}^1$ is localized in five elliptic regions and $\boldsymbol{T}^2$ (which is defined through a fairly complex combination/composition of transcendental functions, see (25), (27)–(29)) is spread along the domain. As above, the columns of $\boldsymbol{S}^1$ mimic the snapshots and the rows, the grid points. The first 5, 10, and 15 collocation points/snapshots selected by LUPOD are given in Fig. 5. As can be seen, the first five collocation points/selected snapshots are located precisely in those localized regions defined by the matrix $\boldsymbol{T}^1$, while subsequent points/snapshots are dominated by the spread complexity associated with $\boldsymbol{T}^2$.

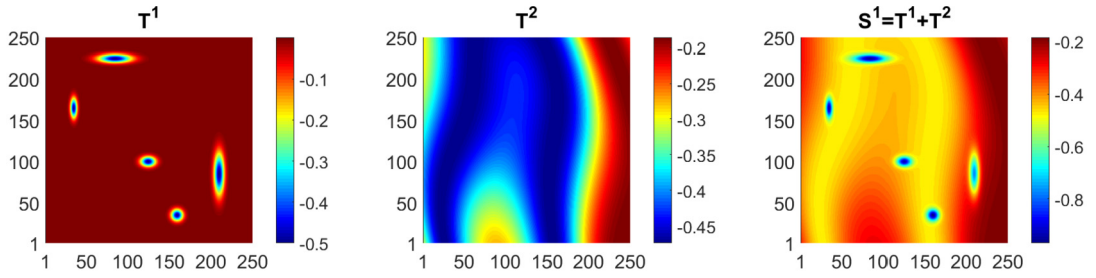

Fig. 4. The matrices $\boldsymbol{T}^1$ (left) and $\boldsymbol{T}^2$ (middle) defining the toy model snapshot matrix $\boldsymbol{S}^1$ (right) using (23).

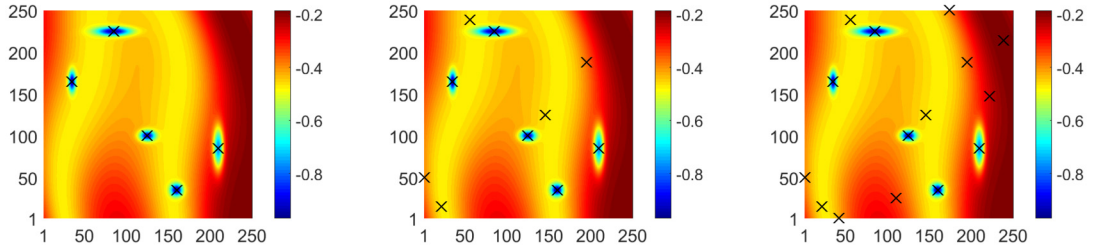

Fig. 5. Toy model snapshot matrix $\boldsymbol{S}^1$: location of the first 5 (left), 10 (middle), and 15 (right) collocation points/snapshots selected by LUPOD.

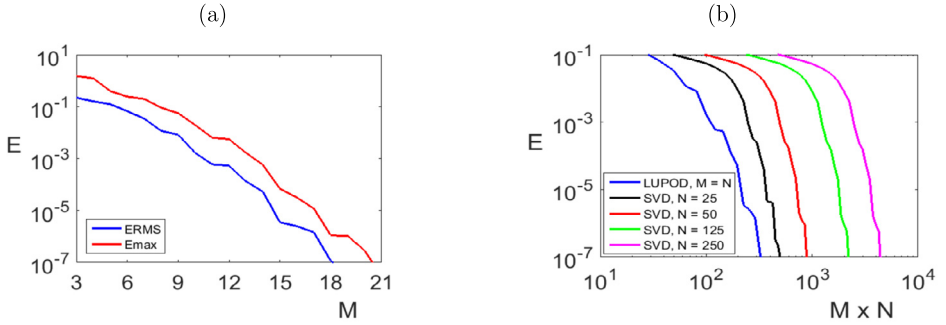

Fig. 6. Toy model snapshot matrix $\boldsymbol{S}^1$. (a) LUPOD-reconstruction error vs. the number of collocation points/selected snapshots/retained modes, $M = N$. (b) Comparison of the E_{RMS} error using LUPOD with $M = N$ (blue) and using SVD based on N equispaced collocation points (remaining curves) vs. MN.

The accuracy of the LUPOD reconstruction using an increasing number N of collocation points/selected snapshots and retaining $M = N$ POD modes is considered in Fig. 6(a), where the reconstruction RMS and maximum errors, defined in (9)–(10), are plotted vs. $M = N$. As can be seen, the errors decrease spectrally and reach very small values for moderate values of $M = N$. Fig. 6(b) shows a comparison of LUPOD with standard SVD using simpler, equispaced distributions of N collocation points retaining an increasing number of POD modes $M \leq N$. In order to get a fair comparison, the E_{RMS} error is plotted vs. MN, which is the computational complexity when using N collocation points and M modes to project the governing equations in POD-based ROMs. As can be seen, LUPOD yields a much smaller error than SVD with an equispaced distribution of points.

As a second example, the following toy model snapshot matrices,

$$\boldsymbol{S}^2 = \boldsymbol{T}^2 + \boldsymbol{T}^3 \quad \text{and} \quad \boldsymbol{S}^3 = \boldsymbol{T}^2 + \boldsymbol{T}^3 + \boldsymbol{T}^4, \tag{30}$$

are considered, with the matrix $\boldsymbol{T}^2$ as defined in (25), plotted in Fig. 4, while $\boldsymbol{T}^3$ and $\boldsymbol{T}^4$ are defined as

$$T^3_{jk} = -\frac{1}{6}\left[\chi_1(j,k) + \chi_2(j,k) + \chi_3(j,k)\right], \tag{31}$$

$$T^4_{jk} = \frac{1}{600}\sum_{m=1}^{15} \sin(2j\widetilde{r}_m)\cos(3k\widetilde{r}_{m+15}/2), \tag{32}$$

where χ_1, χ_2, and χ_3 are the characteristic functions of the squares of side 2, centered at the points

$$p_1 = (63, 57), \quad p_2 = (165, 176), \quad \text{and} \quad p_3 = (186, 36), \tag{33}$$

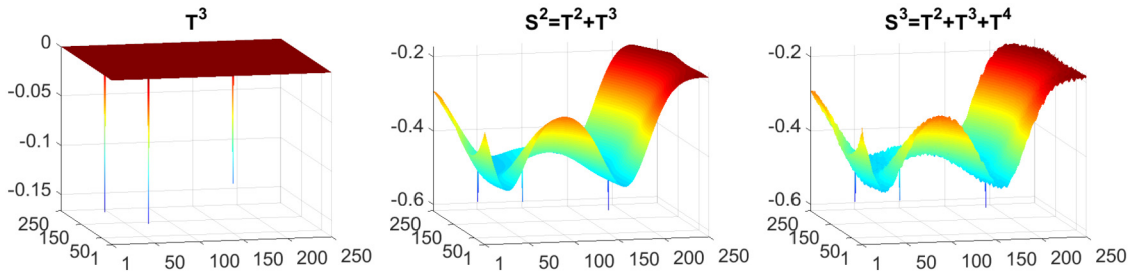

Fig. 7. The matrices $\boldsymbol{T}^3$ (left), $\boldsymbol{S}^2$ (middle), and $\boldsymbol{S}^3$ (right), defined in eqs. (30)–(32).

