Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2017.m1624

Vol. **10**, No. 4, pp. 872-894
November 2017

Adaptive Dimensionality-Reduction for Time-Stepping in Differential and Partial Differential Equations

Xing Fu and J. Nathan Kutz*

Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA

Received 6 July 2016; Accepted (in revised version) 13 April 2017

Abstract. A numerical time-stepping algorithm for differential or partial differential equations is proposed that adaptively modifies the dimensionality of the underlying modal basis expansion. Specifically, the method takes advantage of any underlying low-dimensional manifolds or subspaces in the system by using dimensionality-reduction techniques, such as the proper orthogonal decomposition, in order to adaptively represent the solution in the optimal basis modes. The method can provide significant computational savings for systems where low-dimensional manifolds are present since the reduction can lower the dimensionality of the underlying high-dimensional system by orders of magnitude. A comparison of the computational efficiency and error for this method are given showing the algorithm to be potentially of great value for high-dimensional dynamical systems simulations, especially where slow-manifold dynamics are known to arise. The method is envisioned to automatically take advantage of any potential computational saving associated with dimensionality-reduction, much as adaptive time-steppers automatically take advantage of large step sizes whenever possible.

AMS subject classifications: 74S25, 65M70, 37L65

Key words: Time-stepping, dimensionality reduction, proper orthogonal decomposition.

1. Introduction

Computation is ubiquitous across the physical, biological and engineering sciences, revolutionizing many fields of study by allowing one to solve complex problems through algorithmic/numerical methods. And with the continued and significant increase in computing power for a fixed cost, even problems thought once to be intractable are now routinely solved with high-performance computing architectures or even desktop/laptop computing. The role of improved processor performance is unquestioned in helping to revolutionize the impact of computational science. However, equally as important are the algorithmic developments over the past few decades that have exploited any mathematical structure

*Corresponding author. *Email address:* kutz@uw.edu (J. N. Kutz)

 ©2017 Global-Science Press

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

in the equations of interest to its full advantage. As an example, one needs only to consider adaptive time-stepping algorithms that are applied to the solutions of differential and partial differential equations. Such algorithms aim to take as large a time-step as possible while being constrained to some absolute or relative error measure. So prevalent are these adaptive time-steppers that they are the standard, and expected, routines in most high-level language packages such as MATLAB, Octave or Scipy. In a similar fashion, we propose a simple and efficient *adaptive* time-stepping scheme where the *dimensionality* of the underlying system is adapted in order to more efficiently solve a given differential or partial differential equation system. Although dimensionality reduction techniques for ODEs and PDEs are well known (See [1, 2] and references therein), the method advocated here provides a new, natural framework for a fully adaptive, robust and general time-stepping scheme which exploits optimal proper orthogonal decomposition (POD) basis modes whenever possible.

Dimensionality-reduction techniques, although around for more than a century [3–6], have recently grown in importance for solving a wide range of physical problems. The rise of such techniques often is related to the numerical discretization of PDE systems, for instance, that can often yield a system of equations with millions or billions of degrees of freedom (or more). Thus dimensionality reduction methods can form the underpinnings of developing reduced-order models through projections such as the proper orthogonal decomposition [1–9]. This manuscript aims to bring together well-known dimensionality reduction and adaptive time-stepping schemes with the goal of producing an intuitively appealing and natural, perhaps even obvious, time-stepping algorithm for differential and partial differential equations that automatically takes advantage of any low-dimensional, slow-manifold dynamics that exists in the system. Thus an adaptive time-stepping algorithm is demonstrated where dimensionality reduction is exploited whenever possible in an automated way. The basis of such a strategy is rooted in the vast literature on coherent structures (low-dimensionality) that are observed to be prevalent in, for instance, PDE systems across a broad range of the physical, biological and engineering sciences [16]. Indeed, some underlying low-dimensional and slow-manifold seems to be often exhibited in the dynamics of high-dimensional dynamical systems.

Many other techniques have also been recently advocated with the aim of exploiting some form of dimensionality reduction. In Schaeffer et al. [17] and Jokar et al. [18], *sparsity* is exploited for minimizing the number of basis functions for evaluations of PDE.

By reducing or compressing the information needed to represent the solution at every step, only the essential dynamics are represented. Thus the basis modes are not modified, they are simply turned on and off as necessary at each time-step. Alternatively, one can capitalize on the newly developed dynamic mode decomposition (DMD) technique which projects onto a lower modal basis and evolves the future state according to a linear set of differential equations [19]. Although dimensionality reduction is achieved, the time-stepping equations are constrained to be linear. In yet another technique, an empirical interpolation method (DEIM) [20, 21] provides a modification of the POD method that reduces the complexity of evaluating the nonlinear terms of the reduced model to a cost proportional to the number of reduced variables obtained by POD. However, no

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

adaptive stepping is considered. A version of *spatial* or *local* adaptivity for reduced order modeling has also recently been considered [22], but it enforces spatial dimensionality reduction rather than any adaptive time-stepping. And in a similar fashion, gappy POD has been considered as a means of using a small amount of data for efficient full state reconstruction [23,24]. Further algorithmic innovations have also been used to adaptively refine reduced-order models [25], potentially generate a set of internal variables to perform hyper-reduction of the complex system [26] and/or take larger time steps using online model reduction [27].

More closely related to the present work is a recent adaptive selection method for snapshots in time that limits offine training costs by selecting snapshots according an error control mechanism similar to that found in adaptive time-stepping ordinary differential equation solvers [28]. Such adaptation strategies can also be used to efficiently approximate the dominant POD modes by recomputing updates with the addition of each snapshot either potentially increasing or decreasing the dimensionality [29]. All the methods above have a clear goal: for a given accuracy, improve computational efficiency and significantly reduced computational time by exploiting the changing low-rank structure of the data or simulation. The present technique advocated transforms to the guaranteed optimal basis (in an L^2 sense) so that a thresholding does not have to be applied at each time-step (unlike Ref. [17]). Further, nonlinear evolution is allowed and adaptivity is central to the scheme unlike Refs. [19] and [21] respectively. Thus the method is demonstrated to be a highly successful, stand-alone algorithm which potentially could integrate some of the other techniques [17,19,21] to even greater effect.

The paper is outlined as follows. In the next section, the proper orthogonal decomposition with Galerkin projection is outlined as it applies to high-dimensional dynamical systems (in this case PDEs). Section 3 outlines the algorithm that takes advantage of the low-dimensional projections found in Section 2. In particular, the methods used to go from high-to-low and low-to-high dimensions are advocated. Section 4 demonstrates the successful application of the adaptive dimensionality time-stepper with three example PDEs. This is followed in Section 5 with an analysis of the computation time and error associated with implementing the time-stepping scheme. The paper is concluded in Section 6 with an overview and outlook for the proposed time-stepping scheme.

2. Proper orthogonal decomposition and Galerkin projection

High-dimensional dynamical systems, which are often derived from partial differential equations, are typically beyond the reach of analytic methods and techniques for informing one of the underlying system dynamics. Computational methods, which yield high-dimensional systems, are thus employed to explore the behavior of the system in time and space as a function of the system parameters. But as is often the case, many high-dimensional systems elicit low-dimensional dynamical behaviors [16] that can be better understood using dimensionality-reduction methods [3–9].

One of the most important dimension reduction techniques is the proper orthogonal decomposition (POD) [8] that is based upon the singular value decomposition [30]. It

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

reduces the dimension of the system of interest by projecting the full system onto the optimal (in an L^2 sense [30]) set of basis functions determined from empirical and/or computational data, where the computational data are snapshots of solutions of the PDE. The basic idea of the POD method is as follows: given a set of data that lies in space of dimension n, find a subspace of fixed dimension r, where $r \ll n$, such that the error in the projection onto the subspace is minimized.

