Adv Comput Math (2021) 47: 48
https://doi.org/10.1007/s10444-021-09870-x

Adaptive sampling and modal expansions in pattern-forming systems

M.-L. Rapún[1] · F. Terragni[2] · J. M. Vega[1]

Received: 9 June 2020 / Accepted: 8 April 2021 / Published online: 16 June 2021

Abstract

A new sampling technique for the application of proper orthogonal decomposition to a set of snapshots has been recently developed by the authors to facilitate a variety of data processing tasks (*J. Comput. Phys.* 335, 2017). According to it, robust modal expansions result from performing the decomposition on a limited number of relevant snapshots and a limited number of discretization mesh points, which are selected via Gauss elimination with double pivoting on the original snapshot matrix containing the given data. In the present work, the sampling method is adapted and combined with low-dimensional modeling. This combination yields a novel adaptive algorithm for the simulation of time-dependent non-linear dynamics in pattern-forming systems. Convenient snapshot sets, computed on demand over the evolution, are stored to record local temporal events whose underlying mechanisms are essential for the approximations. Also, a collection of sparse grid points, which are used to construct the mode basis and the reduced system of equations, is adaptively sampled according to unlinked spatial structures. The outcome is a reduced order model of the problem that (i) yields reliable approximations of the dynamical transitions, (ii) is well-suited to describe localized spatio-temporal complexity, and (iii) provides fast computations. Robustness, accuracy, and computational efficiency of the proposed algorithm are illustrated for some relevant pattern-forming systems, in both one and two spatial dimensions, exhibiting solutions with a rich spatio-temporal structure.

Keywords Adaptive algorithms · Pattern-forming systems · Reduced order models · Sampling techniques · Collocation methods · Proper orthogonal decomposition · Kuramoto-Sivashinsky equation · Complex Ginzburg-Landau equation

Communicated by: Stefan Volkwein

✉ M.-L. Rapún
marialuisa.rapun@upm.es

[1] E.T.S.I. Aeronáutica y del Espacio, Universidad Politécnica de Madrid, 28040 Madrid, Spain

[2] G. Millán Institute and Department of Mathematics, Universidad Carlos III de Madrid, 28911 Leganés, Spain

Mathematics Subject Classification (2010) 65M60 · 76M25

1 Introduction

Low-dimensional modeling of evolution problems governed by partial differential equations is widespread in many fields. The increasing interest it sparked in the last few decades is due to several reasons. A truncated expansion of the state variable in terms of (generally orthogonal) modes is a typical approach, which leads to uncover the essence of non-linear dynamics, thus providing insights into the underlying physical phenomena [49]. This reduction to a finite and frequently small number of degrees of freedom also entails a decreasing of the computational cost of required numerical simulations, making it appealing in both scientific and industrial applications. The core of a large class of reduced order models (ROMs) is proper orthogonal decomposition (POD), invented by Pearson in 1901 [35]. Given a set of K (real or complex) vectors of size J, called snapshots, POD provides a hierarchical set of modes that form an orthonormal basis of the linear subspace generated by the vectors. In numerical approaches to problems where partial differential equations naturally arise, the snapshots are discretized spatial distributions of a state vector at K different values of time or a parameter [45]. Taking advantage of links imposed by the involved physical laws, the $M \ll \min\{J, K\}$ most energetic POD modes may describe the relevant degrees of freedom of the system, providing the best root mean square (RMS) approximation of the snapshot set. Projection of the governing (real or complex) equations onto the identified POD modes yields a system of M ordinary differential equations, whose integration is in principle much faster than solving the full problem.

POD-based ROMs are designed to simulate both steady [3, 4, 9] and unsteady [2, 13, 18, 20, 26, 33, 48] dynamics. Often, they are non-adaptive, meaning that POD is applied at the outset to representative snapshots calculated offline (in a preprocess) by a standard numerical solver (NS) along the whole desired time or parameter span. The resulting low-dimensional model is then integrated to approximate attractors of the system (not transients), thus making the online simulation more efficient than using the NS. However, the offline stage may be quite computationally costly. On the other hand, adaptive POD-based ROMs [12, 14, 39, 41, 50, 52] intend to be a way of drastically alleviating this step, by relying on the continuous dependence of the POD subspace on time and parameters. The mode expansion and the low-dimensional system are adjusted or updated over the evolution, by means of various strategies, in order to accurately follow the actual trajectories. However, for general non-linear time-dependent problems, the computational cost of projecting the equations, often via the Galerkin method, scales as $M \times J$, where M and J are the numbers of retained modes and selected spatial discretization points, respectively. Indeed, this factor may be huge if the chosen inner product is based on all discretization points, which magnifies the effort, especially when non-linearity forces to perform the projection at each time step of the reduced model integration.

Similarly as in a collocation-like spectral framework [25], a way to overcome such difficulty consists in using a suitable reduced product, defined in terms of a limited number of mesh points, say $N \sim M$. For instance,

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

for some conveniently sampled grid points denoted by the indexes $j_1, \ldots, j_N$ (here, $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ are vectors in $\mathbb{C}^J$, while the overbar stands for the complex conjugate). The set of points implicit in Eq. (1) can result from various sampling techniques, such as the missing point estimation [7], the empirical interpolation [10, 23], the discrete empirical interpolation and improvements [17, 21, 36], the hyper-reduction [5, 43], or ad hoc selections privileging regions with rich spatial structures [4, 50]. We refer to [22] for an interesting comparative study between missing point estimation, discrete empirical interpolation, and gappy POD methods, when dealing with reduced order modeling of the spatio-temporal dynamics in a predator-prey problem. See also [11, 54] for the use of reduced sets of grid points in the context of low-dimensional modeling. In [40], we developed a collocated POD, called LUPOD, in which "optimal" sets of snapshots and points are sampled, via an incomplete LU decomposition (i.e., truncated Gauss elimination) with double pivoting of the snapshot matrix. Precisely, the N most linearly independent snapshots and the N spatial points better accounting for such independence are identified. Then, POD is applied to the selected snapshots through a product as in Eq. (1) based on the selected points. The LUPOD method turned out to be simple, robust, and computationally cheap in the context of data processing tasks.

The main goal of the present work is the construction of a collocated LUPOD-based ROM, in which adaptation to the local dynamics of a given system is carried out in three different stages:

1) Snapshots selection is non-homogeneous in time and captures the most different states. This reinforces POD from a temporal point of view, namely, the computed modes convey local information about transitions and detours of the trajectories.
2) Points for the reduced product in Eq. (1) are non-uniformly sampled and mainly located in spatial regions showing strong variation of the solution structures. This reinforces POD from a spatial point of view, since the computed modes describe quite well spatially localized patterns. Moreover, Galerkin projection of the equations is based on the selected sparse collocation points, whose number N is close to the number of retained modes, say $N \sim M$.
3) Modes are adapted, in both space and time, by applying POD to sets of snapshots collected, throughout the simulation, in different time intervals and by means of locally "optimal" sets of grid points. Hence, past information helps to predict future dynamics.

Adaptation is achieved by the LUPOD sampling technique, which is employed on demand as time proceeds, starting from some input snapshots computed by a standard

NS. More precisely, relying on an idea discussed in a previous work [39], an alternation between the NS and a low-dimensional system obtained by Galerkin projection, called the Galerkin system (GS), defines the main loop of the proposed algorithm. The constructed GS is time-integrated as long as accuracy and absence of high-order mode truncation instabilities [42] are guaranteed. If not, an updating procedure based on the three steps above is turned on, which results in a new, improved GS that is better adapted to the actual local dynamics. The outcome is an adaptive LUPOD-assisted, collocated ROM, in which the involved sets of snapshots, collocation points, and POD modes vary over time to provide an approximation within a desired accuracy. Thus, this ROM retains the local physical degrees of freedom of the system solutions, meaning that the number of degrees of freedom that is accounted for is never larger than necessary. These features mark a crucial difference versus previous approaches by the authors, e.g., [39], in which the spatio-temporal adaptation was absent and no sampling technique was taken into account. Indeed, only POD modes were updated, in a different and more involved way, by means of uniformly distributed snapshots and equally spaced grid points, fixed throughout a whole simulation. Also, due to their static and non-optimal distribution, the number of both snapshots and grid points had to be much larger than the amount of retained modes.

Summarizing, the novel LUPOD-based method intends to be an improvement of former reduced order models in terms of the following: (i) reliability of the low-dimensional approximations, (ii) applicability to spatio-temporally complex dynamics not attainable by former strategies, and (iii) computational performance. These improvements will be tested via application to typical pattern-forming systems, namely non-equilibrium systems with the ability to self-organize into temporal, spatial, and spatio-temporal structured states from homogeneous initial conditions. Besides appearing in a variety of fields like physics, chemistry, and biology, such systems will allow us to examine several dynamical behaviors and spatial complexities, including spatio-temporal chaos and traveling structures. We want to emphasize that, for the illustrated numerical experiments (see Section 4), adaptation is mandatory. Indeed, the proposed method is designed to approximate not only attractors but also transients. For the latter, the initial POD subspace is able to properly describe the dynamics on some time interval, but when drifts occur these modes are no longer suitable to continue the ROM integration and updating is needed. For this reason, our adaptive method not only monitors mode truncation errors but also anticipates mode truncation instabilities by using residual estimates. The necessity of both controls, as well as some comparisons with non-adaptive strategies, was thoroughly discussed for previous versions of the method (see [39, 41]). Indeed, for all cases tested in the present work, standard non-adaptive POD plus Galerkin projection methods (with or without collocation/sampling strategies), or any other non-adaptive approach, will be even more likely to fail either when the dynamics suffer drifts, similarly as shown in [39, 41], or especially when concentrated spatio-temporal complexity demands a non-homogeneous selection of collocation points and snapshots.

The remainder of the manuscript is organized as follows. Standard POD and the LUPOD sampling algorithm are outlined in Section 2, while the novel adaptive LUPOD-based ROM is presented in Section 3. Section 4 is devoted to numerical experiments on some benchmark problems, the Kuramoto-Sivashinsky equation,

the one-dimensional complex Ginzburg-Landau equation with or without drift, and the standard two-dimensional complex Ginzburg-Landau equation with constant or space-dependent linear growth coefficient. Some final remarks can be found in Section 5.

2 The LUPOD sampling method

Let us consider a set of K (generally complex) snapshots of dimension J, namely $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K \in \mathbb{C}^J$. Standard truncated POD applied to these vectors [16, 45] provides $M < K$ (orthonormal) POD modes, $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M \in \mathbb{C}^J$, such that each $\boldsymbol{s}_k$ can be approximated as

$$\boldsymbol{s}_k \simeq \sum_{m=1}^{M} \alpha_k^m \boldsymbol{u}_m, \tag{2}$$

where the coefficients α_k^m result from the orthogonal projection of the snapshots onto the linear subspace generated by the modes, namely $\alpha_k^m = \langle \boldsymbol{s}_k, \boldsymbol{u}_m \rangle$. POD is optimal, in the sense that Eq. (2) yields the best joint RMS approximation of the snapshots based on M modes, with respect to the considered inner product $\langle \cdot, \cdot \rangle$ (e.g., the Euclidean product in $\mathbb{C}^J$ given by $\langle \boldsymbol{s}_1, \boldsymbol{s}_2 \rangle = \sum_{j=1}^{J} s_1^j \bar{s}_2^j$). Note that POD can be computed via singular value decomposition (SVD) [24]. In this case, the joint RMS error of the approximation in Eq. (2) can be calculated, using well-known formulae [24], in terms of the singular values of the matrix whose columns are the snapshots, called the snapshot matrix.

In the context of low-dimensional modeling of partial differential equations, snapshots are typically spatial distributions of the state variable q (assumed for simplicity to be scalar), discretized on a grid with J points, at K different time instants $t_1, \ldots, t_K$. Thus, the spatially discretized state variable is a vector of size J, whose components are the values of q at the grid points. Denoting this vector by $\boldsymbol{q}$, the snapshots are written as $\boldsymbol{s}_k = \boldsymbol{q}(t_k) \in \mathbb{C}^J$. They should be representative of the dynamical behavior of the solution along the considered trajectory. Besides that, a second important ingredient of POD is the inner product, which generally depends on the specific system and application. In this context, the advantage of using the reduced product in Eq. (1), based on a limited number of mesh points, is twofold: (i) the computational cost is alleviated (as already mentioned in Section 1) and (ii) a careful selection of the grid points, implicit in Eq. (1), can lead to identify or emphasize patterns and spatial structures hidden in the data.