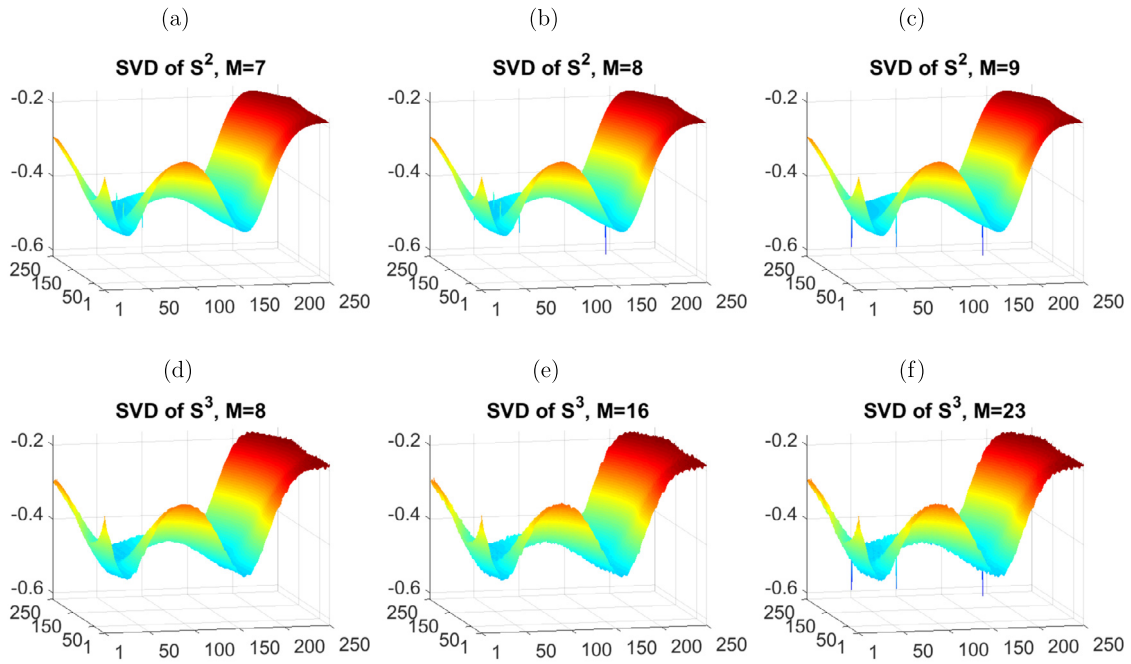

Fig. 8. Application of standard SVD to the toy model matrices $\boldsymbol{S}^2$ (top) and $\boldsymbol{S}^3$ (bottom), defined in (30), retaining the indicated number M of modes.

respectively, and $\widetilde{r}_1, \ldots, \widetilde{r}_{30}$ denote 30 uniformly distributed random numbers between 0 and 1 (computed with the MATLAB command 'rand'). The matrices $\boldsymbol{T}^3$, $\boldsymbol{S}^2$, and $\boldsymbol{S}^3$ are plotted in Fig. 7. Note the very sharp localized peaks appearing in both $\boldsymbol{S}^2$ and $\boldsymbol{S}^3$, which are due to the matrix $\boldsymbol{T}^3$.

When applying standard SVD to the matrices $\boldsymbol{S}^2$ and $\boldsymbol{S}^3$, the localized peaks are lost or poorly described, as shown in Fig. 8(a,b,d,e). Only if a sufficiently large number of SVD modes is retained, namely 9 modes for $\boldsymbol{S}^2$ and 23 modes for $\boldsymbol{S}^3$, such peaks can be captured as seen in Fig. 8(c,f). This is because, as anticipated in §1, the peaks exhibit very small energy (defined by the $\|\cdot\|_{\text{Fro}}$-norm), and thus they are masked in $\boldsymbol{S}^2$ by the spread modes associated with the matrix $\boldsymbol{T}^2$. The situation is even worse in the noisy toy model matrix $\boldsymbol{S}^3$, in which the pseudo-random noise included in the matrix $\boldsymbol{T}^4$ produces modes that are also spread and thus contribute to mask the peaks even further. Namely, 16 modes are not enough to localize the peaks of $\boldsymbol{S}^3$, as observed in Fig. 8(e), while 9 modes suffice to properly describe them for the noise-free matrix $\boldsymbol{S}^2$, see Fig. 8(c).

The LUPOD method, instead, locates collocation points near the peaks and is able to identify the associated localized complexity retaining a smaller number of modes, as seen in Fig. 9. Namely, 7 and 8 modes are enough for LUPOD to capture the peaks in $\boldsymbol{S}^2$ and $\boldsymbol{S}^3$, respectively, while as already observed standard SVD requires 9 and 23 modes, respectively.

3.3. Application to an aerodynamic database

Let us now illustrate the method in a less academic example, in which the snapshots are defined in a two-dimensional spatial mesh. We consider the (two-dimensional) pressure distribution on the surface of a swept wing for 13 equispaced values of the angle of attack AoA, in the range $-3° \leq \text{AoA} \leq 3°$, as calculated using an industrial CFD code; see [1] for further details. The spatial mesh is structured, mounted on a curvilinear coordinate system on the wing surface, which contains 225 and 75 points along the chord and span directions, respectively. The chord coordinate varies from the trailing

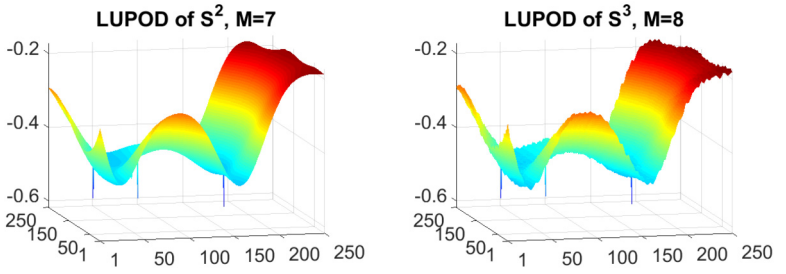

Fig. 9. LUPOD reconstruction of the matrices $\boldsymbol{S}^2$ (left) and $\boldsymbol{S}^3$ (right) retaining the indicated number M of modes.

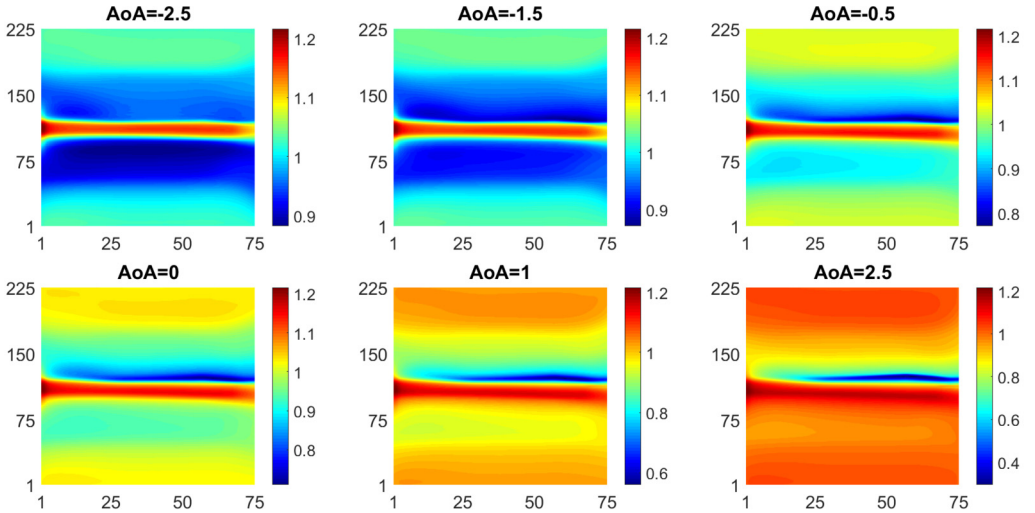

Fig. 10. Six representative snapshots in the aerodynamic database, associated with different values of the angle of attack AoA.

edge at the pressure side to the trailing edge at the suction side, concentrating points near the leading edge, where the pressure distribution is quite steep, while the span coordinate varies from the root section towards the wing tip; see Fig. 10. The values of the pressure coefficient could be organized in a $225 \times 75 \times 13$-third-order tensor, defined as

$$P_{j_1 j_2 k}, \tag{34}$$

where the indexes j_1, j_2, and k are associated with the chord, the span, and the angle of attack, respectively. However, the first two indexes are unfolded into a single index j, which varies in the range $1 \leq j \leq 225 \times 75$, obtaining a snapshot matrix $\boldsymbol{S}$ whose 13 columns (one for each value of the angle of attack) give the snapshots as vector entities.