For our purposes, and to illustrate the concept, consider a given PDE of the form

$$\frac{\partial A}{\partial t} = N\left(x,\ t,\ \frac{\partial A}{\partial x},\ \frac{\partial^2 A}{\partial x^2},\ \cdots\right), \tag{2.1}$$

where $N(\cdot)$ is in general a non-constant coefficient, time-dependent, and nonlinear function of $A(x,t)$ and its derivatives. Associated with Eq. (2.1) are also a set of boundary conditions in x and an initial condition $A(x,0)$. Solutions to Eq. (2.1) can be computed numerically by discretizing $A(x,t)$ into a series of spatial locations at time $t = t_j$ so that $A(x,t_j) \to \mathbf{a}_j$ where $\mathbf{a}_j = [a(x_1,t_j)\ a(x_2,t_j)\ \cdots\ a(x_n,t_j)]$. The discretized solution can then be advanced in time using standard methods including spectral, finite-difference, or finite element techniques, whichever is most appropriate for the given boundary conditions and geometry. In numerically solving Eq. (2.1), solution trajectories of the n-dimensional system in time are achieved. Our objective is to then arrange the numerical solution in a data matrix that includes m snapshots of the system:

$$\mathbf{X} = \left[\mathbf{a}_1,\ \cdots,\ \mathbf{a}_m\right], \tag{2.2}$$

where $\mathbf{a}_j$ with $j = 1,2,\cdots,m$ determines the sampling of the solution at the prescribed times t_j.

Application of the singular value decomposition(SVD) to the data matrix $\mathbf{A}$ is guaranteed to yield the matrix decomposition

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \left[\boldsymbol{\phi}_1,\ \cdots,\ \boldsymbol{\phi}_n\right] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix} \begin{bmatrix} v_{1,1} & \cdots & v_{1,m} \\ & \vdots & \\ v_{m,1} & \cdots & v_{m,m} \end{bmatrix}, \tag{2.3}$$

where the diagonal matrix $\mathbf{\Sigma} \in \mathbb{R}^{n\times m}$ can be used to determine the rank (or approximate rank) of the data matrix and the matrices $\mathbf{U} \in \mathbb{C}^{n\times n}$ and $\mathbf{V} \in \mathbb{C}^{m\times m}$ contain the temporal and spatial projections of the data respectively.

The column vectors of $\mathbf{U}$, denoted by $\{\boldsymbol{\phi}_j\}_{j=1}^n$, are called the POD basis modes. The POD modes $\boldsymbol{\phi}_j$ are n-dimensional vectors. The length of the vectors is determined by the spatial discretization of the original continuous PDE into a discrete set of n coupled differential equations. They are arranged in order of *importance* in an L^2 (energy) sense. Thus the first mode $\boldsymbol{\phi}_1$ accounts for the most "energy" in the data set $\mathbf{X}$, the second mode $\boldsymbol{\phi}_2$ contains the next most "energy" content and so forth. Moreover, all the modes are orthogonal so that their inner product satisfies the relation

$$\langle \boldsymbol{\phi}_i,\ \boldsymbol{\phi}_j \rangle = \boldsymbol{\phi}_i^* \boldsymbol{\phi}_j = \delta_{ij}, \tag{2.4}$$

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

where δ_{ij} is the Dirac delta function and the integration is performed over the entire x range of the data set $\mathbf{X}$. Note that under discretization, the integration becomes simply a dot product, potentially weighted by a quadrature rule. In what follows, we use a simple trapezoidal rule.

The critical observation of dimensionality reduction involves the so-called singular (diagonal) matrix $\mathbf{\Sigma}$. It is this matrix that will determine the nature of the low-rank subspace used in representing the dynamics contained in $\mathbf{X}$. The energy of each mode is defined as

$$E_j = \frac{\sigma_j^2}{\sum_{i=1}^{n} \sigma_i^2} \in [0,1]\,. \tag{2.5}$$

By knowing the energy in each mode, and knowing they are ordered in importance from $j = 1$ (most dominant) to $j = n$ (least dominant), we can arbitrarily truncate the full n dimensional space to a low-rank subspace of dimension r where $r \ll n$. A guarantee of the POD basis is that they represent the optimal truncation in an L^2 sense. For instance, one may decide to work with a low-rank modal approximation that captures 99.9% of the energy, i.e. $\sum_{j=1}^{r} E_j < 0.999$. Thus the rank r is determined by the arbitrarily imposed energy constraint.

So far, we have computed the POD basis of the solution given from the data matrix $\mathbf{X}$. We can then construct reduced order models using Galerkin projection on to the low-dimensional subspace using the truncated POD basis. We apply a Galerkin projection by simply projecting the original vector field $N(\cdot)$ in Eq. (2.1) onto the desired r-dimensional subspace spanned by $\{\boldsymbol{\phi}_j\}_{j=1}^{r}$:

$$\mathbf{A}_r(x,\,t) = \sum_{j=1}^{r} b_j(t)\boldsymbol{\phi}_j\,. \tag{2.6}$$

Substituting this approximation in to the governing equation (2.1), multiplying by $\boldsymbol{\phi}_k^*$ and applying orthonormality generates the low-rank dynamic approximation

$$\frac{db_k}{dt} = \left\langle \boldsymbol{\phi}_k,\, N\left(x,\,t,\,\frac{\partial \mathbf{A}_r}{\partial x},\,\frac{\partial^2 \mathbf{A}_r}{\partial x^2},\,\cdots\right)\right\rangle \tag{2.7}$$

for $k = 1,\,\cdots,\,r$ and where the $b_k(t)$ characterize the evolution of the system in the POD bases $\boldsymbol{\phi}_k$. This resulting set of ODEs describes the evolution of the low-rank subspace spanned by $\mathbf{A}_r(x,t)$. Provided $r \ll n$, the reduced system is of a much lower dimension while still capturing the dynamic features of the full PDE system. Such POD-Galerkin approximations have a variety of potential drawbacks, including the fact that the reduced order models often exhibit numerical instabilities and fail to agree quantitatively with the full PDE [2,31]. However, for parametrized systems, a suite of local ROMs can be constructed to avoid the well-known numerical instabilities generated from a global model encompassing many parameter regimes [2]. Moreover, Petrov-Galerkin time-stepping can be used instead of POD-Galerkin which can greatly improve the stability of time-stepping schemes. Regardless, simulating the future state of the system is much more efficient in this basis

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

decomposition since there are r ODEs instead of n ODEs to simulate. The error of the time-stepping will undoubtedly by determined by the truncation applied in determining the rank r of the subspace used. Recent results on using the DEIMs method [21] can potentially reduce the computation of the nonlinear terms and provide further computational savings. It should also be noted that constraining the solution to the above low-rank approximation proves difficult to rigorously justify, with only a few analytic results achieved to date [32]. Indeed, rigorous error estimates using dimensionality reduction techniques are an area of ongoing and active research given its wide-spread application in almost all branches of science.

This methodology, which we emphasize is fairly standard and well-known, has been applied successfully for representing the time-evolution of high-dimensional dynamical systems [7–9]. However, there is a clear disadvantage to this technique. Namely, once the low-rank basis modes $\{\boldsymbol{\phi}_j\}_{j=1}^{r}$ have been selected, they are fixed and are incapable of representing “new" dynamics in the system, i.e. the modal basis set can only represent dynamics that are captured in the data matrix $\mathbf{X}$. So if there is a bifurcation parameter in the system, or perhaps simply coefficients/parameters that evolve on slow time scales, the POD basis selected will generically be unable to accurately predict the future state of the system. In our method, this problem is circumvented by evaluating wether or not a new low-rank approximation is needed to represent the dynamics. We adapt the basis in time as required by the data matrix and subsequent sampling of the dynamics. The novelty of this manuscript is in developing a technique and algorithm for adapting the modal basis in order to provide highly-efficient computations of high-dimensional dynamical systems.

3. Adaptive rank-selection algorithm

The adaptive dimensionality stepping algorithm begins by considering the given partial differential equation (2.1) that is discretized on a specified domain into an n-dimensional system. The first step of the adaptive algorithm is to time-step the full, n-dimensional PDE system for a prescribed time to obtain the solution data matrix $\mathbf{X}$ which samples the solution over a prescribed time Δt. Due to the fact that the POD approach doesn't rely on any prior knowledge of the process that generates the data, any appropriate numerical method can be applied to the full PDE simulation, for instance, a spectral method or finite difference scheme. As soon as enough simulation data has been obtained to construct the data matrix $\mathbf{X}$, a check is made to see whether or not it is possible to switch to a low dimensional POD basis and solver. Typically this happens if the simulation has evolved onto the low-dimensional (slow) manifold of the governing equations. If no reduction is possible, then the full n-dimensional time-stepping algorithm continues to advance the solution in time. It should be noted that switching between high-dimensional and low-dimensional models is a current and active research area in the reduced order modeling community, with some recent methods that are capable of efficiently using low-rank updates to aid in the process [33].