The sampling technique described in [40], called LUPOD, provides an efficient selection of both snapshots and mesh points, thus enhancing the ability of POD to extract essential information from a dataset, built by the given snapshots in the present case. The core of LUPOD is an incomplete LU decomposition (i.e., Gauss elimination) of the snapshot matrix $\boldsymbol{S}$, with double pivoting, based on linear combinations of its columns. This allows to sample a subset $\{j_1, \ldots, j_N\}$ of rows of $\boldsymbol{S}$ and

a subset $\{k_1, \ldots, k_N\}$ of columns of $\boldsymbol{S}$, which identify N *collocation points* and N *relevant snapshots*, respectively, to efficiently perform POD. The outcome of truncated POD based on the selected points and snapshots gives a set of M POD modes. The complete method is summarized in the Algorithm 1.

Algorithm 1 The LUPOD method.

1: **function** LUPOD($\boldsymbol{S}$, ε_N, ε_M)
2: points $\leftarrow \{\,\}$
3: snaps $\leftarrow \{\,\}$
4: $\hat{\boldsymbol{S}} \leftarrow \boldsymbol{S}$
5: $n \leftarrow 1$
6: **while** $\|\hat{\boldsymbol{S}}\|_{\max} / \|\boldsymbol{S}\|_{\max} \geq \varepsilon_N$ **do**
7: $S_{j_n k_n} \leftarrow \|\hat{\boldsymbol{S}}\|_{\max}$ ▷ pivot
8: $\hat{\boldsymbol{S}} \leftarrow \text{Gauss}(\hat{\boldsymbol{S}}, S_{j_n k_n})$ ▷ Gauss elimination
9: points$[n] \leftarrow j_n$
10: snaps$[n] \leftarrow k_n$
11: $n \leftarrow n + 1$
12: **end while**
13: M, $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{M_1}\} \leftarrow \text{POD}(\text{points}, \text{snaps}, \varepsilon_M)$ ▷ 'reduced', truncated POD
14: **return** points, snaps, M, $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{M_1}\}$
15: **end function**

According to this algorithm, at the nth iteration, a new collocation point and a new relevant snapshot are identified by the indexes of the largest element (in absolute value), $S_{j_n k_n}$, of the current "modified" snapshot matrix $\hat{\boldsymbol{S}}$, namely, indexes j_n and k_n define the nth collocation point and the nth selected snapshot, respectively. Using $S_{j_n k_n}$ as pivot, $\hat{\boldsymbol{S}}$ is the result of applying Gauss elimination with double pivoting to the previous "modified" matrix, namely the outcome of setting to zero the elements of its j_nth row by linear combinations of columns and removing its k_nth column. Note that, in the algorithm, $\|\boldsymbol{S}\|_{\max} = \max_{jk} |S_{jk}|$ is the maximum norm and ε_N is a given level of accuracy. Hence, according to the condition in line 6, all original snapshots are approximated, within a relative maximum error ε_N, by linear combinations of the selected snapshots using their values at the sampled grid points. This is consistent with the essence of POD, as Gauss elimination somehow maximizes linear independence in connection with the choice of pivots. In other words, collocation points are related to extrema of the "modified" snapshots, occurring at spatial regions where the original snapshots are more independent. At the end of the main loop of the LUPOD algorithm (line 13), truncated POD is performed in the following way. Standard SVD is applied to a "reduced" snapshot matrix, which is constructed by considering in $\boldsymbol{S}$ only those rows and columns corresponding to the selected N collocation points and N relevant snapshots, respectively. This is equivalent to performing standard POD on

the sampled snapshots with the reduced product in Eq. (1), based on the collocation points. The number M of retained modes is defined in terms of a threshold ε_M as

$$\mathrm{RMSE}_M^R = \frac{\sqrt{\sigma_{M+1}^2 + \ldots + \sigma_R^2}}{\sqrt{\sigma_1^2 + \ldots + \sigma_R^2}} < \varepsilon_M, \tag{3}$$

where the left-hand side is the relative RMS error in the reconstruction of the "reduced" snapshot matrix by means of M modes (σ_i and R are the involved singular values and rank, respectively). The computed POD modes $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$ are orthonormal with respect to the reduced product in Eq. (1), based on the collocation points. These POD modes can be reconstructed at all grid points by algebraic manipulations. Further details about how POD is applied in the context of LUPOD can be found in [40], where the reader can also find a numerical experiment comparing the accuracy and computational effort of LUPOD versus standard SVD/POD based on equispaced collocation points. Observe that there are only two tunable parameters in the LUPOD algorithm, namely ε_N and ε_M, determining the number N of sampled points/snapshots and the number M of retained modes, respectively. By construction, $N \geq M$. In addition, note that a number $M_1 > M$ of modes is actually retained as output of truncated POD in line 13 of Algorithm 1, which will be needed and discussed in the next section.

Let us note that LUPOD differs from other methods in the literature, such as DEIM [17] or the so-called Q-DEIM [21]. Certainly, LUPOD and (Q)-DEIM share similarities as both are related to Gauss elimination. However, the strategies are different and consequently the sampled points and the resulting bases are distinct. Indeed, Q-DEIM is an improvement of standard DEIM (discrete empirical interpolation method) [17] that consists of two steps. First, truncated POD is applied to the full snapshot matrix, which yields a number of POD modes. In the second step, a QR decomposition with pivoting is applied to the mode matrix, which selects a set of spatial points to be used for Galerkin projection. On the other hand, as discussed above, the LUPOD method consists in performing a truncated (with a given accuracy) Gauss elimination (or LU decomposition) with double pivoting on the full snapshot matrix, which samples the N most linearly independent snapshots and a set of N spatial points better accounting for such independence. Thereafter, POD is applied to the chosen snapshots using the selected points only and Galerkin projection is performed considering the values of the POD modes at the same collocation points. In Fig. 12 below we will show a graphical comparison between points selected by the LUPOD and DEIM methods.

Summarizing, the LUPOD sampling technique aims at selecting the "optimal" points for the reduced product in Eq. (1), namely those where the considered snapshots are more independent. Additionally, as thoroughly discussed in [40], the method identifies the "best" snapshots to compute POD (note that temporal sampling can also be performed by other techniques, for instance the reduced basis method [38]). In other words, LUPOD reveals an improved accuracy of the approximations due to a *double sampling* in space and time, namely:

– Localized spatial patterns, associated with small L_2-norm but steep gradients of the state variable, are identified and described using a small amount of properly concentrated mesh points.

– Temporal complexity, confined to few snapshots and fast events, is also taken into account via sampling over time.

Thus, in view of the inherent ability of LUPOD to detect spatio-temporal complexity, we aim at *adapting* and *integrating* this sampling technique into low-dimensional modeling of pattern-forming systems, where a rich variety of transitions gives rise to the emergence of temporal, spatial, and spatio-temporal structures in a natural way.

3 ROMs based on adaptive sampling and modal expansions

The combination of POD with Galerkin projection of the equations under study is a widespread technique to generate low-dimensional approximations of infinite dimensional dynamical systems [12, 39, 44, 50]. In the present work, we consider non-linear evolution problems of dissipative type, which typically arise in the context of pattern-forming systems (a discussion about the assumptions on the involved equations that favor the construction of flexible reduced order descriptions can be found in [51]). A standard approach starts from the following ansatz for the discretized state variable $\boldsymbol{q}$,

$$\boldsymbol{q} \simeq \boldsymbol{q}_{\mathrm{GS}}^{M} = \sum_{m=1}^{M} a_m(t)\boldsymbol{u}_m = \boldsymbol{U}\boldsymbol{a}\,, \tag{4}$$

justified by the separation of variables principle. In Eq. (4), $\boldsymbol{u}_m$ are the spatial POD modes, stored as columns in the matrix $\boldsymbol{U}$, and $a_m(t)$ are the corresponding time-dependent amplitudes, which are collected into the vector $\boldsymbol{a}$. Then, this linear expansion is substituted into the governing equations, which are finally projected onto the POD basis by means of a convenient inner product. The result is a system of M ordinary differential equations for the unknown mode amplitudes $a_1, \ldots, a_M$. As anticipated, it will be called Galerkin system (GS), which is a ROM of dimension M. If the K snapshots considered in the POD procedure are strongly dependent, then $M \ll \min\{J, K\}$ and, in principle, the dynamics of the given full system, integrated by a standard NS, can be approximately recovered by the solution of the GS, $\boldsymbol{q}_{\mathrm{GS}}^{M}$, with the advantage of having a simplified model that can be integrated in terms of limited computational resources.

Our main goal consists in constructing adaptive ROMs based on the LUPOD sampling technique recalled in Section 2. The proposed novel LUPOD-based low-dimensional model will alternate the direct integration of the governing equations by a standard NS and the reduced integration of a GS in interspersed time intervals, called I_{NS} and I_{GS}, respectively. This alternation is inspired by some ideas that can be found in [39], but all fundamental elements comprised by the new method and the implemented algorithm are markedly different, resulting in a novel ROM in terms of construction, core ingredients, implementation, and range of applicability, as it will be discussed below. A summary of the new adaptive LUPOD-based ROM to efficiently approximate the dynamics of a given system in the timespan $0 < t \leq T$, within accuracy ε, is given in the Algorithm 2.

Algorithm 2 The LUPOD-based ROM.

1: **function** LUPOD-BASED_ROM($\varepsilon, \varepsilon_N, \varepsilon_M, \varepsilon_{\text{old}}$)
2: solution $\leftarrow \{\}$
3: $t \leftarrow 0$
4: **while** $t < T$ **do**
5: $\boldsymbol{S} \leftarrow \text{NS}(I_{\text{NS}})$ ▷ NS (full problem)
6: solution $\leftarrow$ {solution, $\boldsymbol{S}$}
7: $t \leftarrow t + I_{\text{NS}}$
8: points, snaps, M, $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{M_1}\} \leftarrow \text{LUPOD}(\boldsymbol{S}, \varepsilon_N, \varepsilon_M)$ ▷ LUPOD
9: **if** not first I_{NS} **then**
10: points, snaps, M, $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{M_1}\} \leftarrow$ updating(snaps, $\varepsilon_N, \varepsilon_M, \boldsymbol{S}_{\text{old}}, \varepsilon_{\text{old}}$)
11: **end if**
12: **while** $E_M^{M_1} < \varepsilon$ and $E_{\text{res}}^{M_1} < \varepsilon/10$ **do** ▷ error controls
13: $\boldsymbol{S}_{\text{GS}}, I_{\text{GS}} \leftarrow \text{GS}(\text{points}, \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{M_1}\})$ ▷ GS (reduced problem)
14: **end while**
15: solution $\leftarrow$ {solution, $\boldsymbol{S}_{\text{GS}}$}
16: $t \leftarrow t + I_{\text{GS}}$
17: $\boldsymbol{S}_{\text{old}} \leftarrow$ snaps ▷ snapshots storing
18: **end while**
19: **return** solution
20: **end function**

As seen in this algorithm, in the first I_{NS} interval, the NS provides an initial set of snapshots to which the LUPOD technique is applied, yielding a set of collocation points, a set of relevant snapshots, and a set of modes. These allow to construct, by means of the reduced product in Eq. (1) based on the sampled collocation points, a GS that is time-integrated in the next I_{GS} interval. Whenever any of the error controls in line 12 is not satisfied, then a *triple updating* is performed to preserve accuracy and stability of the approximation in future time instants. Precisely, in a new I_{NS} interval:

1. The collection of available snapshots is renewed by adding new relevant snapshots selected by LUPOD from a set of initial NS-computed states. The new snapshots are joined together with some of the "old" snapshots (stored in $\boldsymbol{S}_{\text{old}}$) sampled by LUPOD in the previous I_{NS} interval (*snapshots update*).
2. A final LUPOD is applied to the resulting set in point 1, yielding a new POD basis (*modes update*).
3. The last LUPOD run also provides new collocation points to construct a new GS for the approximation of the dynamics in a new I_{GS} interval (*collocation points update*).