Applying the LUPOD method to the aerodynamic snapshot matrix $\boldsymbol{S}$, retaining an increasing number of collocation points/selected snapshots and modes, $M = N$, yields the results that are summarized in Fig. 11. As can be seen in the upper plots, LUPOD tends to locate the collocation points along the leading edge ($90 \leq j_1 \leq 130$), and tends to select them near the wing tip ($j_2 = 75$), where the pressure distribution is steepest, as anticipated. Concerning the lower plots:

- Since $M = N$, LUPOD retains as many POD modes as the number of selected snapshots, meaning that these snapshots are reconstructed exactly. In other words, the selected snapshots can be identified in the lower plots because the errors are equal to the zero-machine.
- The remaining snapshots are very well reconstructed in spite of the fact that calculations are based on a small number of collocation points. For instance, using just five collocation points, the maximum and RMS errors are $\sim 10^{-2}$ and $\sim 10^{-6}$, respectively.

4. Computationally efficient reduced models based on LUPOD

Let us now consider the use of LUPOD in the construction of POD-based ROMs for various infinite dimensional dynamical systems. The application of the method for general systems of partial differential equations is first considered, in §4.1. The general method is then applied to the complex Ginzburg–Landau equation (CGLE), both in 1D (§4.2) and 2D (§4.3).

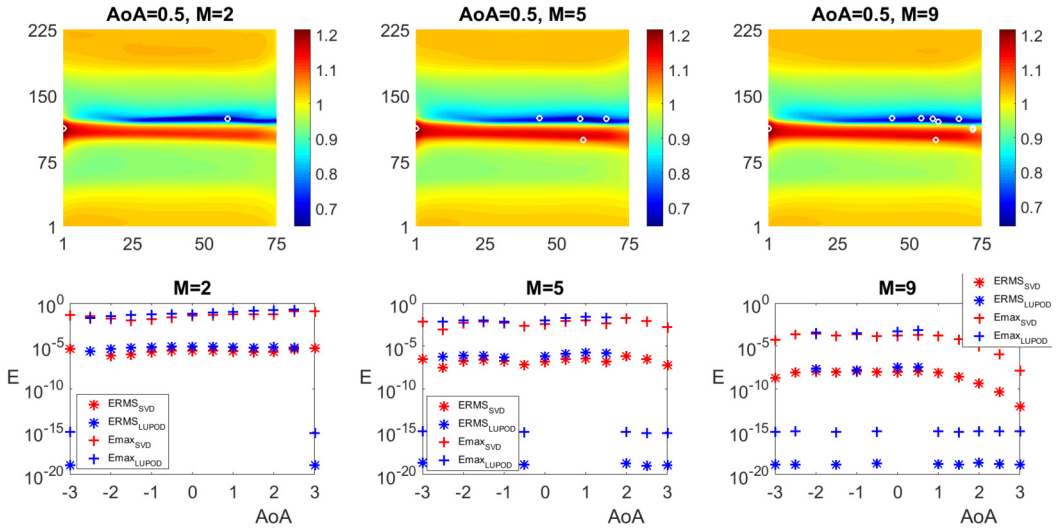

Fig. 11. Top: Reconstruction of the angle of attack AoA $= 0.5°$ snapshot with the indicated number of collocation points/selected snapshots/retained modes, M; collocation points are plotted with white circles. Bottom: RMS and maximum errors, as defined in (9)–(10), in the reconstruction of the 13 snapshots using both LUPOD and standard SVD, as indicated.

4.1. Application to a general system

Let us consider the application to a general system of partial differential equations, considering the evolution problem

$$\partial_t \boldsymbol{u} = \boldsymbol{\mathcal{L}}\boldsymbol{u} + \boldsymbol{f}(\boldsymbol{u}, t), \tag{35}$$

where $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{f}$ are (generally unbounded) linear and nonlinear operators, respectively, such that the operators $\boldsymbol{\mathcal{L}}^{-1}$ and $\boldsymbol{u} \to \boldsymbol{\mathcal{L}}^{-1}\boldsymbol{f}(\boldsymbol{u}, t)$ are both compact [23]. As further discussed in [30], these properties hold, in particular, if the differential operator $\boldsymbol{\mathcal{L}}$ includes the highest-order derivatives and is elliptic, and $\boldsymbol{f}$ satisfies mild assumptions only. Compactness facilitates obtaining flexible low-dimensional descriptions of (35). These assumptions apply to a variety of dissipative systems of scientific and industrial interest resulting from, e.g., pattern formation, reaction–diffusion–convection, aerodynamics, and microfluidics.

The operators $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{f}$ are spatially discretized and denoted also as $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{f}$, respectively. Temporal discretization with a timestep h at $t_p = ph$, with $p = 0, 1, \ldots$, is performed using, e.g., the Crank–Nicolson plus Adams–Bashforth scheme [8], which yields the following numerical scheme

$$\frac{2}{h}(\boldsymbol{u}^{p+1} - \boldsymbol{u}^p) = \boldsymbol{\mathcal{L}}(\boldsymbol{u}^{p+1} + \boldsymbol{u}^p) + 3\boldsymbol{f}(\boldsymbol{u}^p, t_p) - \boldsymbol{f}(\boldsymbol{u}^{p-1}, t_{p-1}), \tag{36}$$

which will be referred to below as the *full model* (FM).

The simplest POD plus Galerkin projection ROM is constructed by first expanding the state variable $\boldsymbol{u}$ as a linear combination of POD modes, as

$$\boldsymbol{u}(\boldsymbol{x}, t) \simeq \boldsymbol{u}^M_{\mathrm{GS}}(\boldsymbol{x}, t) = \sum_{m=1}^{M} a_m(t)\boldsymbol{u}_m(\boldsymbol{x}), \tag{37}$$

substituting this into (36), and projecting the resulting equations onto the set of POD modes, $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$. It follows that the amplitude vector, $\boldsymbol{a} = [a_1, \ldots, a_M]^\top$, is determined by the following ROM (cf. (36))

$$\frac{2}{h}(\boldsymbol{a}^{p+1} - \boldsymbol{a}^p) = \boldsymbol{\mathcal{L}}^{\mathrm{GS}}(\boldsymbol{a}^{p+1} + \boldsymbol{a}^p) + 3\boldsymbol{f}^{\mathrm{GS}}(\boldsymbol{a}^p, t_p) - \boldsymbol{f}^{\mathrm{GS}}(\boldsymbol{a}^{p-1}, t_{p-1}), \tag{38}$$

where the components of the $M \times M$-matrix $\boldsymbol{\mathcal{L}}^{\mathrm{GS}}$ and the M-dimensional function $\boldsymbol{f}^{\mathrm{GS}}$ are given by

$$\mathcal{L}^{\mathrm{GS}}_{ij} = \langle \boldsymbol{u}_i, \boldsymbol{\mathcal{L}}\boldsymbol{u}_j \rangle, \quad f^{\mathrm{GS}}_i = \left\langle \boldsymbol{u}_i, \boldsymbol{f}\left(\sum_{m=1}^{M} a_m \boldsymbol{u}_m, t\right) \right\rangle. \tag{39}$$

Here, the inner product $\langle \cdot, \cdot \rangle$ is precisely that used to calculate the POD modes, namely that defined in (7), based on a limited set of collocation points. The collocation points, in turn, are calculated as explained in §2.3, remark 4, considering the snapshots and derivative snapshots computed from the FM at K values of the temporal index p, as

$$\boldsymbol{s}_k = \boldsymbol{u}^{p_k}, \quad \boldsymbol{s}_k^{\mathrm{D}} = \frac{2}{h}(\boldsymbol{u}^{p_k+1} - \boldsymbol{u}^{p_k}) \quad \text{for } k = 1, \ldots, K. \tag{40}$$

Now, the following remarks are important:

- According to the definition (40), the derivative snapshots $\boldsymbol{s}_k^{\mathrm{D}}$ are the left-hand sides of (36), which approximate the temporal derivatives of the state vector. Note, invoking (36)–(39), that approximating well not only the snapshots but also the derivative snapshots intends to obtain good approximations of both the outcomes of the FM and the left and right sides of (36) (which approximately coincide), precisely what is projected to obtain the ROM (38).
- The weight w appearing in (21) is set to $w = 0.5$ in all simulations below.
- Upon application of the LUPOD strategy to the enlarged snapshot matrix defined in (21) and (40), some collocation points may be selected twice. However, all identified collocation points are considered only once in all applications below.
- In principle, two tunable parameters are involved in the LUPOD method, namely the threshold ε_N for the collocation points/snapshots selection, according to (11), and the threshold ε_M, which defines the number of retained modes using (13). They need to be selected such that $\varepsilon_N \leq \varepsilon_M$ (namely, the number of collocation points/snapshots, N, must be larger than or equal to the number of retained modes, M).
- When data are conveniently correlated (namely, when the singular values appearing in (13) decay conveniently fast), M and N will be very close to each other, and both will be conveniently small if compared to the numbers of mesh points and snapshots, J and K, respectively.