We will introduce *three* approaches for testing for a potential reduction from the full n-dimensional discretization to an r-dimensional POD approximation where $r \ll n$. If the

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

decision made by the algorithm is to switch to a low dimensional POD evolution, it will compute the POD basis using a standard SVD decomposition. The initial condition used for the POD expansion is computed using the projection of the last snapshot of the data matrix $\mathbf{X}$ onto the POD basis. The time-stepping algorithm will then advance the solution in the low-rank subspace of dimension r. Typically, this is the approach advocated when implementing the POD bases [7–9]. However, for the time-stepping advocated here, it is critical to check when the POD basis fails to properly encode, or represent, the dynamics of the full system n-dimensional system. Thus it is also important to determine when to go from the low-rank system back to the full-rank system in the simulation. In what follows, *two* approaches are demonstrated which systematically check for the required *dimensionality-inflation* procedure. Thus if a dimensionality-inflation is required, then the time-stepping algorithm in the r-dimensional POD basis is recast back into the full n-dimensional computational framework. Fig. 1 shows a flowchart of the adaptive POD algorithm described above. By judiciously using dimensionality-reduction (rank r) and -inflation (rank n), an efficient time-stepping algorithm is demonstrated that optimally takes advantage of the low-dimensional behavior in the system whenever it is possible to do so.

3.1. Dimensionality reduction: from rank n (high) to rank r (low)

The following subsections propose three different dimensionality reduction checks that are capable of providing robust low-rank approximations to the governing equations. They are presented in order of increasing computational complexity: from simple $\mathcal{O}(n)$ checks to full $\mathcal{O}(n^3)$, SVD based methods. Of course, one would use the simple scheme when possible and resort to the SVD-based check only when necessary. The checking represents an overall computational *overhead* for evaluating the future state of the system. Note that the test need not be applied at every numerical time step δt, but rather at some fixed interval of time Δt. For instance, a time-stepper may require $\delta t \sim \mathcal{O}(10^{-4})$ whereas a dimensionality reduction test may be applied only every $\Delta t \sim \mathcal{O}(1)$. Thus the overhead for evaluation is small relative to the computational time for simulation. Each test has a different computational cost and can potentially give more or less information about the rank of the evolution. For instance, in the SVD test proposed, it is the most computationally expensive, but gives the most accurate quantification of the rank. The first test only identifies steady-state solutions so is limited to a specific solution form. In practice, all three tests can be applied in the order suggested, with the first test being the most rapid to evaluate and the third test the most expensive. In each case, the test can potentially be exploited for the adaptive dimensionality selection.

3.1.1. Test 1: Steady-state L^2-norm test

The first test for dimensionality reduction is a low-cost method based upon an L^2 norm test of the solution. This approach works especially well for detecting steady state solutions. Assume $A(x,t)$ is the solution of the full PDE (2.1) in the subinterval $t \in [t_k,\ t_k + \Delta t]$

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

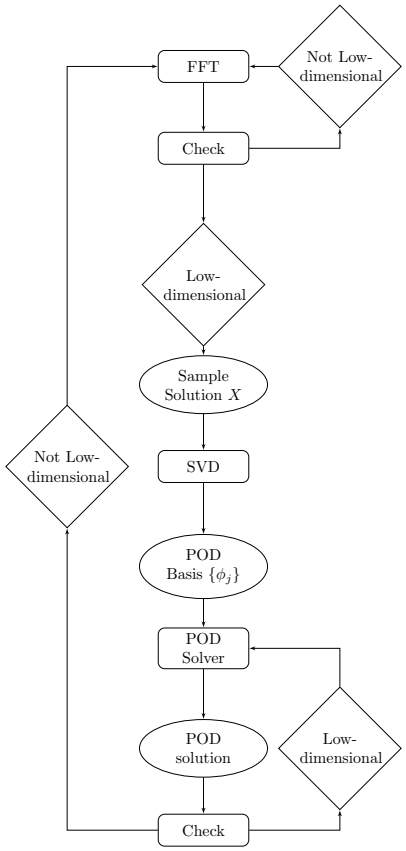


Figure 1: Example flowchart for implementing the adaptive POD algorithm. In this case, a spectral method using the fast Fourier transform is implemented so that the FFT represents the full simulation of rank n. The POD solver is implemented using an r-rank approximation where $r \ll n$.

for some prescribed Δt which is not the time-integration time-step (generally $\Delta t \gg \delta t$ where δt is the numerical time-step). If it is a steady state solution, then a simple check for low-rank behavior can be obtained by considering the quantity:

$$e_n = \frac{\left\| A(x, t_k + \frac{\Delta t}{2}) - A(x, t_k + \Delta t) \right\|}{\|A(x, t_k + \Delta t)\|} < \epsilon \tag{3.1}$$

for the given Δt. For very small ϵ, in the range of numerical roundoff, this property suggests a switch to the r-dimensional low-rank POD solver without any trouble since a low-rank subspace (steady-state) solution has been reached. Of course, as will be shown, the steady state solution can abruptly be made unstable if parameters in the governing PDE are modified. This can lead to bifurcations or modification of the solution so that

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

the POD modes selected are no longer good for approximating the solution. For such parametrized systems, the steady-state tests gives a method for exploiting the temporary low-rank dynamics using a simple $r = 1$ approximation.

3.1.2. Test 2: Center-of-mass test

Although the L^2-Norm Test works perfectly well for stationary solutions, it fails to identify low-rank, time-dependent solutions such as, for instance, breathers. Breather solutions, which are common in nonlinear wave propagation problems, are typically characterized by a localized solution which undergoes temporal oscillations [42]. Thus the breather solutions, in dissipative systems for instance, have a limit cycle attractor. The limit cycle itself is low-dimensional and a rank-truncated POD basis can be used to characterize it. Since breather solutions are low dimensional, a low dimensional POD solver can be applied. However, many time-dependent, perhaps chaotic, solutions are not low-rank and the low-dimensional POD solver is a poor choice for representing the solution. Thus it is important to differentiate between low-rank breather-type solutions and more exotic, and not low-rank, time-dependent solutions in the adaptive time-stepping method. To do this, we introduce $\bar{x}$ which is the center-of-mass defined as

$$\bar{x}(t) = \frac{\int xA(x,t)dx}{\int A(x,t)dx}. \tag{3.2}$$

For symmetric breather solutions, the value of $\bar{x}(t)$ is constant. Of course, just like Test 1, the center-of-mass test fails to capture low-rank solutions with symmetry or invariance properties such as rotation or translation. However, the test can be modified and/or redesigned for this as well [9,41].

For multi-pulse stationary solutions considered here, for instance, the center of mass $\bar{x}(t)$ is also a constant. So in each subinterval, we can check whether the dynamics is low dimensional by evaluating the center-of-mass function $\bar{x}(t)$. Assume we are working on subinterval $t \in [t_k,\ t_k + \Delta t]$ for a prescribed Δt as before. The time shift indicator $\bar{x}_{shift}$ is defined in the following form:

$$\bar{x}_{shift} = \frac{1}{\Delta t}\int_{t_k}^{t_k+\Delta t} \left|\bar{x}(t) - \bar{x}(t_k + \Delta t)\right| dt\,. \tag{3.3}$$

If $\bar{x}_{shift}$ is greater than a given threshold value ϵ, we can be confident that in the current subinterval, it is not a simple steady-state of time-periodic solution. However, if $\bar{x}_{shift}$ is less than the prescribed threshold value of ϵ, a low rank POD solver can be applied to obtain a more efficient computation of the solution in time. Like the first test, this is a more computationally efficient test than a full SVD reduction.

3.1.3. Test 3: Number of SVD modes test

An SVD decomposition is the most robust, and costliest test, available for evaluating the rank of the data matrix. Thus in addition to the two previous tests, both of which are

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

numerically efficient to evaluate, we can compute the SVD of a given number of snapshots of the data in the time interval $t \in [t_k,\ t_k + \Delta t]$.