This defines the role of the novel updating procedure in line 10 of the Algorithm 2. Note that, according to it, the numbers of locally involved snapshots, modes, and collocation points may vary over time. The update of the sets of relevant snapshots and collocation points, described in steps 1 and 3 above, was absent in previous

approaches and makes the new adaptive ROM a powerful tool to efficiently describe evolving patterns (see Section 4). Note that proper numbers of involved collocation points, snapshots, and POD modes are determined by means of tunable tolerances, as discussed in the remarks below. Thus, different and non-uniform sets of collocation points and relevant snapshots are selected on demand via LUPOD sampling to enhance the quality of computed modes. As a result, Galerkin projection is computationally cheap, robust, and well-suited to deal with non-homogeneity in space and time.

Now, some comments are needed to further clarify a few aspects of Algorithm 2:

- The threshold ε_N can take the same value in all applications of LUPOD throughout the whole simulation. Observe that, in order to speed up calculations, the input snapshots for the sampling technique can be considered at a reduced number of uniformly distributed mesh points (say, one out of each 5 or 10 points).
- Two numbers of modes, namely M and $M_1 > M$, are computed in line 8 as the smallest integers satisfying (see Eq. (3))

$$\mathrm{RMSE}_M^R < \varepsilon_M, \qquad \mathrm{RMSE}_{M_1}^R < \frac{\varepsilon_M}{100}, \tag{5}$$

  where the involved thresholds can be calibrated with flexibility (see related works as [41, 50]). More precisely, M and M_1 are calculated by the LUPOD function, namely in line 13 of Algorithm 1. Both numbers are needed for the error controls (line 12) defined below.
- The first error control (line 12) assisting the GS integration is

$$E_M^{M_1,j+1} = \sqrt{\sum_{m=M+1}^{M_1} |a_m^{j+1}|^2 \Big/ \sum_{m=1}^{M_1} |a_m^{j+1}|^2}, \tag{6}$$

  where a_m^{j+1} are the mode amplitudes at the $(j+1)$th time iteration. This control checks the reliability of the GS solution, since the right-hand side of Eq. (6) is a standard instantaneous, relative mode truncation error estimate, namely $E_M^{M_1,j+1} \simeq ||\boldsymbol{q}_{\mathrm{GS}}^{M_1,j+1} - \boldsymbol{q}_{\mathrm{GS}}^{M,j+1}|| \,/\, ||\boldsymbol{q}_{\mathrm{GS}}^{M_1,j+1}||$, where $\|\cdot\|$ is the norm associated with the reduced product in Eq. (1), which is based on the locally selected collocation points. Similar error estimates can be found in other spectral methods [25]. The parameter ε appearing in line 12 is thus the desired accuracy of the local approximation provided by the GS with M modes (which is stored in $\boldsymbol{S}_{\mathrm{GS}}$ for each I_{GS} in the Algorithm 2).
- The second control in line 12 involves an estimate of the discretized *residual* of the governing equations. Indeed, residual estimates play an important role in detecting mode truncation instabilities of systems obtained via Galerkin projection (see [39] and references therein). Hence, the threshold $\varepsilon/10$ appearing in line 12 reinforces local accuracy by preventing possible instabilities of the current GS with M modes. In the numerical experiments in Section 4, we will consider systems of the form

$$\partial_t \boldsymbol{q} = \boldsymbol{\mathcal{L}}\boldsymbol{q} + \boldsymbol{f}(\boldsymbol{q}, t), \tag{7}$$

in which $\mathcal{L}$ and $\boldsymbol{f}$ stand for linear and nonlinear terms. Upon discretization in time by the Crank-Nicolson plus Adams-Bashforth method [15], our discretized residual estimate is defined as

$$E_{\text{res}}^{M_1,j+1} = \frac{\left\| \frac{2}{\Delta t}(\boldsymbol{q}_{\text{GS}}^{M_1,j+1} - \boldsymbol{q}_{\text{GS}}^{M_1,j}) - \boldsymbol{F}_{\text{GS}}^{M_1,j} \right\|}{2\|\mathcal{L}\boldsymbol{q}_{\text{GS}}^{M_1,j}\| + 2\|\boldsymbol{f}^{j-1}\|}, \tag{8}$$

with

$$\boldsymbol{F}_{\text{GS}}^{M_1,j} = \mathcal{L}(\boldsymbol{q}_{\text{GS}}^{M_1,j+1} + \boldsymbol{q}_{\text{GS}}^{M_1,j}) + 3\boldsymbol{f}^{j} - \boldsymbol{f}^{j-1},$$

where $\boldsymbol{f}^j = \boldsymbol{f}(\boldsymbol{q}_{\text{GS}}^{M_1,j}, t_j)$ and Δt is the integration time step. The right-hand side of Eq. (8) corresponds to the scaled norm of the residual of the system in Eq. (7). Indeed, for the sake of simplicity in the illustration, the Crank-Nicolson plus Adams-Bashforth scheme will be used in the following for both the NS and the GS. If another time-integration method was used, a consequently different form of the residual would result. Regarding this issue, which is not the object of the present work, a discussion can be found in [39] and references therein.

- In the novel updating procedure applied in line 10, only "old" snapshots whose associated pivots (see Section 2) are larger than a threshold $\varepsilon_{\text{old}} > \varepsilon_N$ (e.g., $\varepsilon_{\text{old}} = 100\,\varepsilon_N$) need to be considered. Another option, in order to avoid calibrating ε_{old}, consists in taking into account the first M_1 (associated with the M_1 largest pivots) "old" snapshots. We thoroughly checked that results are very robust in connection with this issue, which is illustrated for one test case in Section 4.2. It is worth highlighting that the proposed updating strategy is linked to the essence of LUPOD: snapshots and pivots drive the sampling. Furthermore, it is easy to be implemented and computationally fast.
- A different updating process was used in related works [39, 41, 50]. According to it, "old" and "new" modes were mixed in order to update a POD basis, but no update of snapshots and involved points was done. Another drawback of the updating procedure used in former works is that modes must be appropriately weighted, which is somewhat subtle.
- An initial guess for the length of first and subsequent I_{NS} intervals is fixed as input for the algorithm (see Section 4 for specific values in our numerical experiments). Then, if needed, each length is dynamically adjusted (by adding new NS-computed snapshots), such that the length of the next I_{GS} interval is at least equal to a prescribed minimum value $\delta_{\text{GS,min}}$. The value of $\delta_{\text{GS,min}}$ can be taken as few times the characteristic timescale of the system, namely $\delta_{\text{GS,min}} = kT_0$, where k is a small positive integer and T_0 can be calculated by Eq. (10) below (see [39] for further details). This empirically results in larger I_{GS} intervals and less updating steps. A further discussion on the empirical calibration of the I_{NS} intervals length, which is not crucial for an efficient performance of the algorithm, can be found in [39] and references therein.
- The GS time-integration (line 13) may also be stopped if the length of the current I_{GS} interval is larger than a reference length $\delta_{\text{GS,max}}$. The latter helps

to eliminate older snapshots that are no longer necessary to describe the actual dynamics, thus improving both the quality of the POD basis and the overall performance of the iterative process. The value of $\delta_{\mathrm{GS,max}}$ can be selected as a fraction of the time interval where the dynamics are expected to reach their attractor, for instance $\delta_{\mathrm{GS,max}} = T/3$. Nevertheless, it is important to remark that, without imposing any maximum length $\delta_{\mathrm{GS,max}}$, the LUPOD-based method still works fine, with the only drawback of retaining few more modes than necessary in case the dynamics get simpler (namely, the dimension of the approximating linear subspace decreases) as time proceeds.

Summarizing, the adaptive LUPOD-assisted ROM is able to exploit spatio-temporal relationships of computed trajectories and describe, within a prescribed accuracy, the local structure of the solutions in terms of a limited number of "optimal" snapshots and collocation grid points. The proposed algorithm is the non-straightforward result of combining a former idea on alternating NS and GS runs with a new sampling technique and a new updating procedure. Hence, the fundamental novel elements of the present LUPOD-based method are the following:

- The snapshots for the on-demand application of POD are selected according to local temporal linear independence and concentrated temporal complexity. Hence, their distribution is *non-homogeneous* in time. This allows to capture fast and sporadic events that might be lost by a non-guided or uniform sampling of snapshots.
- The selected collocation mesh points, used to perform both POD and Galerkin projection, are chosen according to local spatial linear independence and concentrated spatial complexity. Hence, their distribution is non-uniform and *sparse* in the spatial domain. In this way, isolated peaked patterns are accurately recognized, while standard POD, which emphasizes spread structures with large L_2-norm, frequently overlooks them.
- Relevant states computed over the evolution of the system are stored to record mechanisms that may be useful for future approximations. Hence, the algorithm has a *memory* and uses the collected snapshots when needed.

Indeed, the synergistic combination of the above-mentioned ingredients of the method is highly non-trivial and results in the following: (i) enhanced reliability of the approximated transitions in the dynamics, (ii) extension to a wide range of systems showing concentrated spatio-temporal patterns (not accessible to former approaches), and (iii) faster computations. These advantages will be corroborated in the applications considered in Section 4. Finally, note that an extension of the methodology and the Algorithm 2 to problems as in Eq. (7) describing the evolution of a state variable over some parameter (instead of time) should be straightforward, thanks to the continuous weak dependence of the POD subspace on parameter values. In this context, similar ideas as those discussed and exploited in [51, 52] could be very useful.

4 Numerical experiments

In this section, we illustrate the performance of the proposed adaptive ROM for two well-known benchmark equations of pattern formation: the Kuramoto-Sivashinsky equation (Section 4.1) and the complex Ginzburg-Landau equation (Section 4.2 for the one-dimensional case and Section 4.3 for the two-dimensional case). Before starting with the numerical experiments, let us first introduce a few definitions.

The reconstructed solutions at all grid points, obtained from the low-dimensional dynamics computed via local GS in the LUPOD-based method, will be compared with those provided by the NS, which is regarded as "exact". Thus, since collocation points are selected among the mesh points used in the NS discretization, the following instantaneous, relative RMS error

$$E^j_{\mathrm{RMS}} = \frac{\|\boldsymbol{q}^{M,j}_{\mathrm{GS}} - \boldsymbol{q}^j_{\mathrm{NS}}\|_2}{\|\boldsymbol{q}^j_{\mathrm{NS}}\|_2} \tag{9}$$

measures the actual truncation error, retaining M modes, at the jth time step, and can be compared with the estimate $E^{M_1,j}_M$ in Eq. (6) in each I_{GS} interval. Note that, in Eq. (9), $\|\cdot\|_2$ is the usual Euclidean norm based on all grid points. Moreover, the error E^j_{RMS} will be set to zero in the I_{GS} intervals (imposing the current NS state to be equal to its GS counterpart) at time instants that are multiples of a quantity T_0 proportional to the characteristic timescale of the system. This is necessary when dealing with complex dynamics, in order to avoid a natural divergence of nearby trajectories due to small errors in the involved frequencies. Specifically, T_0 is defined in each I_{GS} interval as

$$T_0 = \frac{5}{N}\sum_{n=1}^{N} T_0^n, \quad \text{with} \quad T_0^n = 2\pi\sqrt{\frac{\int_{I_{\mathrm{GS}}} |\boldsymbol{q}^M_{\mathrm{GS}}(x_n,t)|^2 \mathrm{d}t}{\int_{I_{\mathrm{GS}}} |\partial_t \boldsymbol{q}^M_{\mathrm{GS}}(x_n,t)|^2 \mathrm{d}t}}, \tag{10}$$

where T_0^n is a measure of the timescale at the nth sampled collocation point in the I_{GS} interval. See, e.g., [52] for further details about T_0. On the other hand, the NS considered for all problems in this section is based on second-order, centered finite differences in a uniform grid of J spatial points and the Crank-Nicolson plus Adams-Bashforth temporal scheme [15]. Spatial step has been calibrated to guarantee that the relative RMS error in Eq. (9) between the NS solutions with J and $2J$ mesh points is smaller than $\varepsilon = 10^{-2}$ (which will be the accuracy required to the ROM), in time intervals of length T_0. Consequently, time step Δt has been chosen to ensure the numerical stability of the temporal scheme in the various cases considered below. Finally, the efficiency of simulating the system dynamics over $0 < t \leq T$, within accuracy ε, by means of the adaptive ROM will be measured by the *online acceleration factor*

$$C_{\mathrm{online}} = \frac{\text{NS CPU time}}{\text{ROM CPU time}}, \tag{11}$$

using a desktop personal computer, with an Intel i7–3.5GHz microprocessor and 8GB RAM.