In all simulations below, a preprocessed ROM will be constructed to accelerate the integration of the system (35) along an attractor. To compute the snapshots, the FM (36) will be run over a time interval $0 \leq t \leq T$. After discarding an initial transient in which the attractor is approached, K snapshots and derivative snapshots will be calculated in the time interval

$$T_{\mathrm{attr}} \leq t \leq T, \tag{41}$$

as indicated in (40). The ROM (38) will be constructed using N collocation points and M POD modes, as explained above, and used to approximate the solution of the system in the time interval (41). Consistently with (9)–(10), the *instantaneous relative RMS and maximum errors* of the approximation by the preprocessed ROM are defined as

$$E_{\mathrm{RMS}} := \frac{\|\boldsymbol{u}_{\mathrm{ROM}} - \boldsymbol{u}_{\mathrm{FM}}\|_{\mathrm{RMS}}}{\|\boldsymbol{u}_{\mathrm{FM}}\|_{\mathrm{RMS}}}, \qquad E_{\max} := \frac{\|\boldsymbol{u}_{\mathrm{ROM}} - \boldsymbol{u}_{\mathrm{FM}}\|_{\infty}}{\|\boldsymbol{u}_{\mathrm{FM}}\|_{\mathrm{RMS}}}, \tag{42}$$

respectively. Here, $\boldsymbol{u}_{\mathrm{ROM}}$ and $\boldsymbol{u}_{\mathrm{FM}}$ are the solutions provided by the ROM and the FM, respectively, and $\|\cdot\|_{\mathrm{RMS}}$ and $\|\cdot\|_{\infty}$ are the root mean square ($\|\cdot\|_{\mathrm{RMS}} = \|\cdot\|_2/\sqrt{J}$) and maximum norms, respectively, based on all mesh points used in the discretization of the FM.

As further explained in [22], if these errors were plotted for increasing t in a large time interval, they would increase unboundedly even for the simplest periodic attractors, due to small errors in the fundamental frequency. For more complex, quasi-periodic attractors the situation is similar and for chaotic attractors is even worse, due to divergence of nearby orbits. Thus, the intention cannot be approximating the particular time dependent solutions, but the underlying dynamical system. This is done by ensuring that the instantaneous errors (42) remain small only in time intervals of length T_0 somewhat larger (say, five times as large) than the characteristic timescale of the system. Specifically, T_0 is defined as

$$T_0 = \frac{5}{J}\sum_{j=1}^{J} T_0^j, \quad \text{with } T_0^j = 2\pi\sqrt{\frac{\int_{T_{\mathrm{attr}}}^{T} |\boldsymbol{u}(x_j,t)|^2 dt}{\int_{T_{\mathrm{attr}}}^{T} |\partial_t \boldsymbol{u}(x_j,t)|^2 dt}}. \tag{43}$$

Here, T_0^j is a measure of the timescale at the j-th mesh point, meaning that T_0 is five times the average of T_0^j in the computational grid. Thus, eq. (43) selects T_0 for each particular system (e.g., the CGLE in 1D and 2D in the next subsections) and each particular dynamics of the system. For further details about the calibration of T_0, we refer to [31,22].

4.2. Application to the CGLE in 1D

Let us consider the one-dimensional complex Ginzburg–Landau equation (CGLE), with homogeneous Dirichlet boundary conditions, in the unit interval

$$\partial_t u = (1 + \mathrm{i}\alpha)\partial_{xx}^2 u + \mu u - (1 + \mathrm{i}\beta)|u|^2 u, \quad \text{with } u(0,t) = u(1,t) = 0, \tag{44}$$

where μ, α, and β are real parameters. The state variable u is complex and can be seen as the *complex amplitude* when using (44) as the weakly nonlinear 'normal form' [17] that applies at the onset of oscillatory instabilities in many systems, including fluid systems. The CGLE is a well-known paradigm of pattern forming systems [11]. It is a simple nonlinear equation that exhibits intrinsically complex dynamics [2] due to the modulational instability if $\alpha\beta < -1$ (Newell's condition) and μ exceeds a threshold value.

Table 1
The numbers M (retained modes) and N (collocation points/selected snapshots), and the online acceleration factor $C_{\text{online}}^{\text{LUPOD}}$ for the considered test cases and sets of tolerances. The acceleration factor if the ROM were constructed retaining M modes and using all mesh points is $C_{\text{online}}^{\text{ALL}}$.

		M	N	$C_{\text{online}}^{\text{LUPOD}}$	$C_{\text{online}}^{\text{ALL}}$
Test case 1	*set 1*	1	1	175.78	3.16
($T_0 = 1$)	*set 2*	1	1	175.78	3.16
	set 3	1	1	175.78	3.16
Test case 2	*set 1*	5	10	11.23	2.73
($T_0 = 0.23$)	*set 2*	6	10	10.82	2.70
	set 3	5	12	10.64	2.73
Test case 3	*set 1*	17	43	5.77	2.07
($T_0 = 0.16$)	*set 2*	21	43	5.39	1.90
	set 3	17	50	5.49	2.07

The problem (44) is invariant under the $D_1 \times SO(2)$ group generated by spatial reflection and phase translations, namely

$$x \to 1 - x \text{ and } u \to u e^{\mathrm{i}c}. \tag{45}$$

In all calculations below, the 1D CGLE is integrated with the initial condition

$$u(x, 0) = \mathrm{i} \sin(2\pi x) + (1 + \mathrm{i}) \sin(3\pi x), \tag{46}$$

which is not reflection-symmetric to avoid restriction of the dynamics to an invariant sub-manifold.

The FM (36) for the problem (44) is obtained straightforwardly, by discretizing the spatial derivatives via centered second-order finite differences in a grid of J equally spaced points that are interior to the interval $0 < x < 1$; this gives $J + 1$ spatial intervals and a total number of $J + 2$ grid points when the end points, $x = 0$ and 1 where $u = 0$, are also considered. After calibration for the most complex test case considered below (namely, the test case 3), the number of interior mesh points $J = 249$ and the timestep $h = 5 \cdot 10^{-4}$ have been selected such that the relative RMS and maximum errors (see (42)) between the numerical solutions computed (with the FM) with J and $2J + 1$ interior mesh points be smaller than 10^{-2} in time intervals of length T_0, where T_0 is as defined in the last subsection (see (43)).