The SVD can be expensive to evaluate since discretization of the PDE yields an n-dimensional system of differential equations. Of course, the evaluation of the SVD is only performed occasionally in comparison to the numerical time-step δt. Moreover, an economy SVD can be performed which limits the computational cost of the algorithm by the number of snapshots used. However, if one would wish to check continuously, then incremental updating of the SVD could potentially be used [28, 34, 35]. Alternatively, for very high-dimensional systems, one could also compute the SVD efficiently using randomized techniques [36]. For this method, we evaluate the number of modes required to represent a certain amount of energy in the data matrix $\mathbf{X}$, say 99.999% of total energy. For steady state and breather solutions, less then five modes are often sufficient to achieve this threshold. For more exotic solutions, more modes may be required, but as long as they do not increase the rank of the dynamics substantially, then computational savings can be established. Utilizing the above strategy, the number-of-modes test can be used to classify the current low-rank nature of the dynamics of the equation, ultimately leading to a switching from full PDE simulations to a low dimensional POD solver.

In practice then, a simulation of the full PDE is performed over a given subinterval $t \in [t_k,\ t_k + \Delta t]$ to generate the data matrix $\mathbf{X}$. Applying the SVD to this matrix yields the singular matrix which allows for an evaluation of the low-rank nature of the data. If the SVD shows the solution to be of sufficiently low-rank, as determined by the user, the POD modal basis $\{\phi_j\}_{j=1}^{r}$ is chosen and the Galerkin projection (2.6) is used to time-step forward into the future. This basis is used until it no longer is able to capture the correct dynamics of the full PDE system.

3.2. Dimensionality inflation: from rank r (low) back to rank n (high)

This subsection is concerned with determining when the POD approximation breaks down. In particular, it is fairly easy to understand when the solution has become low-rank, and the previous section describes three techniques for doing so. Once a POD-Galerkin projection has been applied, it is imperative to determine when one has to once again return to the full, high-dimensional PDE computation. Two robust techniques are highlighted here. Note that unlike some of the dimensionality reduction methods used for reducing rank, the tests used for determining when to return to the full state simulation are computationally inexpensive in comparison to the costs of POD-Galerkin simulations. Moreover, the tests need only be applied occasionally at every Δt which is much larger than the time-integration time-step δt.

3.2.1. Test 1: Auxiliary modes test

Assuming the first r modes $\{\phi_j\}_{j=1}^{r}$ from the full, high-dimensional PDE simulations contain a prescribed amount of energy required to approximate the system with some precision, we can add m extra modes to the truncated POD basis and project the solution onto

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

the combined $r+m$ modes. This is called the extended basis. Note that similar methods of extending the basis in ROMs has been applied for LES modeling of turbulent flows [37,38].

From our numerical experiments, we know that the m auxiliary modes should only contain negligible amounts of energy beyond the truncation at r modes. Thus we can use the energy content in these m auxiliary modes as a probe which can tell us if the low rank POD solver works well for the current subinterval, or if it needs to be replaced by full PDE simulation. More precisely, since the energy on the extra m modes are quite low (usually less than 10^{-10} in our applications), adding them to the rank r truncated POD basis $\{\phi_j\}_{j=1}^{r}$ should not affect the accuracy of the solver. However, when the extended POD basis fails to capture the dynamics of the system, the energy of these m auxiliary modes will grow significantly. Thus a reliable test for checking for the necessity of switching to full PDE simulation is provided by monitoring the energy on these m modes. Since it is not possible to track the energy as defined in Eq. (2.5), we use instead the proxy energy metric:

$$\mathscr{E}_j = \frac{b_j^2(t)}{\sum_{i=1}^{r+m} b_i^2(t)} \in [0,1], \tag{3.4}$$

where the b_j are the POD coefficients obtained by Galerkin projection defined in Eq. (2.6). These coefficients are easily tracked from the reduced order model constructed by POD-Galerkin projection. Essentially, this definition of energy is equivalent to Eq. (2.5). However, it can be easily computed in our low dimensional POD solver. With this solver, the energy in the m auxiliary modes can be easily evaluated. It is observed that when the energy in the auxiliary modes grow, then the rank r POD projection no longer accurately captures the full PDE dynamics and dimensionality inflation must be performed.

3.2.2. Test 2: Residual test

In addition to the auxiliary mode test, which is fairly simple and computationally efficient to perform, there is a second obvious test for evaluating the appropriateness of the low-rank POD approximation. This test, called the residual test, simple checks the to see if the POD expansion satisfies the governing equations to some prescribed accuracy. Specifically, the PDE satisfies (2.1). The residual of the PDE is defined as

$$\mathbf{R} = \frac{\partial \mathbf{A}_r}{\partial t} - N\left(x,\, t,\, \frac{\partial \mathbf{A}_r}{\partial x},\, \frac{\partial^2 \mathbf{A}_r}{\partial x^2},\, \cdots\right), \tag{3.5}$$

where $\mathbf{A}_r$ is the low-rank POD solution. Although the residual is not equivalent to the error, it is a proxy for it. Thus intuitively, if the residual is greater than a specified tolerance, then the numerical algorithm is considered to have failed in representing the PDE solution. In contrast, if the residual remains bounded below some prescribed tolerance level, then the low-rank POD projection is considered to be a successful representation of the full solution. This second dimensionality inflation test is more computationally costly than the auxiliary mode test, but it does provide a more concise measure of the error in approximating the solution.

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

4. Applications

To test the dimensionality reduction and inflation techniques advocated in the previous section in an adaptive dimensionality numerical scheme, three example PDE systems will be considered. Not only will a direct comparison be made between the full PDE and low-rank POD projection, but an evaluation of the computational efficiency will be provided. The methods advocated for going from high-to-low (reduction) and low-to-high (inflation) dimensions are shown to be quite effective, thus supporting the idea of adaptive time-stepping schemes based upon dimensionality.

4.1. Model 1: Cubic-quintic Ginzburg-Landau equation

One of the prototype models of mathematical physics is the cubic-quintic complex Ginzburg-Landau equation (CQGLE). We will specifically consider this model in the context of mode-locked laser theory so that [42]

$$\begin{aligned} &i\frac{\partial u}{\partial z}+\frac{D}{2}\frac{\partial^2 u}{\partial t^2}+\gamma|u|^2u-\nu|u|^4u \\ =&ig(z)\left(1+\tau\frac{\partial^2}{\partial t^2}\right)u-i\delta u+i\beta|u|^2u-i\mu|u|^4u, \end{aligned} \tag{4.1}$$

where in the case of mode-locking, gain saturation is provided with the auxiliary equation

$$g(z)=\frac{2g_0}{1+\frac{1}{e_0}\int_{-\infty}^{\infty}|u|^2dt}. \tag{4.2}$$

Here $u(z,t)$ represents the complex envelope of the electric field in the frame of reference of a pulse propagating in a laser cavity. Thus the z variable represents the time-like variable, which is the propagation distance, while t is the space-like variable representing time ahead and behind the pulse. The dispersion coefficient D characterizes the chromatic dispersion in the laser cavity which is positive for anomalous dispersion and negative for normal dispersion. The cubic and quintic terms, both real and imaginary parts, represent the nonlinear cavity effects arising from the saturating gain-loss in the cavity [43]. The parameters g_0 and e_0 are the saturating gain and energy of the cavity while τ measures the gain bandwidth. Linear cavity losses due to output coupling and attenuation are modeled by the parameter δ. Periodic boundary conditions are assumed in the computational scheme with a given non-zero (white-noise) initial condition. Such initial conditions are standard for modeling mode-locked lasers since laser cavities typically start with a *cold start* whereby cavity noise seeds the initial pulse formation. However, since the mode-locked solution is an attractor, a variety of other initial conditions can be used for reproducibility. Most common among these is a Gaussian pulse centered in the computational domain.

In our numerical experiment, the parameters are taken to be the following: $D=0.4$, $\gamma=1$, $\nu=0.01$, $\tau=0.1$, $\delta=1$, $\beta=0.3$, $\mu=0.02$ and $e_0=1$. Our bifurcation parameter is the gain parameter g_0. Specifically, when the value of gain g_0 is increased, the

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

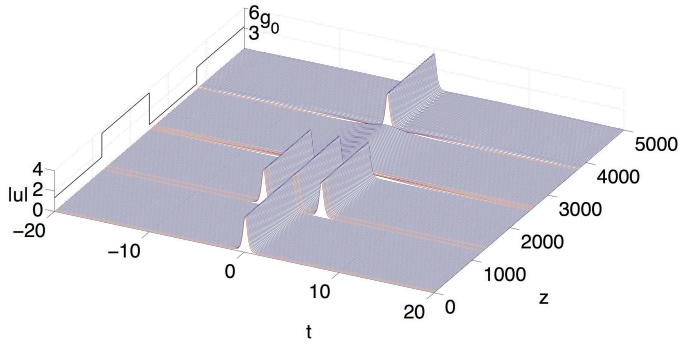


Figure 2: Waterfall plot of solution to the CQGLE obtained by the adaptive POD algorithm. The simulation matches the full PDE simulation as demonstrated in the error analysis of Sec. 5. The red region corresponds to the region where the full PDE solver is evoked by the algorithm while the blue region corresponds to where the low dimensional POD solver is chosen. The time-dependent bifurcation parameter g_0 is illustrated at left.

stable pulse solution undergoes a multi-pulsing transition which is commonly observed in mode-locking system [42]. This property allows us to test whether the adaptive POD algorithm can capture the dynamics of PDEs when the parameter is changed and corresponding change of dynamics occurs.