As anticipated, the adaptive method described in Section 3 will be applied in the following pattern-forming systems: the one-dimensional Kuramoto-Sivashinsky equation (KSE), Eq. (12), subject to periodic boundary conditions, the one-dimensional complex Ginzburg-Landau equation (CGLE-1D) with drift, Eq. (18), satisfying homogeneous Dirichlet boundary conditions, and the standard two-dimensional complex Ginzburg-Landau equation (CGLE-2D), Eq. (20), also with homogeneous Dirichlet boundary conditions. The values of the coefficients for the various test cases that will be considered for these equations are given in Table 1, which also shows the length L of the spatial domain, the simulated timespan T, and the arithmetic mean of T_0 along the I_{GS} intervals. In order to assess the performance of the novel adaptive ROM, these representative cases cover a variety of different spatially localized patterns and temporal behaviors (periodic, quasi-periodic, and chaotic).

As discussed in Section 3, in order to efficiently approximate the dynamics within a desired accuracy ε, parameters of the adaptive LUPOD-based method that must be calibrated are ε_N, ε_M, $\varepsilon_{\mathrm{old}}$, and $\delta_{\mathrm{GS,min}}$ (additionally $\delta_{\mathrm{GS,max}}$ if a maximum length for the I_{GS} intervals is imposed). Results are rather robust in terms of their tunable values. A series of numerical experiments was carried out to check robustness, but they are not fully reported for the sake of brevity; instead, only a few of them corresponding to test case 1 for the CGLE-1D are presented in Section 4.2. Indeed, for each test case, the selected parameter values are not the optimal ones. We decided to fix the same values for all simulations to stress that only a rough calibration is needed to achieve efficiency. The number of initial snapshots in the first I_{NS} interval (K_{init}) and in subsequent ones (K) is also fixed for all test cases. Only the minimum and maximum lengths for the I_{GS} intervals vary from one case to another, in accordance with

Table 1 Values of the length L of the spatial domain, the simulated timespan T, and the equation coefficients for the considered test cases. The arithmetic mean of the values of T_0 computed along the I_{GS} intervals is also shown

	Domain		Equation coefficients					
	L	T	η	α	β	μ	c	T_0
KSE								
Case 1	2π	1	84.25					0.02
Case 2	2π	1	54					0.015
CGLE-1D								
Case 1	1	3		0.5	-7	85	0	0.05
Case 2	60	300		1	-2	0.3	1	3
Case 3	60	300		0.45	-2	2	1	3
CGLE-2D								
Case 1	1	3		2	-3.5	40		0.1
Case 2	1	3		3.5	-10	Eq. (21)		0.04

Table 2 Values for the number J of spatial discretization points, the time step Δt, the number of initial snapshots in the first I_{NS} interval (K_{init}) and in subsequent ones (K), the minimum ($\delta_{\mathrm{GS,min}}$) and maximum ($\delta_{\mathrm{GS,max}}$) lengths of the I_{GS} intervals, the desired accuracy ε, and the thresholds to select the number of modes (ε_M), relevant snapshots and collocation points (ε_N), and "old" snapshots ($\varepsilon_{\mathrm{old}}$)

	J	Δt	K_{init}	K	$\delta_{\mathrm{GS,min}}$	$\delta_{\mathrm{GS,max}}$	ε	ε_M	ε_N	$\varepsilon_{\mathrm{old}}$
KSE										
Cases 1,2	1001	$5\cdot10^{-6}$	100	3	0.05	0.3	10^{-2}	10^{-3}	10^{-7}	10^{-5}
CGLE-1D										
Case 1	1001	$5\cdot10^{-5}$	100	3	0.1	1	10^{-2}	10^{-3}	10^{-5}	10^{-3}
Cases 2,3	1001	$5\cdot10^{-4}$	100	3	10	60	10^{-2}	10^{-3}	10^{-5}	10^{-3}
CGLE-2D										
Cases 1,2	251^2	$5\cdot10^{-5}$	100	3	0.1	1	10^{-2}	10^{-3}	10^{-5}	10^{-3}

the length of the simulated timespan T and the computed T_0. Considered values of the tunable parameters, details about the discretization, and the desired accuracy are reported in Table 2 for the pattern-forming systems and test cases given in Table 1.

4.1 Application to the one-dimensional Kuramoto-Sivashinsky equation

Let us consider the following modified *Kuramoto-Sivashinsky* equation (KSE) with periodic boundary conditions

$$\partial_t q = -4\,\partial^4_{xxxx} q - \eta \partial^2_{xx} q - \frac{\eta}{2}(\partial_x q)^2 + \frac{\eta}{4\pi}\int_0^{2\pi} (\partial_x q(x,t))^2\,\mathrm{d}x, \tag{12}$$

$$q(x+2\pi,t) = q(x,t),\ \ \partial_x q(x+2\pi,t) = \partial_x q(x,t),\ \ \partial^2_{xx} q(x+2\pi,t) = \partial^2_{xx} q(x,t), \tag{13}$$

for $0 < x < 2\pi$, where q is the real state variable and η is a positive parameter. Equation (12) is obtained from the standard KSE,

$$\partial_t q = -4\,\partial^4_{xxxx} q - \eta \partial^2_{xx} q - \frac{\eta}{2}(\partial_x q)^2, \tag{14}$$

upon the independent variable change

$$q \to q - \frac{1}{2\pi}\int_0^{2\pi} q(x,t)\,\mathrm{d}x, \tag{15}$$

which eliminates the unbounded growth of q as time proceeds by subtracting from q its spatial mean value. The standard KSE in Eq. (14) is a well-known pattern forming equation [19] that has been extensively studied in the literature, since the pioneering work by Kuramoto [31] and Sivashinsky [46], who independently derived the equation as a "normal form" at the onset of the modulational instability and the cellular instability of plane fronts, respectively. Here, for convenience, we consider the formulation in Eqs. (12)–(13) that maintains q bounded but adds the last term in the right-hand side of Eq. (12), which represents a *mean drift* in the mean value of q.

Such drift will be computed by means of the composite trapezoidal rule, which is quite convenient for periodic functions [8].

The problem in Eqs. (12)–(13) is invariant under the actions $x \to -x$ and $x \to x + c$ ($c \in \mathbb{R}$), corresponding to spatial reflection and translations, respectively. The initial condition

$$q(x, 0) = \cos(x) + \sin(2x) + \sin(3x) + \cos(4x) \tag{16}$$

has been selected as non-reflection-symmetric, in order to avoid restriction to an invariant subspace. As mentioned in [28], the solutions to the KSE are characterized by the coexistence of temporal chaos with coherent spatial structures. Depending on the value of the parameter η, the attracting solution manifolds can be either unimodal, multimodal or chaotic (see [28, Table I]).

In order to simulate the system dynamics over the time interval $0 < t \leq 1$ for the two illustrative cases anticipated in Table 1, we set the discretization and the adaptive ROM parameter values detailed in the first row of Table 2. In addition, all snapshots are computed by the NS at temporal distance equal to $10\,\Delta t$ and their values at 200 equispaced points are given as input to LUPOD to speed up calculations. It is worth noting that such distance is suitable for all simulated dynamics in the paper. Indeed, there are test cases in which it can be enlarged. However, as previously mentioned, our intention is to consider the same method's parameter values for all simulations, in order to emphasize that choosing optimal values is not essential for a good performance.

The first test case corresponds to $\eta = 84.25$. It was previously studied in [1], where the authors propose a modal reduction method combined with a procedure for snapshots selection, finding out that the attracting solution manifold can be described in terms of five modes. After a transient of length approximately equal to 0.25, the

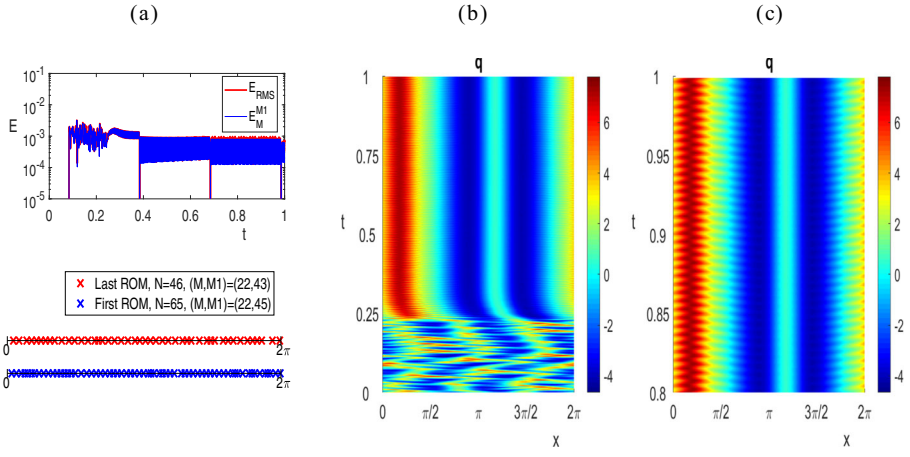

Fig. 1 Test case 1 for the KSE in Eq. (12), with $\eta = 84.25$ and initial condition as in Eq. (16). **a** Error E_{RMS} in Eq. (9) and estimate $E_M^{M_1}$ in Eq. (6) (upper plot); selected collocation points for the first and last I_{GS} intervals (lower plot). **b** Spatio-temporal color map of q for $0 < t \leq 1$. **c** Blow-up of plot (**b**) showing the spatio-temporal color map of q for $0.8 \leq t \leq 1$

behavior of the solution changes, resembling an asymptotic periodic attractor, as seen in Fig. 1b. Although not clearly appreciated in this plot, the solution for $t > 0.25$ becomes simpler but is highly oscillatory, as can be seen in Fig. 1c, where the spatio-temporal color map of q is plotted for the shorter time interval $0.8 \leq t \leq 1$. The E_{RMS} and $E_M^{M_1}$ errors for the parameter set indicated in Table 2 are depicted in the upper plot of Fig. 1a. The I_{NS} intervals are easily identified since there both errors are set to zero. In the remaining intervals, namely the I_{GS} intervals, we observe that the errors compare very well and maintain below the desired accuracy $\varepsilon = 10^{-2}$. The first I_{NS} interval, of approximate length 0.08, provides an initial basis of $M = 22$ modes (with $M_1 = 45$ for the estimate $E_M^{M_1}$ in Eq. (6)), computed by using $N = 65$ collocation points and relevant snapshots selected by LUPOD. This basis is able to properly describe the solution behavior up to $t \simeq 0.4$, thanks to the adaptive temporal sampling that captures the sporadic events characterizing the initial transient detours of the trajectory. At $t \simeq 0.4$ an update is performed in a very short I_{NS} interval. From that time instant on, the dynamics are simpler and only $N = 46$ collocation points are required for the ROM. Note that a new updating is needed (and performed) at $t \simeq 0.68$ due to the parameter $\delta_{\text{GS,max}}$. The distribution of the sampled collocation points in the first and last I_{GS} intervals is given in the lower plot of Fig. 1a. The latter is quite homogeneous in a spatial period, due to the influence of the transient where complexity is fairly even over it. The online acceleration factor resulted from applying the LUPOD-based method is $C_{\text{online}} = 11.83$.