The following test cases for the 1D CGLE, involving representative attractors, one steady and two periodic, are considered to check the performance of the LUPOD collocation strategy:

- *test case 1* (steady) for $(\alpha, \mu, \beta) = (2, 60, -3.5)$
- *test case 2* (simple periodic) for $(\alpha, \mu, \beta) = (2, 80, -3.5)$
- *test case 3* (complex periodic) for $(\alpha, \mu, \beta) = (2.23, 70, -11.34)$

The general method described in §4.1 is applied to these test cases, with the following values of the various tunable parameters, which have been selected after some calibration:

- The endpoints of the time interval (41) where the snapshots are selected, after discarding a transient stage $0 \le t < T_{\text{attr}}$, are $T_{\text{attr}} = 14$ and $T = 15$. The number of selected equispaced snapshots in this time interval is $K = 400$.
- Regarding the tolerances that fix the number N of collocation points/ selected snapshots and the number M of retained POD modes, three pairs of values are considered, namely

$$\textit{set 1: } (\varepsilon_N, \varepsilon_M) = (10^{-6}, 10^{-5}), \quad \textit{set 2: } (\varepsilon_N, \varepsilon_M) = (10^{-6}, 10^{-6}), \quad \textit{set 3: } (\varepsilon_N, \varepsilon_M) = (10^{-7}, 10^{-5}). \tag{47}$$

  The sets 2 and 3 help to elucidate the effect of increasing (from the baseline case in set 1) the numbers of retained modes, M, and collocation points, N, individually. Detailed results for a fourth case, $(\varepsilon_N, \varepsilon_M) = (10^{-7}, 10^{-6})$, in which both M and N are increased simultaneously, are omitted to avoid too involved plots.
- The values of the timespan T_0 for the three test cases are indicated in Table 1, as calculated using eq. (43).

Finally, the efficiency of simulating the attractor over $[T_{\text{attr}}, T]$ by means of the preprocessed ROM based on the selected collocation points and snapshots is measured by the *online acceleration factor*

$$C_{\text{online}} = \frac{\text{CPU time (full model)}}{\text{CPU time (preprocessed ROM)}}, \tag{48}$$

in terms of the CPU times (using a desktop PC, with an Intel i7 – 3.5 GHz microprocessor and 8 GB RAM) required for the integration of the FM and the ROM. The online acceleration factors (denoted by $C_{\text{online}}^{\text{LUPOD}}$) for the three test cases and the various tolerance sets are summarized in Table 1, where it can be seen that they are fairly reasonable. In fact, if the ROM were constructed using the $J = 249$ interior mesh points (retaining the numbers M of modes indicated in Table 1), then

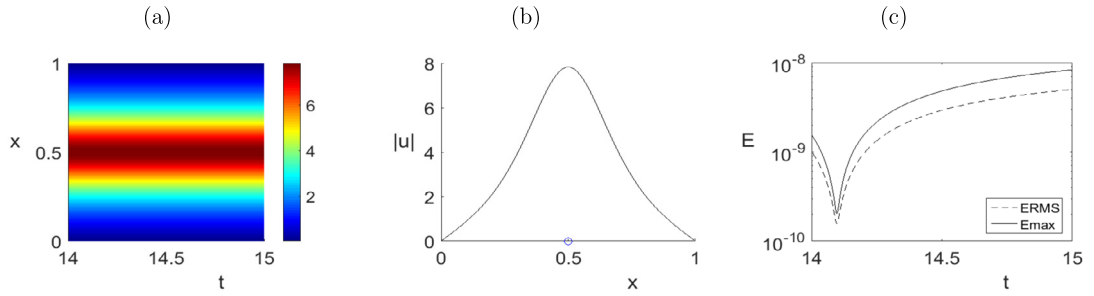

Fig. 12. Test case 1 for the 1D CGLE. (a) Spatio-temporal color map of $|u|$. (b) The selected snapshot and the collocation point (blue circle on the x-axis). (c) Evolution of the errors E_{RMS} (dashed line) and $E_{\max}$ (solid line) defined in (42).

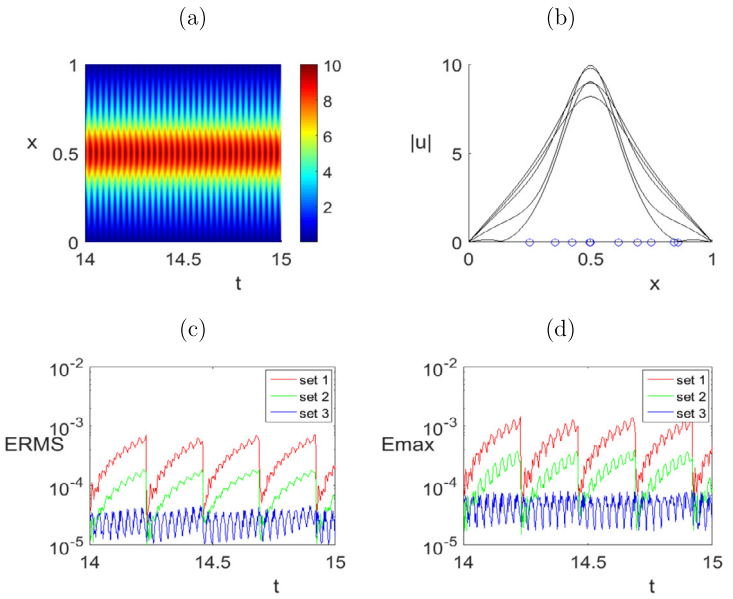

Fig. 13. Test case 2 for the 1D CGLE. (a) Spatio-temporal color map of $|u|$. (b) The selected $N = 10$ collocation points and the first five of the N selected snapshots (corresponding to the time instants $t = 14.2325, 14.8075, 14.0750, 14.2675, 14.0500$) using the tolerance set 1. (c,d) E_{RMS} and $E_{\max}$ errors for the three sets of tolerances defined in (47).

the acceleration factors (denoted by $C_{\text{online}}^{\text{ALL}}$ in Table 1) for the test problems 1, 2, and 3 would be ~ 3.2, ~ 2.7, and ~ 2, respectively, meaning that using the ROM would give a poor efficiency improvement, even in the case in which only one POD mode is retained. Table 1 also shows the numbers of collocation points/selected snapshots and retained POD modes in the three test cases, which are discussed now.

In test case 1, the system exhibits a reflection-symmetric (namely, invariant under the first action in (45)) steady attractor (Fig. 12(a)), meaning that just one reflection-symmetric snapshot and one collocation point (at $x = 0.5$, see Fig. 12(b)) suffice in the application of the LUPOD ROM, and identically coincide under the three tolerance sets 1, 2, and 3 (see (47)). Also, the E_{RMS} and $E_{\max}$ errors (which also coincide for the tolerance sets 1, 2, and 3) plotted in Fig. 12(c) are both fairly small and the online acceleration factor $C_{\text{online}}^{\text{LUPOD}}$ is quite large (see Table 1). Note that the timespan for the calculation of the errors has been set to $T_0 = 1$, and that the errors remain quite small and stabilize as time increases, which is due to the fact that the attractor is steady.

The attractor in the test case 2 is periodic and instantaneously reflection-symmetric, as seen in Fig. 13(a). This figure and Table 1 show that:

- The selected snapshots are all reflection-symmetric (see Fig. 13(b)), which is consistent with the fact that the reflection-symmetry is preserved at each time instant in the periodic attractor. The collocation points, instead, are not symmetrically located.
- As expected, the numbers of collocation points and retained modes, N and M, respectively, are larger than in the test problem 1, as are the E_{RMS} and $E_{\max}$ errors (Fig. 13(c,d)), while the acceleration factor is smaller (Table 1).
- Fig. 13(c,d) shows that slightly increasing the number of modes (set 2 of tolerances) yields much smaller errors and slightly reduces the online acceleration factor, from $C_{\text{online}}^{\text{LUPOD}} = 11.23$ to $C_{\text{online}}^{\text{LUPOD}} = 10.82$, while selecting more collocation

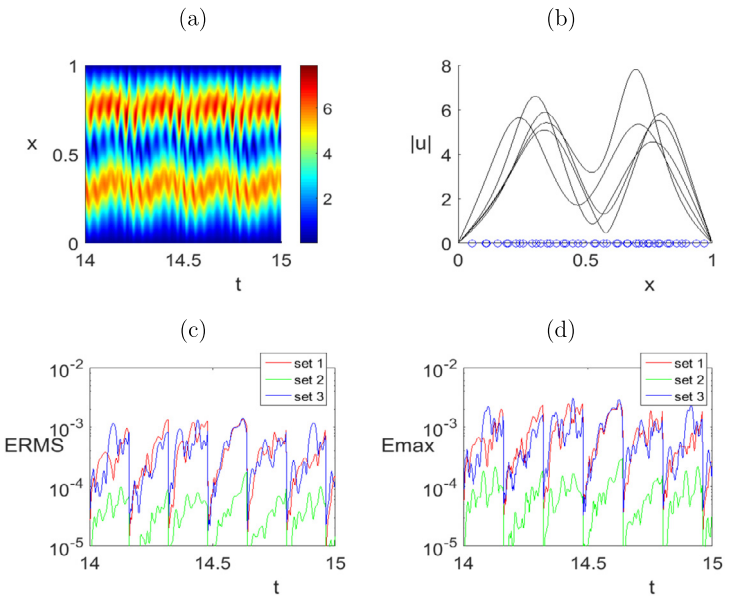

Fig. 14. Counterpart of Fig. 13 for the test case 3. The tolerance set 1 is considered in (b), where only the first five selected snapshots, corresponding to the time instants $t = 14.7525, 14.2350, 14.4950, 14.6900, 14.7875$, are plotted.