In our prototypical numerical experiment, we let g_0 be a step function which is defined as follows

$$g_0(z) = \begin{cases} 2.0 & : z \in [0, 1250), \\ 5.5 & : z \in [1250, 2500), \\ 0.8 & : z \in [2500, 3750), \\ 3.24 & : z \in [3750, 5000]. \end{cases} \tag{4.3}$$

In the initial time interval with $g_0 = 2$ a one-pulse solution quickly forms. When g_0 is increased to 5.5, the solution bifurcates to a three-pulse solution. When g_0 is decreased to 0.8, the one-pulse solution reforms (different than the original one-pulse). Finally, a two-pulse solution is formed when we increase the gain value g_0 to 3.24. The full PDE solution, which represents the *truth*, is performed using a Fast Fourier Transform discretization in the spatial domain in combination with a standard 4th-order Runge-Kutta time-stepping scheme in time.

In implementing the adaptive POD algorithm, the low-dimensional (rank r) Galerkin truncation for (4.1) must be computed. Following (2.4)-(2.7) leads to the low-rank system:

$$\begin{aligned} \frac{db_k}{dz} = &\left(\frac{iD}{2} + g(z)\tau\right)\left\langle \sum_{j=1}^{r} b_j \boldsymbol{\phi}_j'', \boldsymbol{\phi}_k \right\rangle + (\beta + i\gamma)\left\langle \left| \sum_{j=1}^{r} b_j \boldsymbol{\phi}_j \right|^2 \sum_{j=1}^{r} b_j \boldsymbol{\phi}_j, \boldsymbol{\phi}_k \right\rangle \\ &- (\mu + i\nu)\left\langle \left| \sum_{j=1}^{r} b_j \boldsymbol{\phi}_j \right|^4 \sum_{j=1}^{r} b_j \boldsymbol{\phi}_j, \boldsymbol{\phi}_k \right\rangle \\ &+ (g(z) - \delta)\left\langle \sum_{j=1}^{r} b_j \boldsymbol{\phi}_j, \boldsymbol{\phi}_k \right\rangle, \quad k = 1, \cdots, r, \end{aligned} \tag{4.4}$$

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

where the $\langle \cdot\ ,\ \cdot \rangle$ denotes the inner product. Under discretization, the inner product is simply a dot product for the vectors. This prescribes the equations of motion once a dimensionality reduction test has been performed and a low-rank approximation has been determined to be suitable. The flowchart in Fig. 1 outlines the algorithm that is now implemented for solving (4.1). Fig. 2 shows the effectiveness of the adaptive POD solution to the CQGLE with the time-dependent g_0 values given by (4.3). The red regions denote where the full PDE system is implemented while the blue regions are the dimensionality reduced areas. The dominance of the blue regions ensures a significant computational savings.

4.2. Model 2: Waveguide array mode-locking

A second example model considered is that of a system of coupled partial differential equations. Much like the CQGLE example, the application context is in the arena of optical mode-locking. The waveguide array mode-locking (WGAML) model describes the pulse formation and stabilization process in a optical fiber system where saturable absorption if provided by nonlinear, evanescent mode coupling in a waveguide array. Included in the laser cavity is a saturating, bandwidth limited gain which is provided by erbium doped fiber amplification. As with the CQGLE model, the multi-pulsing transition phenomenon is observed. The following equations model the prototypical dynamics of the waveguide array mode-locked laser system [39, 40]:

$$i\frac{\partial A_0}{\partial z} + \frac{D}{2}\frac{\partial^2 A_0}{\partial t^2} + \beta|A_0|^2 A_0 - ig(z)\left(1+\tau\frac{\partial^2}{\partial t^2}\right)A_0 + i\gamma_0 A_0 + CA_1 = 0, \tag{4.5a}$$

$$i\frac{\partial A_1}{\partial z} + c(A_0 + A_2) + i\gamma_1 A_1 = 0, \tag{4.5b}$$

$$i\frac{\partial A_2}{\partial z} + cA_1 + i\gamma_2 A_2 = 0, \tag{4.5c}$$

where the saturating gain (L^2 energy control) is given by

$$g(z) = \frac{2g_0}{1 + \|A_0\|^2/e_0}. \tag{4.6}$$

Here A_0, A_1, A_2 are the envelope of the electric field in waveguides 0, 1, 2 respectively, and $\|A_0\|^2 = \int |A_0|^2 dt$ where the limits of integration are over the entire computational domain. Note that the z variable is the time-like variable while t is the space-like variable for this system. This is consistent with optics coordinates which are written in the frame-of-reference of the propagating pulse. Saturating gain is only provided to the central waveguide 0. Thus the evolution equation for A_0 is affected by the gain bandwith τ and gain strength/saturation parameter g_0 and e_0. The normalized parameters in the WGAML are as following: D represent the strength and type of chromatic dispersion, β is the strength of the Kerr nonlinearity, τ is the bandwidth of the gain, γ_j is the linear loss in the jth waveguide and C is the strength of coupling between adjacent waveguide. In our numerical experiment the parameters above are chosen to be: $D = 1,\ \beta = 8,\ C = 5,$ $\gamma_0 = 0,\ \gamma_1 = 0,\ \ \gamma_2 = 10,\ \tau = 0.1,\ e_0 = 1.$

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

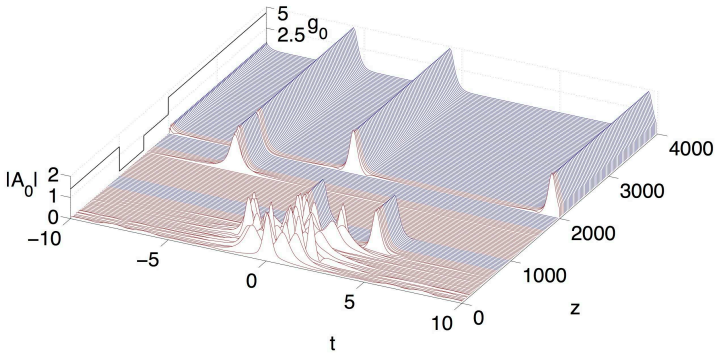


Figure 3: Waterfall plot of solution to the WGAML obtained by the adaptive POD algorithm. The simulation matches the full PDE simulation as demonstrated in the error analysis of Sec. 5. The red region corresponds to the region where the full PDE solver is evoked by the algorithm while the blue region corresponds to where the low dimensional POD solver is chosen. The time-dependent bifurcation parameter g_0 is illustrated at left. Note that in the approximate region where $t \in [1000, 1600]$, the numerical solution produces low-amplitude, radiation waves which propagate in the cavity. the amplitude is low-enough not to be discernable on the displayed scale of $|A_0 \in [0, 2]$.

The gain parameter g_0 is allowed to be varied and is the bifurcation parameter in the experiment. Single-pulse mode locking, breathers, chaotic and multi-pulse solutions can all be obtained as the value of g_0 is varied [39, 40]. In order to test whether the adaptive method is able to handle the transition between different types of dynamics, we let g_0 to be the step function:

$$g_0(z) = \begin{cases} 3.5 & : z \in [0, 1000), \\ 0.6 & : z \in [1000, 1500), \\ 2.3 & : z \in [1500, 2000), \\ 4.3 & : z \in [2000, 5000]. \end{cases} \tag{4.7}$$

For these parameters, the dynamics first exhibits a stable two pulse mode-locking scenario. Upon decreasing the gain to $g_0 = 0.6$, a low amplitude one-pulse solution if formed. Raising the gain to $g_0 = 2.3$ produces a high-amplitude one-pulse mode locking solution that bifurcates to a three-pulse solution immediately after g_0 is increased to 4.3. By Applying the adaptive POD algorithm to WGAML, we generate the same dynamics as that observed in the full PDE simulation (See Fig. 3).