Now, we simulate only the attractor, which is done by integrating the KSE using as initial condition the state (provided by the NS) in the former simulation at $t = 5$. In the new simulation, we observe that (i) the length of the first I_{NS} interval is much shorter (about 0.025) and (ii) the number of collocation points reduces to $N = 30$ in the first I_{GS} interval and $N = 19$ in subsequent ones. Furthermore, sampled points concentrate in regions where the solution presents the steepest gradients, as can be observed in Fig. 2. Besides, the number of required modes is $M = 5$, in agreement

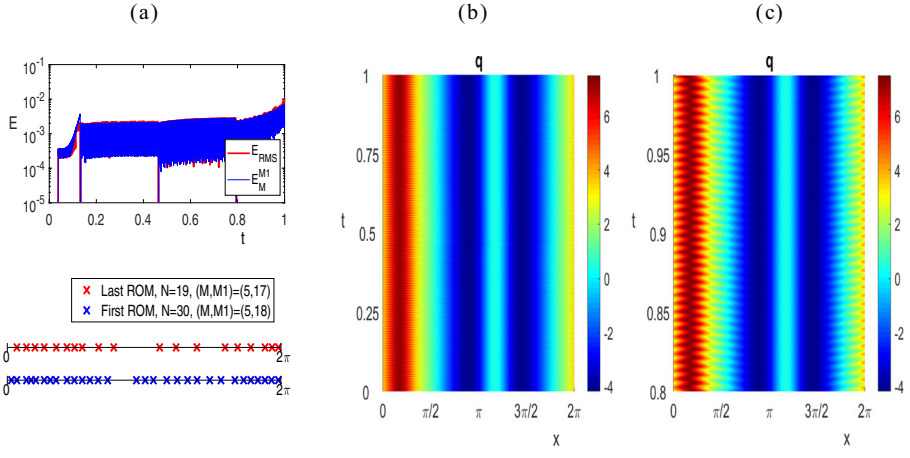

Fig. 2 Counterpart of Fig. 1, but simulating the attractor, as explained in the text

with what observed in [1]. The acceleration factor is $C_{\text{online}} = 32.14$, meaning that the novel adaptive ROM is about 32 times faster than the considered NS.

Similarly as we did in the previous case, let us now initiate the integration of the transient dynamics by the initial condition

$$q(x,0) = \cos(x+3\pi/2)+\sin(2(x+3\pi/2))+\sin(3(x+3\pi/2))+\cos(4(x+3\pi/2)), \tag{17}$$

which is just a spatial translation of the one in Eq. (16). Using as initial condition for the simulation of the attractor the NS state at $t = 5$ from the transient simulation, the counterpart of Fig. 2 is obtained, as given in Fig. 3. The numbers of required collocation points and modes are essentially the same as before, as well as the resulting online acceleration factor. However, the collocation points selected by LUPOD accumulate in a different region (exactly where the steepest gradients appear in the new case). This is in agreement with the essence of the sampling technique. Likewise, the spatio-temporal pattern (extended by periodicity to the whole x line) is a spatial translation of the previous case, which is consistent with the invariance of the problem under spatial translations.

Figures 2 and 3 illustrate fairly well the ability of the LUPOD-based ROM to automatically concentrate collocation points precisely where they are needed, namely in those regions where localized complexity occurs. In the KSE in Eq. (12), the latter is spontaneously promoted through instabilities and its location depends on initial conditions in an intricate way, difficult to be guessed a priori. Thus, a manual selection of the convenient collocation points distribution is not possible in this case, as in many others.

In the second, more demanding test case, we consider the KSE in Eq. (12) with $\eta = 54$, together with the initial condition as in Eq. (16). The dynamics are chaotic, and the obtained results are displayed in Fig. 4. As can be seen, solutions exhibit concentrated spatio-temporal complexity in spatial regions that travel left and right

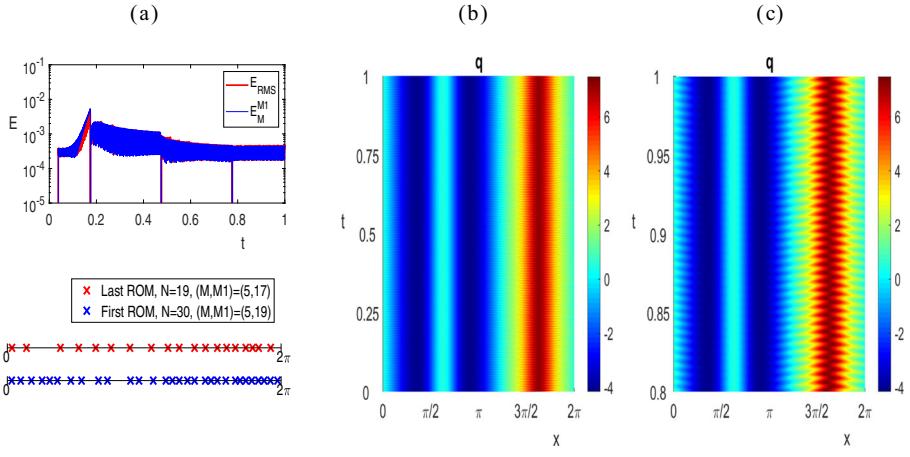

Fig. 3 Counterpart of Fig. 2, but using the initial condition in Eq. (17) to initiate the transient simulation that gives the initial condition for the present simulation

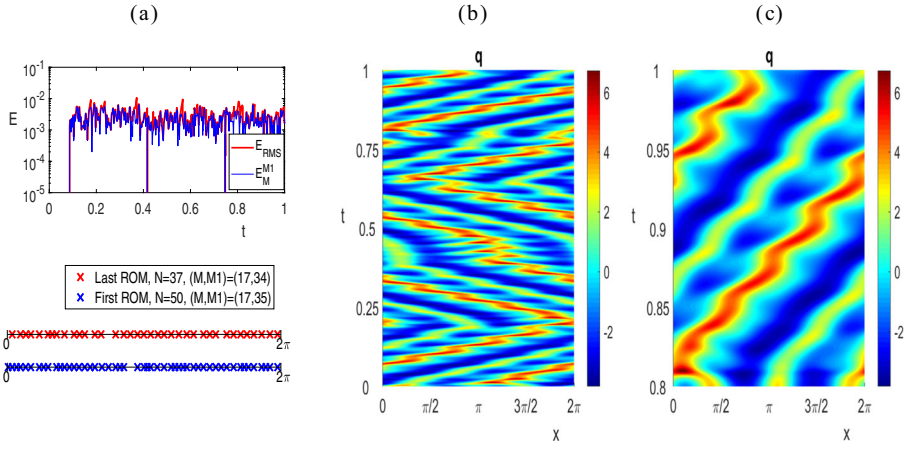

Fig. 4 Counterpart of Fig. 1 for test case 2 of the KSE in Eq. (12), with $\eta = 54$ and initial condition as in Eq. (16)

as time proceeds. Since complexity is not confined, neither in specific spatial regions nor in specific temporal regions, LUPOD does not sample collocation points and snapshots in any particular space or time intervals. The numbers of retained modes for all I_{GS} intervals are $(M, M_1) \simeq (17, 34)$. Indeed, since the complexity of the oscillating patterns is high but similar in the whole timespan $0 < t \leq 1$, such numbers do not essentially change over the evolution. In the first I_{GS} interval, $N = 50$ collocation points are needed, while this number slightly reduces to $N = 37$ in the last one, being almost identical to the number M_1 of modes (recall that necessarily $N \geq M_1$). The online acceleration factor is $C_{\mathrm{online}} = 10.98$. By considering different initial conditions (whose results are not shown for the sake of brevity), we observe a similar behavior: the numbers of modes are again $(M, M_1) \simeq (17, 34)$, the amount of collocation points reduces from the first to the last I_{GS} interval, and the online acceleration factor approximately coincides with that for the reference initial condition.

4.2 Application to the one-dimensional complex Ginzburg-Landau equation with drift

Let us consider the following one-dimensional complex Ginzburg-Landau equation (CGLE-1D) with drift

$$\partial_t q = (1 + \mathrm{i}\alpha)\partial_{xx}^2 q + c\,\partial_x q + \mu\, q - (1 + \mathrm{i}\beta)|q|^2 q \tag{18}$$

for $0 < x < L$, with homogeneous Dirichlet boundary conditions. The state variable q is complex, while coefficients α, c, μ, and β are real. If $c = 0$ (no drift), then Eq. (18) is the standard CGLE-1D. The latter equation stands for the weakly non-linear "normal form" describing the onset of oscillatory instabilities in many spatially uniform systems [27]. Such equation is a well-known paradigm of pattern-forming systems, which may give rise to intrinsically complex dynamics due to the

modulational instability if $\alpha\beta < -1$ and μ is larger than a threshold value [6, 19]. The standard CGLE-1D is reflection-symmetric around the midpoint of the spatial domain, $x = L/2$. However, the drift term breaks reflection-symmetry if $c \neq 0$. With the drift term, Eq. (18) describes the envelope of a traveling wave that propagates in one direction at group velocity equal to c [53].

In the three test cases considered below for Eq. (18), the initial condition is

$$q(x, 0) = \mathrm{i}\sin(2\pi x/L) + (1 + \mathrm{i})\sin(3\pi x/L), \tag{19}$$

which was chosen to be far from the final attractor and non-reflection-symmetric around $x = L/2$, which avoids restriction of the dynamics to an invariant subspace when the equation has that symmetry, as in test case 1.

The parameter values for test case 1 are given in Table 1, which shows that the drift is absent ($c = 0$). Also, the spatial domain is $0 < x < 1$ and the system is simulated over the timespan $0 < t \leq 3$. The adaptive LUPOD-based ROM is constructed using the tunable parameter values shown in Table 2. As clearly visible in Fig. 5b and c, for the present test case dynamics adopt a chaotic behavior. As a consequence, a few updates of the initial POD basis with $(M, M_1) = (11, 19)$ modes are needed, though requiring very short I_{NS} intervals. On one hand, fast temporal fluctuations are properly described by a non-homogeneous sampling that identifies the most linearly independent snapshots. On the other hand, strong spatial variations around $x = 0.5$ are dynamically captured, within the desired accuracy $\varepsilon = 10^{-2}$, by a collection of selected collocation points that tend to gather about the center of the domain. Thanks to the "optimal" distribution yielded by LUPOD, their number can be fairly small, e.g., $N = 19$ (only 2% of discretization mesh points) in the last I_{GS} interval. This yields an online acceleration factor equal to $C_{\mathrm{online}} = 8.62$. In this test case, the updating procedure retaining the first M_1 (associated with the

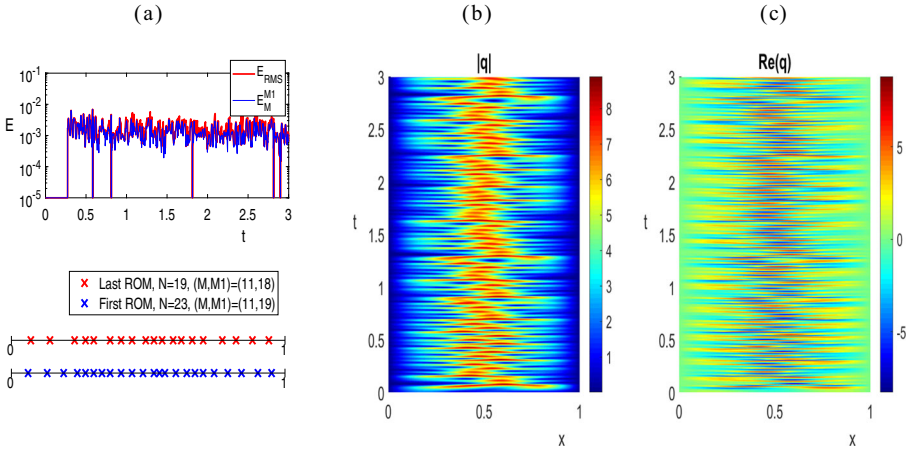

Fig. 5 Test case 1 for the CGLE-1D in Eq. (18), with $c = 0$. **a** Error E_{RMS} in Eq. (9) and estimate $E_M^{M_1}$ in Eq. (6) (upper plot); selected collocation points for the first and last I_{GS} intervals (lower plot). **b** Spatio-temporal color map of $|q|$. **c** Spatio-temporal color map of $\mathrm{Re}(q)$