points/snapshots and keeping the same number of modes (set 3 of tolerances) decreases the errors even more, though is slightly less efficient ($C_{\text{online}}^{\text{LUPOD}} = 10.64$). In other words, increasing the number of collocation points has a larger benefit in terms of accuracy than increasing the number of retained modes in this case. If both M and N are increased simultaneously (the associated error curve is omitted in Fig. 13 to preserve clarity), setting $M = 6$ and $N = 12$, then the accuracy further improves, while the acceleration factor decreases only slightly.

- Both E_{RMS} and $E_{\max}$ errors increase in each timespan of length $T_0 = 0.23$. As anticipated, this increase is due to small errors in the frequency of the periodic attractor.

The periodic attractor of test case 3 is non-reflection-symmetric and fairly complex, as seen in Fig. 14, which also shows that the performance of the method is qualitatively similar to the previous case, except for the following peculiarities. The timespan $T_0 = 0.16$ is smaller due to the more complex oscillations exhibited by the attractor, and the selected snapshots (Fig. 14(b)) are non-reflection-symmetric. Again, increasing the number of retained modes (set 2 of tolerances) enhances the accuracy. However, selecting more collocation points and snapshots does not lead to a better approximation (compare the errors for the tolerance sets 1 and 3). Moreover, increasing both the number of collocation points and the number of retained modes gives a better accuracy. The best online acceleration factor (see Table 1), $C_{\text{online}}^{\text{LUPOD}} = 5.77$, is smaller than in the previous cases due to the larger number of involved points, snapshots, and modes.

Summarizing the above, for the 1D CGLE, LUPOD collocation combined with standard POD plus Galerkin projection gives fairly large acceleration factors keeping a good accuracy. In addition, because of the efficient collocation strategy, the accuracy is strongly improved with a slight increase of the numbers of collocation points/selected snapshots, N, and retained modes, M, which results in a slight decrease of the acceleration factor. This is clearly seen comparing the results in Table 1 and Figs. 13 and 14. However, the comparative benefit of increasing M and N individually depends on the particular dynamics.

4.3. Application to the CGLE in 2D

Let us now consider the two-dimensional CGLE in the unit square $0 < x, y < 1$ (cf (44))

$$\partial_t u = (1 + \mathrm{i}\alpha)(\partial_{xx}^2 u + \partial_{yy}^2 u) + \mu u - (1 + \mathrm{i}\beta)|u|^2 u, \tag{49}$$

$$u(0, y, t) = u(1, y, t) = u(x, 0, t) = u(x, 1, t) = 0, \tag{50}$$

which is invariant under the $D_4 \times SO(2)$ group generated by the actions

$$x \to 1 - x, \quad y \to 1 - y, \quad x \leftrightarrow y, \quad u \to u\mathrm{e}^{\mathrm{ic}}. \tag{51}$$

All simulations below are performed with the initial condition

$$u(x, y, 0) = (1 + 7\mathrm{i})(x - 3y)\sin(2\pi x)\sin(\pi y) + (2 + \mathrm{i})(2x + y)\sin(\pi x)\cos(\pi(1 + 2y)/2), \tag{52}$$

Table 2
Counterpart of Table 1 for the 2D CGLE.

		M	N	$C_{\text{online}}^{\text{LUPOD}}$	$C_{\text{online}}^{\text{ALL}}$
Test case 1	*set 1*	1	1	$6.35 \cdot 10^5$	190.02
($T_0 = 2$)	*set 2*	1	1	$6.35 \cdot 10^5$	190.02
	set 3	1	1	$6.35 \cdot 10^5$	190.02
Test case 2	*set 1*	9	17	$4.24 \cdot 10^4$	79.03
($T_0 = 0.37$)	*set 2*	10	17	$4.03 \cdot 10^4$	73.42
	set 3	9	21	$3.86 \cdot 10^4$	79.03
Test case 3	*set 1*	44	104	$1.10 \cdot 10^4$	21.44
($T_0 = 0.61$)	*set 2*	57	104	$8.74 \cdot 10^3$	15.13
	set 3	44	127	$9.14 \cdot 10^3$	21.44

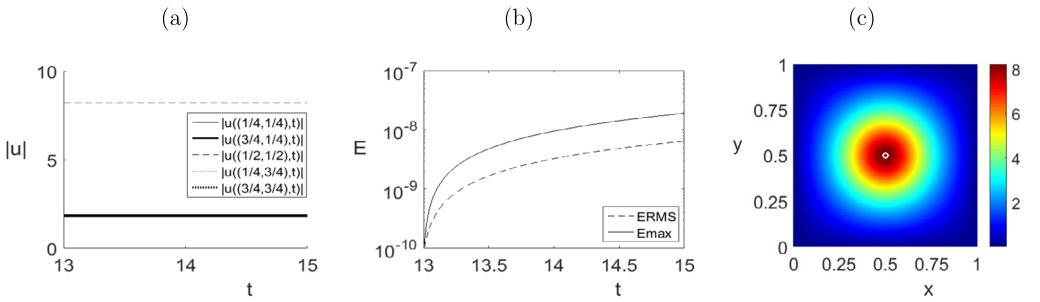

Fig. 15. Test case 1 for the 2D CGLE. (a) Time evolution of $|u|$ at the indicated points of the unit square. (b) E_{RMS} and $E_{\max}$ errors. (c) Color map of $|u|$ for the selected snapshot, with the collocation point indicated by a white circle.

which is not invariant under any of the actions (51).

The FM (36) for the problem (49)–(50) is constructed via spatial discretization using centered second-order finite differences in the x and y directions, in a grid of $(J+2) \times (J+2)$ equally spaced points, which include the $J \times J$ interior points plus the boundary points, where $u = 0$. After some calibration, similar to its counterpart in the one-dimensional cases considered in the last subsection, J has been set to 249, while the size of the timestep has been selected as $h = 5 \cdot 10^{-4}$. As for the one-dimensional case, the FM will be used to compute both the initial snapshots and the reference solution.

The following test cases are now considered to elucidate the performance of the LUPOD collocation strategy in two spatial dimensions:

- *test case 1* for $(\alpha, \mu, \beta) = (2, 60, -3.5)$
- *test case 2* for $(\alpha, \mu, \beta) = (2, 80, -3.5)$
- *test case 3* for $(\alpha, \mu, \beta) = (0.5, 85, -2.5)$

As in 1D, we discard an initial transient, for $0 \le t < T_{\text{attr}}$, and select K snapshots in the time interval $[T_{\text{attr}}, T]$. After some calibration, we set $K = 400$, $T_{\text{attr}} = 13$, and $T = 15$. The tolerances to fix the numbers of collocation points/selected snapshots and retained POD modes, N and M, respectively, are the same as in 1D, namely

$$\textit{set 1: } (\varepsilon_N, \varepsilon_M) = (10^{-6}, 10^{-5}), \quad \textit{set 2: } (\varepsilon_N, \varepsilon_M) = (10^{-6}, 10^{-6}), \quad \textit{set 3: } (\varepsilon_N, \varepsilon_M) = (10^{-7}, 10^{-5}), \tag{53}$$

which illustrates the robustness of the LUPOD strategy.