4.3. Model 3: Two-dimensional cubic-quintic Ginzburg-Landau equation

As a final example, we consider the two-dimensional CQGLE (2D-CQGLE) proposed to model cavity soliton formation [44]. The model is essentially a two-dimensional generalization of the CQGLE already considered. However, the physical role of temporal dispersion is replaced by spatial diffraction. The 2D-CQGLE describes the time evolution of cavity solitons confined in the transverse spatial dimensions of a transverse laser cavity [44]:

$$i\frac{\partial u}{\partial t} + \frac{D}{2}\nabla^2 u + \gamma|u|^2 u - \nu|u|^4 u = iRu, \tag{4.8}$$

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

where$\nabla^2 = \partial_x^2 + \partial_y^2$ is the two dimensional Laplacian operator in the transverse plane and R is the dissipation operator defined as

$$R = g(t)(1 + \tau\nabla^2) - \delta + \beta|u|^2 - \mu|u|^4 \tag{4.9}$$

with the saturating gain given by

$$g(t) = \frac{2g_0}{1 + \|u\|^2/e_0}. \tag{4.10}$$

Here $u(x, y, t)$ represents the averaged complex electric field envelope in the cavity and is dependent on time t and the spatial transverse variables x and y. The parameter D denotes the averaged diffraction coefficient and is positive (negative) for systems with positive (negative) refractive index. The parameters γ and ν are the cubic and quintic self-phase modulation terms respectively that are created by the combined effect of the nonlinear crystal and the saturable quantum well. The gain $g(t)$ depends on the input pump strength g_0 and the total cavity energy given by $\|u\|^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|u|^2 dxdy$ which is parametrized by the saturating energy e_0. Additionally, the bandwidth of the gain is limited, only allowing certain spatial frequencies to be amplified. The range of admissible frequencies is controlled by the bandwidth parameter τ. The parameter δ characterizes the total loss of the laser while β describes the intensity discrimination in the laser. The quintic loss term associated with μ saturates the cubic gain term in order to prevent the soliton solution from blowing up.

The soliton solutions supported by the equation are determined by the supplied amplification, which can be controlled by the pumping strength g_0 or the saturating energy e_0. In the numerical experiments, the saturating energy e_0 is fixed. Thus the gain level can be controlled by g_0 completely. More energy is injected to the laser cavity when g_0 is increased resulting in the solution bifurcating from a single-pulse solution to a double-pulse solution, for instance. Further increasing g_0 can generate steady-state three-pulse and four-pulse solutions as well. In the numerical experiments that follow, the equation parameters are set to the following: $D = 0.5,\ \gamma = 1.5,\ \nu = 0.08,\ e_0 = 1,\ \tau = 0.1,\ \delta = 1,$ $\beta = 0.8$ and $\mu = 0.5$. The time dependent bifurcation parameter g_0 is chosen to be the following step function:

$$g_0 = \begin{cases} 1.45 & : t \in [0, 4500), \\ 3.6 & : t \in [4500, 9000). \end{cases} \tag{4.11}$$

The increase in the gain assures that the solution bifurcates from a single localized cavity soliton to two localized cavity solitons in the transverse field.

The adaptive POD time-stepper applied to the 2D-CQGLE yields the same results as the full PDE simulations of (4.8). Fig. 4 shows a topographical view of the dynamics at different time points as the system undergoes a dynamical state change. Additionally, the bottom panel demonstrates when the system solves the full PDE system (shaded region) versus using a dimensionally reduced bases. The adaptive solver switches between the full PDE solver and the low dimensional solver automatically when appropriate, i.e, in

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

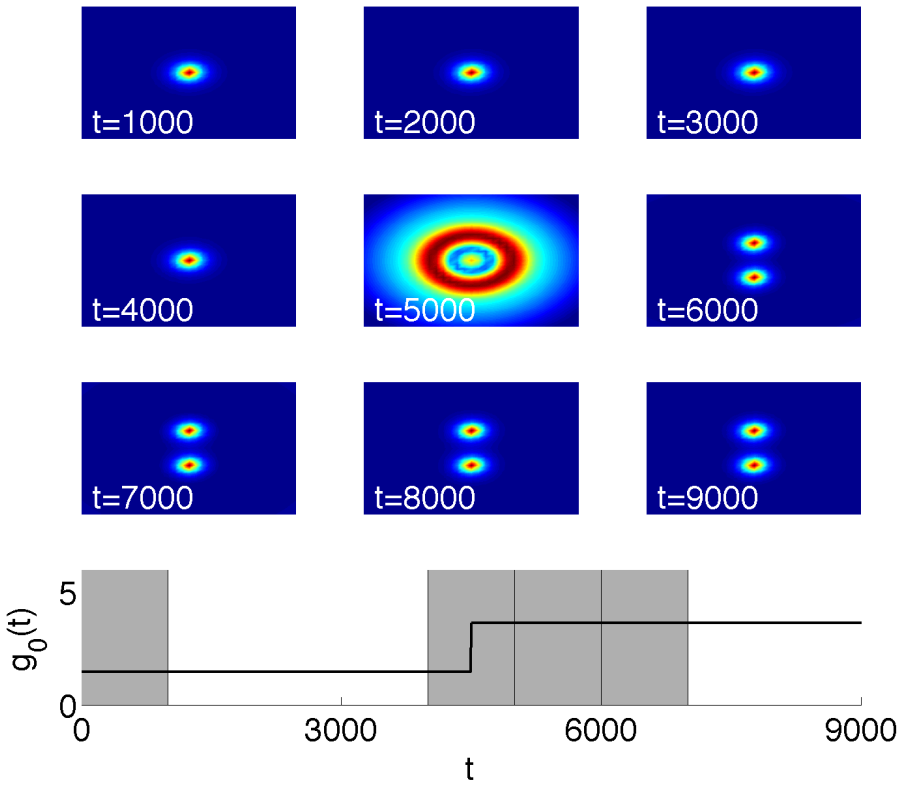


Figure 4: Dynamic evolution of the 2D-CQGLE obtained by using the adaptive POD algorithm. The solution matches the evolution of the full PDE solution (See Sec. 5). As the time dependent bifurcation parameter g_0 is increased, the solution dynamically evolves from a one cavity soliton to a two cavity soliton solution. The bottom panel indicates the changes in the bifurcation parameter as a function of time. In this bottom panel the shaded region corresponds to the region where the full PDE solver is evoked by the algorithm while the unshaded (white) region corresponds to where the low dimensional POD solver is chosen.

the transition region between the one-pulse and two-pulse region. The dimensionality reduction in this case is exceptionally efficient for solving the governing 2D-CQGLE as the 2D discretization yields an $\mathcal{O}(n^2)$ system of ODEs unlike the $\mathcal{O}(n)$ system in one dimension.

5. Computation efficiency and error analysis

Two standard criteria are used to evaluate the adaptive dimensionality time-stepping scheme: the computational efficiency and its accuracy. Since the truncated POD basis approximates well the full PDE solution with only a few modes, the potential for computational improvement is significant since the time-stepping can now be performed with only a few modes of an ODE solver. In what follows, an assessment will be made of the computational time and error of the scheme applied to the three examples of the last section.

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

Table 1: CQGLE CPU time. For the low-rank POD approximation, the computation time includes testing for low-rank structure and performing the SVD to extract POD modes.

method	CPU time
FFT	$2.5282e+3$
POD 1mode	$2.4556e+1$
POD 2modes	$6.4933e+1$
POD 3modes	$1.1852e+2$
POD 8modes	$1.5765e+2$
POD 16modes	$2.8070e+2$
POD 32modes	$4.7241e+3$
POD 64modes	$1.4457e+4$
POD 128modes	$4.3655e+4$

5.1. Computational time

Since the adaptive dimensionality time-stepping algorithm switches between the full PDE simulation and the low dimensional POD solver automatically, the complexity of the algorithm is difficult to evaluate directly. However, the complexity of the low dimensional POD solver can be studied in order to evaluate the potential computational savings.