Table 3 Numbers N_{init} and N_{end} of selected collocation points/snapshots in the initial and final I_{GS} intervals, numbers $(M, M_1)_{\text{init}}$ and $(M, M_1)_{\text{end}}$ of retained modes in the initial and final I_{GS} intervals, and online acceleration factor C_{online} for different values of the thresholds to select the number of collocation points/snapshots (ε_N), "old" snapshots (ε_{old}), modes (ε_M), and the minimum length for the I_{GS} intervals ($\delta_{\text{GS,min}}$), when the desired accuracy is $\varepsilon = 10^{-2}$ in test case 1 for the CGLE-1D

ε_N	ε_{old}	ε_M	$\delta_{\text{GS,min}}$	N_{init}	N_{end}	$(M, M_1)_{\text{init}}$	$(M, M_1)_{\text{end}}$	C_{online}
10^{-5}	10^{-3}	10^{-3}	0.1	23	19	(11,19)	(11,18)	8.62
10^{-5}	10^{-4}	10^{-3}	0.1	23	21	(11,19)	(12,19)	9.17
10^{-6}	10^{-3}	10^{-3}	0.1	28	24	(11,19)	(12,20)	7.43
10^{-6}	10^{-4}	10^{-3}	0.1	28	23	(11,19)	(12,20)	9.08
10^{-5}	10^{-3}	10^{-4}	0.1	21	19	(14,21)	(14,19)	12.50
10^{-5}	10^{-4}	10^{-4}	0.1	21	22	(14,21)	(14,22)	12.32
10^{-6}	10^{-3}	10^{-4}	0.1	27	20	(15,22)	(15,20)	10.69
10^{-6}	10^{-4}	10^{-4}	0.1	27	22	(15,22)	(15,21)	9.85
10^{-5}	10^{-3}	10^{-3}	0.2	23	19	(12,20)	(12,19)	7.61
10^{-5}	10^{-4}	10^{-3}	0.2	23	21	(12,20)	(12,20)	7.06
10^{-6}	10^{-3}	10^{-3}	0.2	28	28	(12,20)	(12,20)	6.67
10^{-6}	10^{-4}	10^{-3}	0.2	28	22	(12,20)	(12,20)	7.78
10^{-5}	10^{-3}	10^{-4}	0.2	22	18	(15,21)	(15,18)	11.89
10^{-5}	10^{-4}	10^{-4}	0.2	22	21	(15,21)	(15,21)	11.51
10^{-6}	10^{-3}	10^{-4}	0.2	27	20	(15,22)	(15,20)	10.36
10^{-6}	10^{-4}	10^{-4}	0.2	27	22	(15,22)	(15,21)	9.70

M_1 largest pivots) "old" snapshots yields completely analogous results, namely the same number of updates, similar numbers of sampled collocation points and retained modes, and $C_{\text{online}} = 8.57$.

As already mentioned at the beginning of Section 4, results are very robust in terms of the tunable parameters of the method. For illustration, we summarize in Table 3 the effect of varying their values for the desired accuracy $\varepsilon = 10^{-2}$, considering $\varepsilon_N \in \{10^{-5},\ 10^{-6}\}$ for collocation points/snapshots selection, $\varepsilon_{\text{old}} \in \{10^{-3},\ 10^{-4}\}$ for choosing "old" snapshots, $\varepsilon_M \in \{10^{-3},\ 10^{-4}\}$ for retaining modes, and $\delta_{\text{GS,min}} \in \{0.1,\ 0.2\}$ for the minimum length of the I_{GS} intervals. These parameter values provide rather similar results for the numbers N_{init} and N_{end} of collocation points/snapshots in the initial and final I_{GS} intervals, and the numbers of retained modes in the initial and final I_{GS} intervals, $(M, M_1)_{\text{init}}$ and $(M, M_1)_{\text{end}}$, respectively. In addition, the online acceleration factor C_{online} does not drastically change in all cases, which means that the method does not require a subtle parameter calibration for a good performance. Indeed, our baseline selection $\varepsilon_N = 10^{-5}$, $\varepsilon_{\text{old}} = 10^{-3}$, $\varepsilon_M = 10^{-3}$, and $\delta_{\text{GS,min}} = 0.1$ is not optimal for this test case.

The effect of the drift term on the CGLE-1D dynamics is studied, considering Eq. (18) with $c = 1$ and the other coefficients shown in Table 1 for test cases 2 and 3.

The tunable parameter values for the adaptive ROM are given in Table 2. Notice that, in comparison with the previous simulation, the space and time intervals are larger. Accordingly, some parameter values for the ROM have been modified, as already mentioned at the beginning of this section.

The spatio-temporal plots for test case 2 shown in Fig. 6b and c reveal that, after a short chaotic-like transient, the solution becomes simpler, being a pure traveling wave in most part of the spatial domain, except for the vicinity of the end-points, $x = 0$ and $x = 60$. The temporal structure is clearly captured by the LUPOD-assisted method: in the first I_{GS} interval, the required numbers of collocation points and modes are $N = 72$ and $(M, M_1) = (31, 55)$, respectively, while they reduce to $N = 24$ and $(M, M_1) = (3, 22)$, respectively, in the last I_{GS} interval. Thus, only three POD modes are needed to approximate the asymptotic dynamics, one describing the central traveling wave and two additional modes that account for the behavior near the end-points. Adaptivity leads to a simplified, three-dimensional model of the problem at large time, which very strongly alleviates the computational effort after the initial chaotic stage. The online acceleration factor is equal to $C_{\mathrm{online}} = 4.46$.

Finally, Fig. 7 shows the results for test case 3, where the solution exhibits chaotic-like spatio-temporal patterns only in the range $0 \leq x \leq 50$ for all $t > 0$, while in the remaining part of the domain the dynamics are much simpler. Consequently, LUPOD-sampled collocation points tend to concentrate in the region where the spatio-temporal structure is richer; only very few of them are located close to the right boundary, where the dynamics are simpler. Overall, spread complexity dominates and induces the novel adaptive LUPOD-based algorithm to deal with quite large numbers of relevant snapshots, required modes, and collocation points, as can be observed in the bottom plot of Fig. 7a.

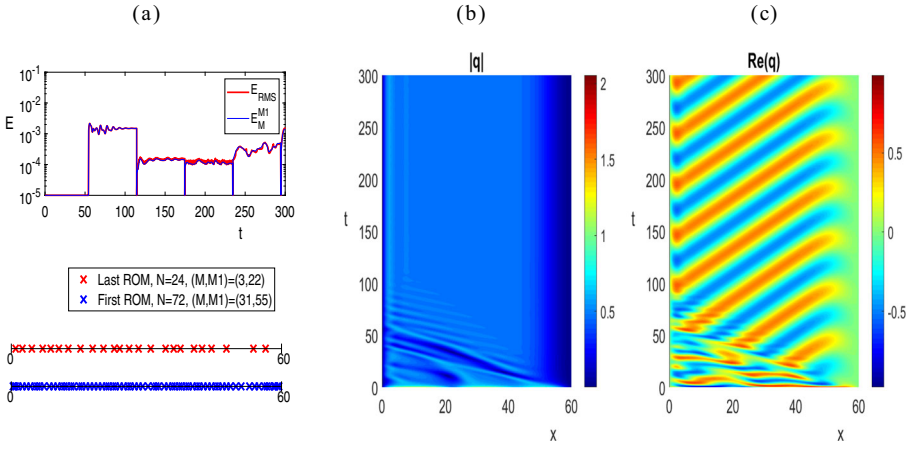

Fig. 6 Counterpart of Fig. 5 for test case 2 of the CGLE-1D in Eq. (18) with drift

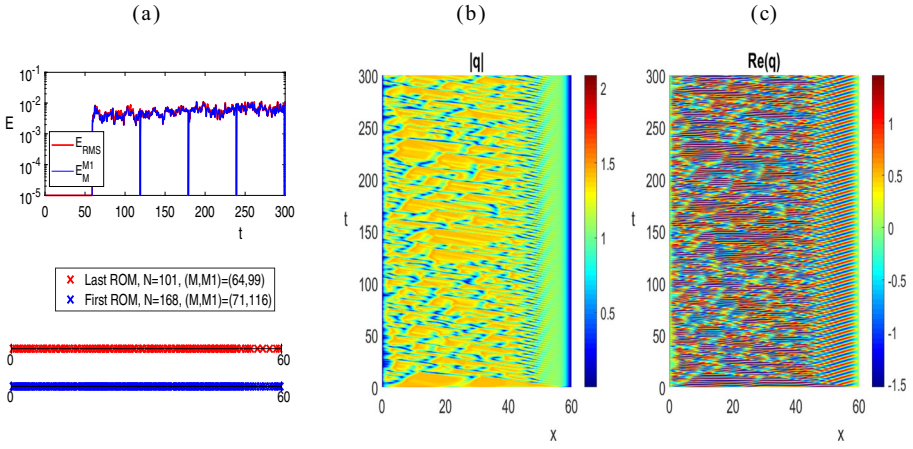

Fig. 7 Counterpart of Fig. 5 for test case 3 of the CGLE-1D in Eq. (18) with drift

4.3 Application to the two-dimensional complex Ginzburg-Landau equation with spatially localized complexity

Let us now consider the following two-dimensional complex Ginzburg-Landau equation (CGLE-2D)

$$\partial_t q = (1 + \mathrm{i}\alpha)\triangle q + \mu\, q - (1 + \mathrm{i}\beta)|q|^2 q \tag{20}$$

for $(x, y) \in (0, 1) \times (0, 1)$, with homogeneous Dirichlet boundary conditions. Again, the state variable q is complex and the parameters α, μ, and β are real. Now, for the sake of brevity, drift is not included since its effect on both the dynamics and the performance of the adaptive ROM has been already illustrated in the last subsection for the one-dimensional version of the equation. Instead, as a novel ingredient, we will consider either a constant linear growth coefficient μ or, in order to enhance spatially localized complexity, a space-dependent function

$$\mu(x, y) = -1 + \rho\, e^{-\gamma(x-x_0)^2-\gamma(y-y_0)^2}, \tag{21}$$

where ρ, γ, x_0, $y_0 \in \mathbb{R}$. Applications of Eq. (20) with non-constant linear growth range from non-linear optics to autocatalytic reactions and biological processes (see, e.g., [29, 32, 34] and references therein).

The initial condition is set to

$$\begin{aligned} q(x, y, 0) = {} & (1 + 7\mathrm{i})(x - 3y)\sin(2\pi x)\sin(\pi y) + \\ & (2 + \mathrm{i})(2x + y)\sin(\pi x)\cos(\pi/2 + \pi y), \end{aligned} \tag{22}$$

while the adaptive LUPOD-assisted ROM is run over the timespan $0 < t \leq 3$, with the same representative tunable parameter values as for the CGLE-1D with $c = 0$. This selection emphasizes robustness with respect to calibration (compare the second and fourth rows in Table 2).

The solution for test case 1 (see Table 1) rapidly converges to an attractor with steady absolute value, which is invariant under the actions (reflection-symmetries)

$x \to 1-x$, $y \to 1-y$, and $x \leftrightarrow y$. Instead, the large-time evolution of $\mathrm{Re}(q)$ is periodic. This behavior can be appreciated in Fig. 8, which shows the two-dimensional distribution of $|q|$ and $\mathrm{Re}(q)$ at three time instants, corresponding to the end of the involved I_{NS} intervals where the POD basis is generated and updated two different times. Concerning the performance of the adaptive LUPOD-based ROM, as in the previous cases, the actual approximation error is well estimated by the algorithm and keeps below the desired accuracy $\varepsilon = 10^{-2}$, as shown in Fig. 9a. Collocation points are mainly selected in the central region of the domain and an "optimal" number is achieved along the simulation, since $N = M_1 = 21$ in the last I_{GS} interval. Furthermore, they are quite spread due to the fairly smooth spatial structure of the bell-shaped solution, as visible in Fig. 9b.