The outcomes of the LUPOD method applied to the nine cases resulting from these tolerances in each of the three test problems are summarized in Table 2. Comparison with Table 1 shows that the online acceleration factor is dramatically larger than in 1D. This is because, as anticipated, the acceleration factor scales with the ratio of the number of numerical degrees of freedom to the number of relevant degrees of freedom (namely, the number of retained POD modes). Now, the number of numerical degrees of freedom has been increased by a factor 249 (249×249 spatial mesh points in 2D and 249 in 1D, while taking the same timestep). The number of relevant degrees of freedom, instead, is only multiplied by a factor 2–3, instead of by a factor 249. This means that the advantage of the method is much more evident when the spatial dimension increases, which is fairly promising keeping in mind three-dimensional problems. As in the previous 1D cases, the benefit of using only the collocation points can be illustrated noting that if the whole set of 249×249 grid points were used (retaining the same numbers of POD modes), then the efficiency in the 2D tests would be much smaller, giving the online acceleration factors $C_{\text{online}}^{\text{ALL}}$ shown in Table 2.

Let us now consider the three test cases in more detail.

As in 1D, the test case 1 corresponds to a D_4-reflection-symmetric (namely, invariant under the first three actions in (51)), steady attractor, considered in Fig. 15, where the evolution of $|u|$ is illustrated at five representative points in the unit

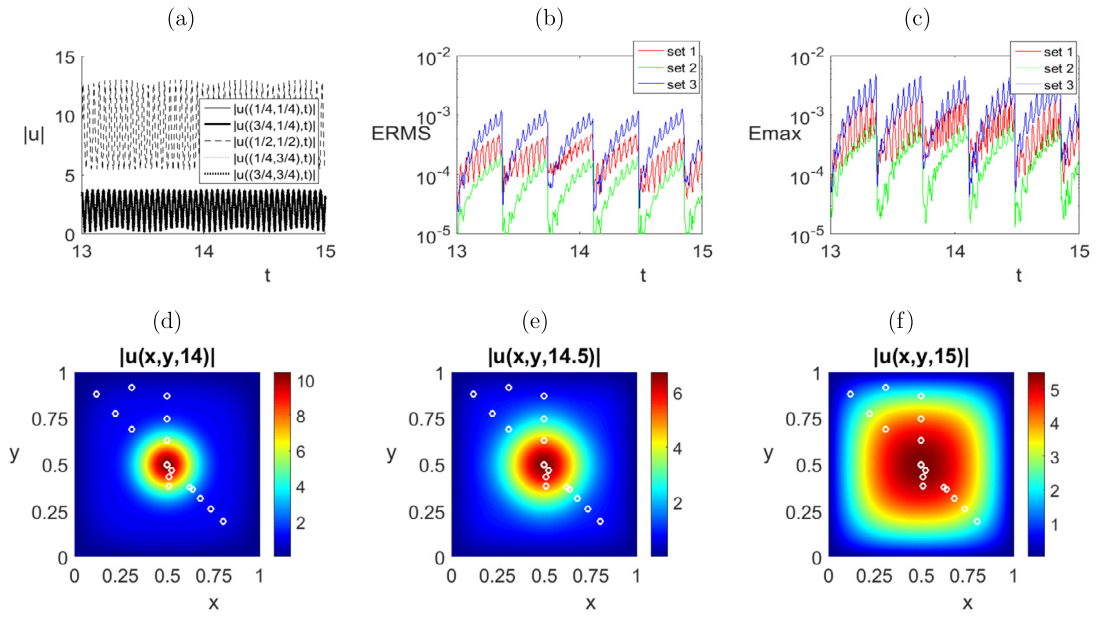

Fig. 16. Counterpart of Fig. 15 for the test case 2 for the 2D CGLE. (a) Time evolution of $|u|$ at the indicated points of the unit square. (b,c) The E_{RMS} and E_{max} errors. (d,e,f) Three representative instantaneous distributions of $|u|$, with the 17 sampled points (white circles) for the tolerance set 1 indicated in (53).

square (Fig. 15(a)). As expected, the LUPOD method selects just one collocation point (the center of the unit square) and one reflection-symmetric snapshot (Fig. 15(c)), which exactly coincide using the tolerance sets 1, 2, and 3. Also, as in 1D, the E_{RMS} and E_{max} errors are both fairly small (Fig. 15(b)), and the online acceleration factor $C_{\text{online}}^{\text{LUPOD}} = 6.35 \cdot 10^5$ (Table 2) is very impressive.

For the test case 2 (considered in Fig. 16), the attractor is periodic and instantaneously D_4-reflection-symmetric, namely invariant under the first three actions in (51) at all values of t. Such periodic attractor is fairly simple, since it represents a beating solution in which $|u|$ just grows and decays in a periodic fashion. As in 1D, the numbers of selected snapshots and collocation points are larger than in the steady attractor associated with the test case 1, and the E_{RMS} and E_{max} errors are larger. The 17 collocation points selected by the LUPOD for the set 1 of tolerances are indicated in Fig. 16(d,e,f) by white circles. We observe that, due to the symmetries in the solution, the selected collocation points are roughly located mainly on the $x = 1 - y$ diagonal and the vertical symmetry axis.

The test case 3, considered in Fig. 17, corresponds to a more complex, non-symmetric attractor. Although the dynamics are complicated, the LUPOD method for the set 1 of tolerances selects just 104 snapshots and collocation points. This number is very reduced in comparison with the original number of interior mesh points, 249×249. In Fig. 17(d,e,f) we observe that the solution does not contain spatially localized complexity. Instead, complexity is spread in the unit square and thus the selected collocation points are spread too. As expected and checked invoking Table 2, the number N of collocation points/selected snapshots increases as the tolerance ε_N decreases, as does the number M of retained POD modes as the tolerance ε_M decreases.

Summarizing the above, the behavior of the method for the 2D CGLE is fairly similar to that in 1D, though the use of the limited amount of collocation points gives much larger online acceleration factors (see Table 2), which is quite promising keeping in mind applications to other multidimensional problems.

5. Conclusions

A collocation method, labeled as LUPOD, has been developed in §2. This is based on a synergic combination of LU decomposition and POD, and is prepared to be used for both performing POD and combining POD with Galerkin projection. As explained in §2.1, the first step is a computationally inexpensive LU decomposition with pivoting of the snapshot matrix, which gives both the collocation points and a set of selected snapshots. Then, POD is applied to the selected snapshots, using an inner product based only on the collocation points. Thus, the method is both quite computationally inexpensive (namely, it does not increase the computational cost of standard POD) and synergic with POD. Moreover, as further explained in §2.2: (a) the selected snapshots are the most uncorrelated (namely, linearly independent) snapshots; and (b) the collocation points are precisely the points that account most efficiently for this strong linear independency. Because of these, for a given number N of retained snapshots, the LUPOD method seems to provide an accuracy that is (at least) as good as that attained with standard POD. We are unable to rigorously prove such statement, but it has been repeatedly tested in both

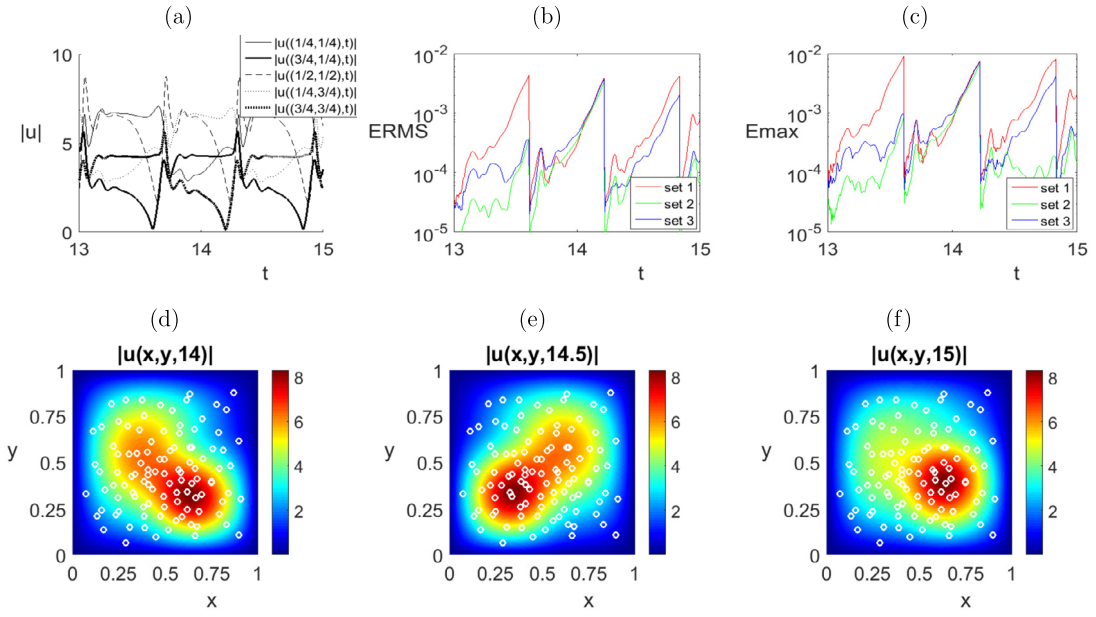

Fig. 17. Counterpart of Fig. 16 for the test case 3 for the 2D CGLE.