In this evaluation, the CQGLE (model 1) is chosen as the test problem. The model is solved as previously using a PDE simulation based upon spectral techniques (Fast Fourier Transform) in interval propagation interval $Z \in [0,50]$ for a fixed g_0 supporting a single pulse. The spatial domain $T \in [-20,20]$ is discretized into $n = 512$ Fourier modes with an initial condition $u(0,t) = \mathrm{sech}(t)$. After the sample (or truth) solution in the interval $Z \in [0,50]$ is obtained, the POD algorithm is applied to solve the equation for the propagation interval $Z \in [50,2000]$. The computation time of the POD solutions are then compared with that of solution computed by full PDE simulation and the results are showed in the Table 1. The table shows that if 16 modes are kept in the POD algorithm, it only cost 10% of the computation time compared with full PDE simulation. As the modes are increased, the POD algorithm can become more costly as there is the extra cost of computing inner products of the POD modes etc. However, 16 modes are shown to be more than enough modes to ensure that the dominant error arises from the Runge-Kutta stepper itself (error of 10^{-6}). In fact, in most cases, by keeping only 8 modes in the POD solver, an accuracy of 10^{-7} can be achieved. Thus the 512 modes are swapped out for an 8 mode representation. This represents a significant, order-of-magnitude computational savings.

We further considered the adaptive POD algorithm applied to all three models presented in the previous section: the CQGLE, WGAML and 2D-CQGLE. The computation times of the adaptive method was compared with that of full PDE simulations. For CQGLE, the propagation interval was $Z \in [0,5000]$ (using $\Delta Z = 0.2$) with spatial domain $T \in [-20,20]$ and $n = 1024$ Fourier modes and initial conditions $u(0,t) = \mathrm{sech}(t)$. For the WGAML model, the propagation interval was $Z \in [0,4000]$ with spatial domain $T \in [-10,10]$ and $n = 512$ Fourier modes with initial conditions $A_0(0,t) = \mathrm{sech}(t)$, $A_1(0,t) = 0$ and $A_2(0,t) = 0$. For the 2D-CQGLE, the time interval $t \in [0,9000]$ with transverse domain

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

Table 2: CPU Time: Adaptive POD vs FFT.

	CQGLE ($n = 1024$)	WGAML ($n = 512$)	2D-CQGLE ($nx = ny = 512$)
FFT	$1.2786e+4$	$1.5407e+4$	$1.6347e+5$
Adaptive POD	$1.0864e+3$	$5.2564e+3$	$7.0931e+4$

Table 3: Computation Time distribution for adaptive POD method. The sum of the computational times (columns) for the various components is given in the second row of Table 2.

	CQGLE	WGAML	2D-CQGLE
FFT	$8.9206e+2$	$4.0525e+3$	$6.5742e+4$
POD	$1.9407e+2$	$1.2031e+3$	$5.1846e+3$
Test	$1.0700e-2$	$3.3051e-1$	$1.3216e+0$
SVD	$3.2981e-1$	$4.9317e-1$	$3.1109e+0$

$[x, y] \in [-20, 20] \times [-20, 20]$ with $n = 512$ Fourier mode discretization in each direction and initial conditions $u(0, x, y) = \exp(-x^2/4 - y^2/4)/2$. A computation time comparison using dimensionality reduction for these three prototype models is shown in Table 2. As is evident, the adaptive POD solver significantly reduces the computation time, typically by an order of magnitude for the same accuracy. In addition, we also tracked the computation time required of each part of the adaptive POD method, which is listed in Table 3. The SVD decomposition and switching test only takes a very small amount of time suggesting that the algorithm is efficient.

5.2. Error analysis

To assess the error of the adaptive dimensionality time-stepping algorithm, simulations are compared directly to highly resolved PDE simulations. To demonstrate this in concrete terms, the solution of the CQGLE is sampled in the interval $Z \in [0, 50]$ and the SVD is applied to the sample solution to generate the POD modes. A rank $r = 1, 2, 3, 8, 16, 32$ and 64 POD mode reduction is considered. The relative error of these reductions is defined as

$$e(t) = \max_{t \in [0,50]} \frac{\left\| |u_{POD}(t)| - |u_{FFT}(t)| \right\|_\infty}{\|u_{FFT}(t)\|_\infty}. \tag{5.1}$$

The error for the dimensionality reductions are showed in Table 4. Note the rapid convergence of the low dimensional POD solver when the number of modes is increased.

6. Conclusions

Dimensionality reduction methods are becoming common place for analyzing and assessing the dynamics and evolution of high-dimensional dynamical systems arising in the engineering, biological and physical sciences. This is due, in large part, to the very high-dimensional nature of such systems when solved using computational discretization methods. And although algorithms are often readily available for time-stepping, the sheer size of

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

Table 4: Convergence of low dimensional POD solver.

Tol	1mode	2 modes	3 modes	8 modes	16 modes	32 modes	64 modes
10^{-5}	2.7×10^{-3}	2.7×10^{-3}	3.7×10^{-7}	1.4×10^{-7}	1.4×10^{-7}	1.4×10^{-7}	1.4×10^{-7}
10^{-16}	2.7×10^{-3}	6.4×10^{-6}	3.7×10^{-7}	1.2×10^{-7}	1.3×10^{-7}	2.7×10^{-7}	3.9×10^{-7}

the resulting discretization renders such computational solutions inefficient and extremely costly. Ultimately, such techniques highlight the need for data analysis methods applied to the underlying dynamical systems.

Our objective in this manuscript is to consider efficient time-stepping algorithms and schemes for solving seemingly high-dimensional dynamical systems. The critical observation is that high-dimensional dynamical systems almost always exhibit dynamical evolutions that occur on low-dimensional manifolds, thus suggesting that if this manifold can be found, the dynamics would then have a natural basis on which to evolve. Moreover, the high-dimensional simulation (of dimension n) that occurs on the discretized basis reduces to simulation on the low-dimensional manifold (or rank r). The computational savings are readily apparent since $r \ll n$. Much like adaptive time-stepping algorithms which take as large a step as possible while preserving a prescribed accuracy, we demonstrate that an adaptive dimensionality algorithm allows for a greatly improved efficiency in solving a number of example PDEs, often performing computations an order or magnitude quicker than their standard discretization counterpart. Such computational savings should clearly be taken advantage of in any time-stepping scheme and we provide a number of robust example methods for performing the required dimensionality reduction and inflation associated with our proposed algorithm.

The adaptive dimensionality for time-stepping has many connections to the existing reduced order modeling (ROM) literature [2]. Indeed, the ROM literature suggests a number of additional advantages that can be exploited, as well as a number of cautionary tales that one must be mindful of. As for cautionary tales, the ROMs community has long understood that a POD-Galerkin projected model can often by unstable [2, 31]. This is problematic, but not necessarily relevant for the adaptive time-stepping suggested here. Specifically, the goal of the algorithm is to use *cheap* computational checks to see if lower-rank approximations can be used. If such low-rank approximations cannot be exploited, then the computation simply proceeds as normal. Often in ROMs, one must by necessity commit to and construct the ROM in order to even enable the computation. No such commitment to a ROM is required of the proposed method. Indeed, if a poor ROM model is constructed, then the algorithm quickly continues with the full-state simulation. If however adjusting the dimension to a low-rank system can be accomplished, then there are a number of innovations in the ROM community that can be used to enhance the reduction procedure. Namely, sparse (gappy) sampling of the state space for interpolating the nonlinear terms required for the subspace projection. Sparsity has been established as a critically enabling mathematical framework for model reduction through methods such as gappy POD [23] and its variants [2]. Indeed, efficiently managing the computation of the nonlinearity was recognized early on in the ROMs community, and a variety of techniques where proposed

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

to accomplish the task. The empirical interpolation method (EIM) [20] and one of its successful variants, the POD-tailored discrete empirical interpolation method (DEIM) [21], have provided a greedy algorithm that allows for nearly optimal reconstructions of nonlinear terms of the original high-dimensional system. Thus low-rank models can be build in a computationally efficient manner.