Figure 10 illustrates three snapshots for test case 2, in a similar fashion as in the previous case. Now, a localized spatial pattern appears in the periodic dynamics due to a non-constant growth term that behaves like a Gaussian, given by Eq. (21) with $\rho = 240$, $\gamma = 20$, $x_0 = 0.25$, and $y_0 = 0.25$. Thus, most of the collocation points sampled by LUPOD cluster in the left-lower corner of the domain where a "pulsating" peak in the solution is clearly visible, as Figs. 11 and 10(b) evidence. It is remarkable that the number of such points is considerably small (for instance, $N = 69$ for the last constructed GS) in comparison with the total number of discretization grid points, which is equal to 63001. The number of retained modes oscillates around $(M, M_1) = (30, 65)$. This is a clear example underlining the reason why the novel LUPOD-assisted adaptive ROM is much more advantageous when the

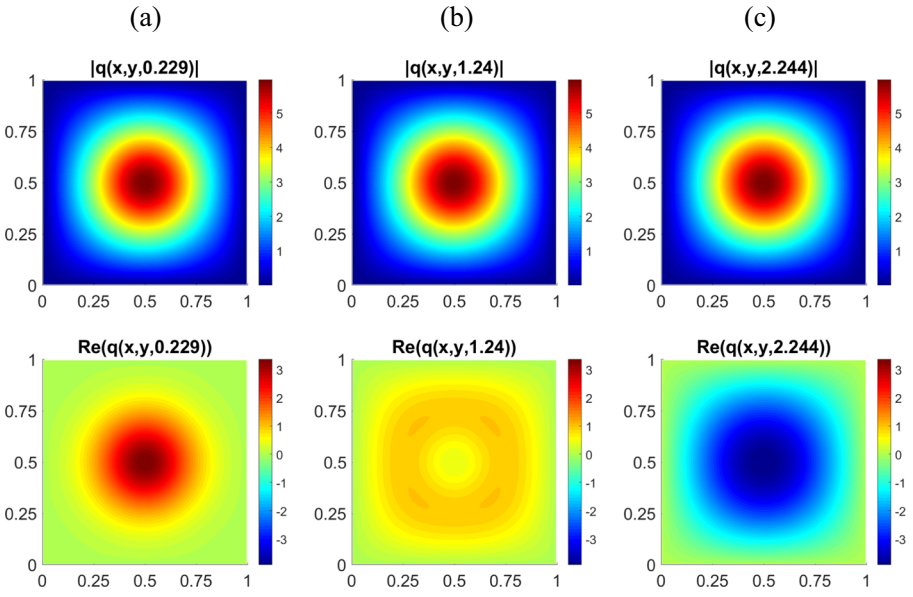

Fig. 8 Test case 1 for the CGLE-2D in Eq. (20), with constant μ. Color maps of $|q|$ (upper plots) and $\mathrm{Re}(q)$ (lower plots) at three different time instants at the end of the first (**a**), second (**b**), and third (**c**) I_{NS} intervals

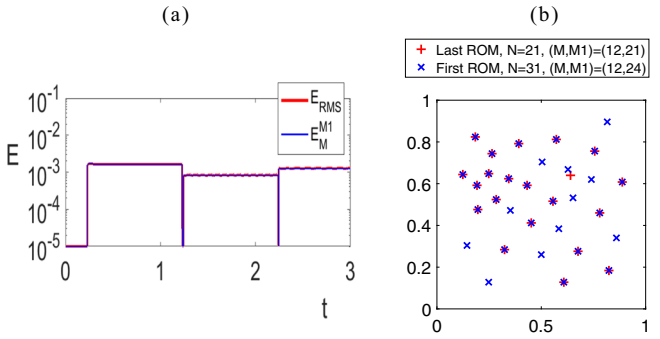

Fig. 9 Test case 1 for the CGLE-2D in Eq. (20), with constant μ. **a** Error E_{RMS} in Eq. (9) and estimate $E_M^{M_1}$ in Eq. (6). **b** Selected collocation points for the first and last I_{GS} intervals

spatial dimension increases, as further checked by means of other simulations (not reported for the sake of brevity) varying the parameter values for the function $\mu(x, y)$ in Eq. (21).

Actually, the online acceleration factor in Eq. (11) scales with the ratio of the number of numerical degrees of freedom (i.e., mesh points) to the number of physical degrees of freedom (i.e., retained modes). Hence, C_{online} is larger in the present two-dimensional case than in the one-dimensional case considered in the previous subsection. Specifically, $C_{\mathrm{online}} = 12.29$ and 23.81 in test cases 1 and 2, respectively. Also, notice that these values are roughly in inverse proportion to the length of the first I_{NS} interval, which is 0.229 in test case 1 and 0.12 in test case 2, meaning

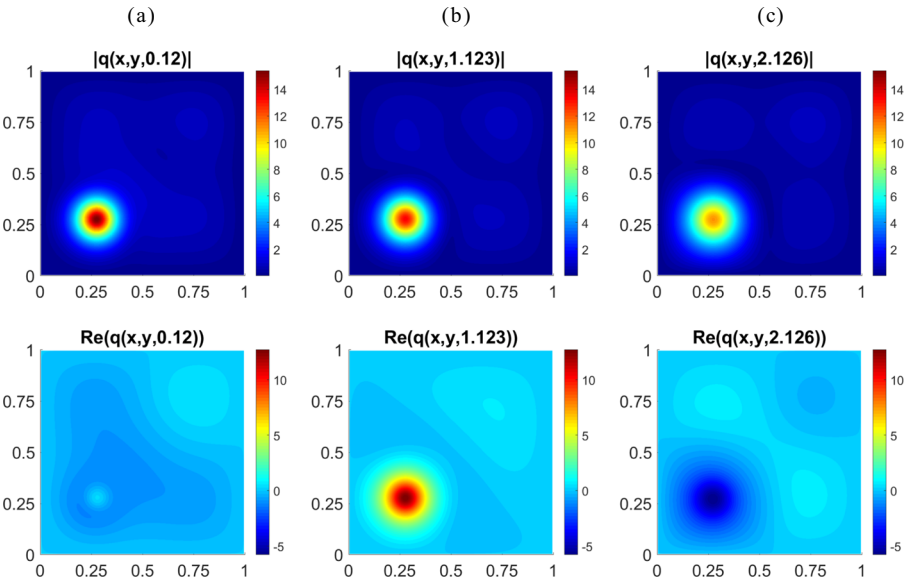

Fig. 10 Counterpart of Fig. 8 for test case 2 of the CGLE-2D in Eq. (20), with μ as in Eq. (21)

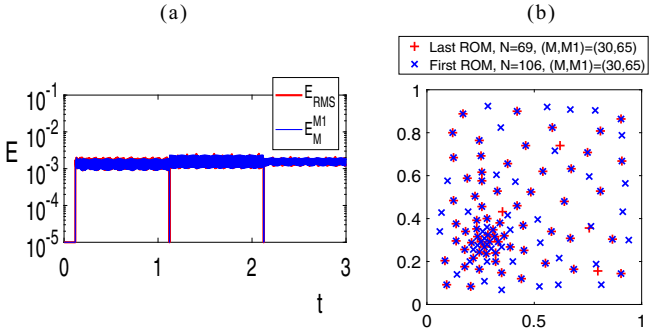

Fig. 11 Counterpart of Fig. 9 for test case 2 of the CGLE-2D in Eq. (20), with μ as in Eq. (21)

that computational cost of performing LUPOD plus GS integration is almost negligible in comparison with the required NS effort. Indeed, the online acceleration factor is approximately equal to the timespan T divided by the sum of the required I_{NS} intervals lengths, which results in 12.93 for test case 1 and 24.39 for test case 2.

To compare the results for points selection when applying LUPOD and DEIM methods, we apply both strategies to the snapshots in the first I_{NS} interval, namely $[0, 0.12]$, in test case 2 for the CGLE-2D. Figure 12 shows that the sampled sets of points are different, even if their distributions follow similar patterns. In this plot, LUPOD-selected points are the 106 initial ones already shown in Fig. 11b, while DEIM-selected points are the first 106 ones computed by Algorithm 1 in [17]. Replacing LUPOD with DEIM in our adaptive ROM would increase the computational cost, since no snapshots selection is performed by DEIM and therefore POD should be carried out by considering all the snapshots (or a calibrated number of equispaced or randomly selected snapshots). However, DEIM also yields a set of modes that can be used for Galerkin projection, even if they are unrelated with the modes computed via LUPOD. Developing an alternative version of our adaptive method based on DEIM is not straightforward. We refer to [37] for a different strategy involving adaptive DEIM.

Finally, Table 4 summarizes the actual computation times (CPU seconds computed by the MATLAB command `cputime`) when applying the standard numerical solver (NS_{time}) or the ROM (ROM_{time}), together with the online acceleration factor C_{online}, for all test cases considered in Section 4.

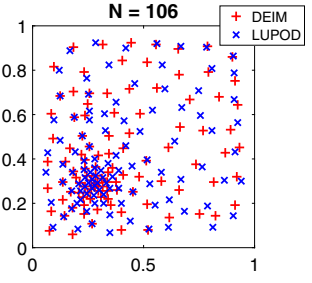

Fig. 12 Comparison between the collocation points selected by LUPOD and DEIM methods, when applied to a set of snapshots in test case 2 for the CGLE-2D

Table 4 Computation times in CPU seconds $\mathrm{NS}_{\mathrm{time}}$ (standard numerical solver) and $\mathrm{ROM}_{\mathrm{time}}$ (ROM), and online acceleration factor C_{online} for all test cases in Section 4

	$\mathrm{NS}_{\mathrm{time}}$	$\mathrm{ROM}_{\mathrm{time}}$	C_{online}
KSE			
Case 1	9760.3	825.1	11.83
Case 1 – attractor	9760.3	303.7	32.14
Case 2	9760.3	888.9	10.98
CGLE-1D			
Case 1	42.82	4.96	8.62
Case 2	494.6	110.89	4.46
Case 3	494.6	208.69	2.37
CGLE-2D			
Case 1	66390	5403	12.29
Case 2	66390	2788	23.81

5 Final remarks

An adaptive, collocated POD-based reduced order model has been proposed to efficiently simulate transients and time-dependent attractors in non-linear pattern-forming systems. According to the novel method, short runs of a standard numerical solver collect local information on the dynamics, then translated to a set of POD modes, and alternate with the integration of a low-dimensional Galerkin system. Adaptivity is achieved, in both space and time, by dynamically selecting along the simulation sets of relevant snapshots and collocation points, which are used to compute appropriate sets of POD modes. On the one hand, the relevant snapshots are (i) non-homogeneously selected according to local temporal linear independence and concentrated temporal complexity, (ii) stored to record underlying mechanisms of the system, and (iii) used for the on-demand application of POD. This allows to capture fast and transient oscillations. On the other hand, the collocation points are (i) sparsely sampled according to local spatial differences and concentrated spatial complexity, (ii) stored to record patterns in the domain, and (iii) used to both perform POD and Galerkin-project the governing equations in a collocation-like, spectral framework. Thus, isolated and localized structures with steep gradients or higher order derivatives are accurately described. Indeed, such spatio-temporal adaptation was absent in previous approaches, in which only POD modes were updated, in a different way, by means of uniformly distributed snapshots and equally spaced grid points, fixed throughout a whole simulation. Instead, the main result in the present work is a powerful algorithm to effectively approximate evolving patterns via time-dependent sets of snapshots, collocation points, and modes.

Sampling is performed by the LUPOD technique [40], which is a combination of Gauss elimination with POD and does not add any significant computational cost to the standard POD effort. LUPOD selects snapshots and collocation points according

to the dynamics of the problem, accounting for the most different information. Thus, also for large snapshot matrices, the cost required by the sampling is typically compensated by a higher computational efficiency of the POD step, which now involves an "optimal" (namely, not larger than necessary) number of spatial points and snapshots, especially for systems exhibiting concentrated spatio-temporal complexity. For large snapshot matrices, however, it would be desirable to parallelize the LUPOD method, which in principle should be simple, since LUPOD is based on computing the maximum value of a matrix (an easy calculation to parallelize) and performing a Gauss elimination (again, easy to parallelize for being a row-wise operation). The current work corroborates that, when applied on demand to perform adaptive low-dimensional modeling of dynamical systems, it leads to a collocation method that is computationally cheap, robust, and well-suited to deal with heterogeneity in space and time. Indeed, the synergistic combination of the above-mentioned ingredients is highly non-trivial and entails (i) enhanced reliability of the approximated transitions and (ii) extension to a wide range of systems showing localized patterns, not accessible to previous approaches by the authors. Time derivative snapshots could also have been used with the aim of reducing the computational complexity or improving the accuracy of the reduced order model, as proposed in [17] for the discrete empirical interpolation method. As mentioned in [30], a suitable selection of snapshots and time derivatives is crucial in this case. Our method could be extended in that direction, to select not only the "best" snapshots and derivative snapshots, but also an "optimal" set of collocation points. However, this extension is out of the scope of the present work.