the examples given in §3 and §4 to illustrate the method and in many additional tests that are omitted for the sake of brevity.

Although the POD computations are based only on the collocation points, the POD modes (and their spatial derivatives, if needed) are reconstructed at all mesh points in a fairly natural way. In addition, the method has been prepared in §2.3 to simultaneously approximate the snapshots and their spatial derivatives.

The method was compared in §3.1 with classical spectral methods, concluding that LUPOD yields very similar sets of collocation points as spectral collocation (see Fig. 1). In addition, the method was illustrated in §3.2 with several toy models, exhibiting both spread and concentrated complexity (Figs. 4–5). The LUPOD collocation points are placed precisely in those regions where complexity is largest, yielding fairly good descriptions in terms of only few collocation points and POD modes, which leads to a fairly favorable comparison with both standard POD and POD based on equispaced distributions of collocation points (see Fig. 6). The advantages of LUPOD are even clearer for very sharp regions, with very steep gradients (or higher order derivatives), especially in the presence of noise (see Figs. 7–9). In addition, the method was applied in §3.3 to an aerodynamic database, obtaining again very good descriptions using a limited amount of collocation points and POD modes (see Fig. 11).

The combination with Galerkin projection was considered in §4, first for a general dissipative system, in §4.1, and then for the CGLE in one (§4.2) and two (§4.3) space dimensions, choosing for both three test cases and three combinations of the numbers of collocation points and retained modes. The results, summarized in Tables 1 and 2, show that the LUPOD method is quite robust and yields fairly large online acceleration factors (comparing with standard numerical simulation), which are much larger than those provided by standard POD. More importantly, the acceleration factor is much larger in 2D than in 1D, which is promising for application to multidimensional large scale problems, which are beyond the scope of this paper and will be considered elsewhere.

Acknowledgements

This work was partially supported by the Spanish Ministry of Economy and Competitiveness, under grants TRA2013-45808-R, MTM2014-56948-C2-1-P, and MTM2014-56948-C2-2-P. Several constructive criticisms and useful suggestions by the Editor and two anonymous referees on an earlier version of the paper are also gratefully acknowledged.

References

[1] D. Alonso, J.M. Vega, A. Velázquez, V. de Pablo, Reduced-order modeling of three-dimensional external aerodynamic flows, J. Aerosp. Eng. 25 (2012) 588–599.
[2] I.S. Aranson, L. Kramer, The world of the complex Ginzburg–Landau equation, Rev. Mod. Phys. 74 (2002) 100–142.
[3] P. Astrid, S. Weiland, K. Willcox, T. Backx, Missing point estimation methods in models described by proper orthogonal decomposition, IEEE Trans. Autom. Control 53 (2008) 2237–2250.
[4] M. Barrault, Y. Maday, N.C. Nguyen, A.T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, C. R. Math. Acad. Sci. Paris 339 (2004) 667–672.
[5] M. Benzi, Preconditioning techniques for large linear systems: a survey, J. Comput. Phys. 182 (2002) 418–477.

[6] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 25 (1993) 539–575.
[7] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods: Fundamentals in Single Domains, Springer-Verlag, Berlin & Heidelberg, 2006.
[8] T. Cebeci, Convective Heat Transfer, Springer, Berlin, 2002.
[9] A. Chatterjee, An introduction to the proper orthogonal decomposition, Curr. Sci. 78 (2000) 808–817.
[10] S. Chaturantbut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (2010) 2737–2764.
[11] M. Cross, P. Hohenberg, Pattern formation outside of equilibrium, Rev. Mod. Phys. 65 (1993) 851–1112.
[12] Ch. Davies, The rotation of eigenvectors by a perturbation, J. Math. Anal. Appl. 6 (1963) 159–163.
[13] R. Everson, L. Sirovich, Karhunen–Loeve procedure for gappy data, J. Opt. Soc. Am. A 12 (1995) 1657–1664.
[14] G.H. Golub, G.T. van Loan, Matrix Computations, John Hopkins Univ. Press, 1996.
[15] G.H. Golub, K. Sølna, P. van Dooren, Computing the SVD of a general matrix product/quotient, SIAM J. Matrix Anal. Appl. 22 (2000) 1–19.
[16] D. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, SIAM, Philadelphia, 1977.
[17] M. Haragus, G. Iooss, Local Bifurcations, Center Manifolds, and Normal Forms in Infinite Dimensional Dynamical Systems, Springer-Verlag, 2010.
[18] L.S. Lorente, J.M. Vega, A. Velázquez, Compression of aerodynamic databases using high-order singular value decomposition, Aerosp. Sci. Technol. 14 (2010) 168–177.
[19] A.I. Moreno, A.A. Jarzabek, J.M. Perales, J.M. Vega, Aerodynamic database reconstruction via gappy high order singular value decomposition, Aerosp. Sci. Technol. 52 (2016) 115–128.
[20] K. Pearson, On lines and planes of closest fit to systems of points in space, Philos. Mag. 2 (1901) 559–572.
[21] J.S. Peterson, The reduced basis method for incompressible viscous flow calculations, SIAM J. Sci. Comput. 10 (1989) 777–786.
[22] M.-L. Rapún, F. Terragni, J.M. Vega, Adaptive POD-based low-dimensional modeling supported by residual estimates, Int. J. Numer. Methods Eng. 9 (2015) 844–868.
[23] M. Renardy, R.C. Rogers, An Introduction to Partial Differential Equations, Texts Appl. Math., vol. 13, Springer-Verlag, New York, 2004.
[24] M.J. Rewiénski, A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems, Ph.D. thesis, Massachusetts Institute of Technology, 2003.
[25] M. Rewiénski, J. White, Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations, Linear Algebra Appl. 415 (2006) 426–454.
[26] D. Ryckelynck, Hyper-reduction of mechanical models involving internal variables, Int. J. Numer. Methods Eng. 77 (2009) 75–89.
[27] G.W. Stewart, Error and perturbation bounds for subspaces associated with certain eigenvalue problems, SIAM Rev. 15 (1973) 727–764.
[28] G.W. Stewart, On the early history of the singular value decomposition, SIAM Rev. 35 (1993) 551–566.
[29] F. Terragni, E. Valero, J.M. Vega, Local POD plus Galerkin projection in the unsteady lid-driven cavity problem, SIAM J. Sci. Comput. 33 (2011) 3538–3561.
[30] F. Terragni, J.M. Vega, On the use of POD-based ROMs to analyze bifurcations in some dissipative systems, Physica D 241 (2012) 1393–1405.
[31] F. Terragni, J.M. Vega, Construction of bifurcation diagrams using POD on the fly, SIAM J. Appl. Dyn. Syst. 13 (2014) 339–365.
[32] M.E. Wall, A. Rechtsteiner, L.M. Rocha, Singular value decomposition and principal component analysis, in: D.P. Berrar, W. Dubitzky, M. Granzowr (Eds.), A Practical Approach to Microarray Data Analysis, Kluwer, 2003, pp. 91–109.