Given the demonstrated success of the method, some outstanding issues still remain. In particular, the method is limited to dynamical systems that exhibit slow manifold dynamics that act as attractors to the system. If the dynamics does not have such underlying structure, then low-rank approximations generally do not hold. The method also fails during transient dynamical responses. Finally, if there are low-rank, slow manifolds that are subject to invariances such as translation or rotation, the method as shown here fails to capture the low rank structure. However, the algorithm can be easily modified to handle such invariances [41]

Acknowledgments We are especially indebted to Steven Brunton and Matthew Williams for invaluable comments and constructive criticisms on the work presented. J. N. Kutz acknowledges support from the U.S. Air Force Office of Scientific Research (AFOSR) (FA9550-15-1-0385).

References

[1] A. Quarteroni and G. Rozza, Reduced Order Methods for Modeling and Computational Reduction, Springer, 2014.
[2] P. Benner, S. Gugercin and K. Willcox, *A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems*, SIAM Rev., 57(4) (2015), pp. 483–531.
[3] K. Pearson, *On lines and planes of closest fit to systems of points in space*, Phil. Mag., 2 (1901), pp. 559–572.
[4] H. Hotelling, *Analysis of a complex of statistical variables into principal components*, J. Educational Psychol., 23 (1933), pp. 498–520.
[5] K. Karhunen, *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*, Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys., 3(37) (1947), pp. 1–79.
[6] M. Loéve, Probability Theory, 4th ed., Graduate Texts in Mathematics 46, Springer-Verlag, 1978.
[7] L. Sirovich, *Turbulence and the dynamics of coherent structures. I - Coherent structures, II - Symmetries and transformations, III - Dynamics and scaling*, Quart. Applied Math., 45 (1987), pp. 561–590.
[8] P. Holmes, J. L. Lumley and G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge, 1998.
[9] J. N. Kutz, Data-Driven Modeling and Scientific Computing, Oxford, 2013.
[10] S. McCormick, Multigrid Methods(Frontiers in Applied Mathematics), SIAM, 1987.
[11] B. Engquist, O. Runborg and Y. H. Tsai, Numerical Analysis of Multiscale Computations: Proceedings of a Winter Workshop at the Banff International Research Station, Lect. Notes Comp. Sci., Springer, New York, 2009.
[12] P. G. Martinsson, V. Rokhlin and M. Tygert, *A randomized algorithm for the approximation of matrices*, App. Comp. Harmonic Ana., 30(1) (2011), pp. 47–68.

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

[13] N. Halko, P. Martinsson and J. Tropp, *Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions*, SIAM Rev., 53(2) (2011), pp. 217–288.
[14] N. Halko, P. G. Martinsson, Y. Shkolnisky and M. Tygert, *An algorithm for the principal component analysis of large data sets*, SIAM J. Sci. Comput., 33(5) (2011), pp. 2580–2594.
[15] J. Xia, S. Chandrasekaran, M. Gu and X. S. Li, *Fast algorithms for hierarchically semiseparable matrices*, Numer. Linear Algebra Appl., 17(6) (2010), pp. 953–976.
[16] M. Cross and P. Hohenberg, *Pattern formation outside of equilibrium*, Rev. Modern Phys., 65 (1993), pp. 851–1112.
[17] H. Schaeffer, S. Osher, R. Caflisch and C. Hauck, *Sparse dynamics for partial differential equations*, Proceedings of the National Academy of Sciences of the United States of America, 110(17) (2013), pp. 6634–6639.
[18] S. Jokar, V. Mehrmann, M. Pfetsch, H. Yserentant, *Sparse approximate solution of partial differential equations*, J. Applied Numerical Mathematics archive, 60(4) (2010), pp. 452–472.
[19] M. Williams, P. Schmid and J. N. Kutz, *Hybrid reduced-order integration with proper orthogonal decomposition and dynamic mode decomposition*, SIAM Multiscale Model. Simul., 11(2) (2013), pp. 522–544.
[20] M. Barrault, Y. Maday, N. C. Nguyen and A. T. Patera, *An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations*, Comptes Rendus Mathematique, 339(9) (2004), pp. 667–672.
[21] S. Chaturantabut and D. C. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32(5), (2010), pp. 2737–2764.
[22] D. Amsallem, M. Zahr, C. Farhat, *Nonlinear Model Order Reduction Based on Local Reduced-Order Bases*, Int. J. Numer. Meth. Eng., 92(10) (2012), pp. 891–916.
[23] R. Everson and L. Sirovich, *The Karhunen-Loéve procedure for gappy data*, J. Opt. Soc. Am. A, 12(8) (1995), pp. 1657–1664.
[24] K. Wilcox, *Unsteady flow sensing and estimation via the happy proper orthogonal decomposition*, Comput. Fluids, 35(2) (2006), pp. 208–226.
[25] K. Carlberg, *Adaptive h-refinement for reduced-order models*, Int. J. Numer. Meth. Eng., 102(5) (2015), pp. 1192–1210.
[26] D. Dychelynck, F. Vincent and S. Cantournet, *Multidimensional a priori hyper-reduction of mechanical models involving internal variables*, Comput. Methods Appl. Mech. Engrg, 225-228(17-20) (2012), pp. 28–43.
[27] T. Kim and D. L. James, *Skipping steps in deformable simulation with online model reduction*, Acm Transactions on Graphics, 28(5) (2009), pp. 1–9.
[28] G. Oxberry, T. Kostova-Vassilevska, W. Arrighi and K. Chand, *Limited-memory adaptive snapshot selection for proper orthogonal decomposition*, Int. J. Numer. Meth. Eng., 109 (2017), pp. 198–217.
[29] A. Varshney, S. Pitchaiah and A. Armaou, *Feedback Control of Dissipative PDE Systems using Adaptive Model Reduction*, AIChE Journal, 55(4) (2010), pp. 906-918.
[30] N. L. Trefethen and D. Bau, Numerical Linear Algebra, SIAM, 1997.
[31] K. Carlberg, M. Barone and A. Harbir, *Galerkin v. least-squares Petrov-Galerkin projection in nonlinear model reduction*, J. Comput. Phys., 26(2) (2016), pp. 24–31.
[32] C. Homescu, L. Petzold, R. Serban, *Error Estimation for Reduced-Order Models of Dynamical Systems*, SIAM J. Numer. Anal., 43(4) (2006), pp. 1693–1714.
[33] B. Peherstorfer and K. Willcox, *Online adaptive model reduction for nonlinear systems via low-rank updates*, SIAM J. Sci. Comput., 37(4) (2015), pp. 2123-2150.
[34] C. Baker, K. Gallivan and P. Van Dooren, *Low-rank incremental methods for computing dom-*

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press

inant singular subspaces, Linear Algebra Appl., 436(8) (2012), pp. 2866–2888.
[35] M. Brand, *Fast low-rank modifications of the thin singular value decomposition*, Linear Algebra Appl., 415(1) (2006), pp. 20–30.
[36] N. Halko, P. G. Martinsson and J. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53(2) (2011), pp. 217–288.
[37] I. Akhtar, Z. Wang, J. Borggaard and T. Iliescu, *A new closure strategy for proper orthogonal decomposition reduced-order models*, J. Comput. Nonlin. Dyn., 7(3) (2012), 034503.
[38] J. Östh, B. Noack, S. Krajnović, D. Barros and J. Borée, *On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high Reynolds number flow over an Ahmed body*, J. Fluid Mech., 747(3) (2014), pp. 518–544.
[39] J. N. Kutz and B. Sandstede, *Theory of passive harmonic mode-locking using wave-guide arrays*, Optics Express, 16(2) (2008), pp. 636–650.
[40] M. Williams, E. Shlizerman and J. N. Kutz, *The multi-pulsing transition in mode-locked lasers: a low-dimensional approach using waveguide arrays*, J. Opt. Soc. Am. B, 27(12) (2010), pp. 2471–2481.
[41] C. W. Rowley and J. E. Marsden, Reconstruction Equations and the Karhunen-Loeve Expansion, Physica D, 2000.
[42] J. N. Kutz, Mode-Locked Soliton Lasers, SIAM Review, 2006.
[43] E. Ding, E. Shlizerman and J. N. Kutz, *Generalized master equation for high-energy passive mode-locking: The sinusoidal Ginzburg-Landau equation*, IEEE J. Quantum Electron., 47(5) (2011), pp. 705–714.
[44] E. Ding, K. Lu and J. N. Kutz, *Stability analysis of cavity solitons governed by the cubic-quintic Ginzburg-Landau equation*, J. Physics B, 44(6) (2011), pp. 157–160.

https://doi.org/10.4208/nmtma.2017.m1624 Published online by Cambridge University Press