In this manuscript, the LUPOD-based method has been applied to different equations, with diverse boundary conditions, spatial patterns, and temporal behaviors, including spatio-temporal chaos and traveling structures. Precisely, numerical experiments on two benchmark pattern-forming systems have been carried out: the Kuramoto-Sivashinsky equation in one spatial dimension and the complex Ginzburg-Landau equation in one and two spatial dimensions. Results point out that the new adaptive reduced order model is computationally much faster than standard numerical integration. The measured online acceleration factor is typically larger in two or more dimensions, since the ratio of physical to numerical degrees of freedom is expected to decrease as the spatial dimension increases, making the novel algorithm to be suitable for multi-dimensional, large scale problems.

It is worth observing that the novel adaptive method is also more efficient than non-adaptive low-dimensional models, if comparing the total offline/online CPU time of simulations, since it can be profitably applied in the offline calculation of the snapshots. On the other hand, an extension to problems describing the evolution of a state variable over some parameter (instead of time), which has a widespread interest, should be straightforward. In addition, our method may be adapted to deal with parameter selection in parameterized model order reduction (see [47] and references therein).

In general, the LUPOD-driven method may be convenient and effective when the number of needed modes is not excessive, for instance in many unsteady, laminar and incipiently transitional flows important in industrial environments. Undoubtedly, what has been illustrated in this paper is an essential and mandatory step forward

to fully exploit the potential of the LUPOD sampling technique, and its integration with adaptive models based on truncated mode expansions. It sets the ground for challenging more complex problems in the near future.

Acknowledgements The authors would like to thank two anonymous referees for some useful comments that helped improve the paper.

Funding This work has been supported by the FEDER / Ministerio de Ciencia, Innovación y Universidades – Agencia Estatal de Investigación, under grants TRA2016-75075-R and MTM2017-84446-C2-2-R.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Adrover, A., Giona, M.: Modal reduction of PDE models by means of Snapshot Archetypes. Physica D **182**, 23–45 (2003)
2. Akkari, N., Hamdouni, A., Liberge, E., Jazar, M.: A mathematical and numerical study of the sensitivity of a reduced order model by POD (ROM-POD), for a 2D incompressible fluid flow. J. Comput. Appl. Math. **270**, 522–530 (2014)
3. Alonso, D., Vega, J.M., Velazquez, A.: Reduced order model for viscous aerodynamic flow past an airfoil. AIAA J. **48**, 1946–1958 (2010)
4. Alonso, D., Vega, J.M., Velazquez, A., de Pablo, V.: Reduced-order modeling of three-dimensional external aerodynamic flows. J. Aerospace Eng. **25**, 588–599 (2012)
5. Amsallem, D., Zahar, M.J., Washabaugh, K.: Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction. Adv. Comput. Math. **41**, 1187–1230 (2015)
6. Aranson, I.S., Kramer, L.: The world of the complex Ginzburg-Landau equation. Rev. Mod. Phys. **74**, 100–142 (2002)
7. Astrid, P., Weiland, S., Willcox, K., Backx, T.: Missing point estimation methods in models described by proper orthogonal decomposition. IEEE T. Automat. Contr. **53**, 2237–2250 (2008)
8. Atkinson, K.E. An Introduction to Numerical Analysis, 2nd edn. Wiley, Inc., New York (1989)
9. Bache, E., Vega, J.M., Velazquez, A.: Model reduction in the back step fluid-thermal problem with variable geometry. Int. J. Therm. Sci. **49**, 2376–2384 (2010)
10. Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. C. R. Acad. Sci. Paris, Ser. **I**(339), 667–672 (2004)
11. Bekemeyer, P., Ripepi, M., Heinrich, R., Görtz, S.: Nonlinear unsteady reduced-order modeling for gust-load predictions. AIAA J. **57**, 1–12 (2019)
12. Bergmann, M., Bruneau, C.H., Iollo, A.: Enablers for robust POD models. J. Comput. Phys. **228**, 516–538 (2009)
13. Berkooz, G., Holmes, P., Lumley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech. **25**, 539–575 (1993)
14. Braconnier, T., Ferrier, M., Jouhaud, J.C., Montagnac, M., Sagaut, P.: Towards an adaptive POD/SVD surrogate model for aeronautic design. Comput. Fluids **40**, 195–209 (2011)
15. Cebeci, T.: Convective Heat Transfer. Springer, Berlin (2002)
16. Chatterjee, A.: An introduction to the proper orthogonal decomposition. Curr. Sci. India **78**, 808–817 (2000)
17. Chaturantbut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. SIAM J. Sci. Comput. **32**, 2737–2764 (2010)
18. Couplet, M., Basdevant, C., Sagaut, P.: Calibrated reduced-order POD-Galerkin system for fluid flow modelling. J. Comput. Phys. **207**, 192–220 (2005)

19. Cross, M., Hohenberg, P.: Pattern formation outside of equilibrium. Rev. Mod. Phys. **65**, 851–1112 (1993)
20. Dowell, E.H., Hall, K.C.: Modeling of fluid-structure interaction. Annu. Rev. Fluid Mech. **33**, 445–490 (2001)
21. Drmac, Z., Gugercin, S.: A new selection operator for the discrete empirical interpolation method – improved a priori error bound and extensions, SIAM. J. Sci. Comput. **38**, A631–A648 (2016)
22. Dimitriu, G., Stefanescu, R., Navon, I.M.: Comparative numerical analysis using reduced-order modeling strategies for nonlinear large-scale systems. J. Comput. Appl. Math **310**, 32–43 (2017)
23. Drohmann, M., Haasdonk, B., Ohlberger, M.: Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. SIAM J. Sci. Comput. **34**, A937–A969 (2012)
24. Golub, G.H., van Loan, G.T.: Matrix Computations. John Hopkins Univ. Press, Baltimore (1996)
25. Gottlieb, D., Orszag, S.A.: Numerical Analysis of Spectral Methods: Theory and Applications. SIAM, Philadelphia (1977)
26. Gräßle, C., Hinze, M., Lang, J., Ullmann, S.: POD model order reduction with space-adapted snapshots for incompressible flows. Adv. Comput. Math. **45**, 2401–2428 (2019)
27. Haragus, M., Iooss, G.: Local Bifurcations, Center Manifolds, and Normal Forms in Infinite Dimensional Dynamical Systems. Springer, Berlin (2010)
28. Hyman, J.M., Nicolaenko, B.: The Kuramoto-Sivashinsky equation: a bridge between PDE's and dynamical systems. Physica D **18**, 113–126 (1986)
29. Kengne, E., Lakhssassi, A., Vaillancourt, R., Liu, W.M.: Exact solutions for generalized variable-coefficients Ginzburg-Landau equation: application to Bose-Einstein condensates with multi-body interatomic interactions, vol. 53 (2012)
30. Kostova-Vassilevska, T., Oxberry, G.M.: Model reduction of dynamical systems by proper orthogonal decomposition: error bounds and comparison of methods using snapshots from the solution and the time derivatives. J. Comput. Appl. Math. **330**, 553–73 (2018)
31. Kuramoto, Y., Tsuzuki, T.: Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. Progr. Theoret. Phys. **55**, 356–369 (1976)
32. Lam, C.-K., Malomed, B.A., Chow, K.W., Wai, P.K.A.: Spatial solitons supported by localized gain in nonlinear optical waveguides. Eur. Phys. J. Special Topics **173**, 233–243 (2009)
33. Lieu, T., Farhat, C., Lesoinne, M.: Reduced-order fluid/structure modeling of a complete aircraft configuration. Comput. Method. Appl. M. **195**, 5730–5742 (2006)
34. Malomed, B.A.: Spontaneous Symmetry Breaking, Self-Trapping, and Josephson Oscillations, Progress in Optical Science and Photonics. Springer, Berlin (2013)
35. Pearson, K.: On lines and planes of closest fit to systems of points in space. Philos. Mag. **2**, 559–572 (1901)
36. Peherstorfer, B., Butnark, D., Willcox, K., Bungartz, H.J.: Localized discrete empirical interpolation method. SIAM J. Sci. Comput. **36**, A168–A192 (2014)
37. Peherstorfer, B., Willcox, K.: Online adaptive model reduction for nonlinear systems via low-rank updates. SIAM J. Sci. Comput. **37**, A2123–A2150 (2015)
38. Peterson, J.S.: The reduced basis method for incompressible viscous flow calculations. SIAM J. Sci. Comput. **10**, 777–786 (1989)
39. Rapún, M.-L., Terragni, F., Vega, J.M.: Adaptive POD-based low-dimensional modeling supported by residual estimates. Int. J. Numer. Meth. Eng. **9**, 844–868 (2015)
40. Rapún, M.-L., Terragni, F., Vega, J.M.: LUPOD: Collocation in POD via LU decomposition. J. Comput. Phys. **335**, 1–20 (2017)
41. Rapún, M.-L., Vega, J.M.: Reduced order models based on local POD plus Galerkin projection. J. Comput. Phys. **229**, 3046–3063 (2010)
42. Rempfer, D.: On low-dimensional Galerkin models for fluid flow. Theor. Comp. Fluid Dyn. **14**, 75–88 (2000)
43. Ryckelynck, D.: Hyper-reduction of mechanical models involving internal variables. Int. J. Numer. Meth. Eng. **77**, 75–89 (2009)
44. Sirisup, S., Karniadakis, G.E., Xiu, D., Kevrekidis, I.G.: Equations-free/Galerkin-free POD assisted computation of incompressible flows. J. Comput. Phys. **207**, 568–587 (2005)
45. Sirovich, L.: Turbulence and the dynamics of coherent structures. Q. Appl. Math. **XLV**, 561–590 (1987)
46. Sivashinsky, G.: Nonlinear analysis of hydrodynamic instability in laminar flames I. Derivation of basic equations. Acta Astron. **4**, 1177–1206 (1977)

47. Soll, T., Pulch, R.: Sample selection based on sensitivity analysis in parameterized model order reduction. J. Comput. Appl. Math. **316**, 369–379 (2017)
48. Stabile, G., Ballarin, F., Zuccarino, G., Rozza, G.: A reduced order variational multiscale approach for turbulent flows. Adv. Comput. Math. **45**, 2349–2368 (2019)
49. Taira, K., Brunton, S.L., Dawson, S.T.M., Rowley, C.W., Colonius, T., McKeon, B.J., Schmidt, O.T., Gordeyev, S., Theofilis, V., Ukeiley, L.S.: Modal analysis of fluid flows: an overview. AIAA J. **55**, 4013–4041 (2017)
50. Terragni, F., Valero, E., Vega, J.M.: Local POD plus Galerkin projection in the unsteady lid-driven cavity problem. SIAM J. Sci. Comput. **33**, 3538–3561 (2011)
51. Terragni, F., Vega, J.M.: On the use of POD-based ROMs to analyze bifurcations in some dissipative systems. Physica D **241**, 1393–1405 (2012)
52. Terragni, F., Vega, J.M.: Construction of bifurcation diagrams using POD on the fly. SIAM J. Appl. Dyn. Syst. **13**, 339–365 (2014)
53. Tobias, S.M., Proctor, M.R.E., Knobloch, E.: Convective and absolute instabilities of fluid flows in finite geometry. Physica D **113**, 43–72 (1998)
54. Zimmermann, R.: Towards best-practice guidelines for POD-based reduced order modeling of transonic flows EUROGEN. In: 2011 Proceedings, CIRA (2011)