SIAM J. SCI. COMPUT.
Vol. 39, No. 5, pp. B778–B796

NONLINEAR MODEL ORDER REDUCTION VIA DYNAMIC MODE DECOMPOSITION[*]

ALESSANDRO ALLA[†] AND J. NATHAN KUTZ[‡]

Abstract. We propose a new technique for obtaining reduced order models for nonlinear dynamical systems. Specifically, we advocate the use of the recently developed dynamic mode decomposition (DMD), an equation-free method, to approximate the nonlinear term. DMD is a spatio-temporal matrix decomposition of a data matrix that correlates spatial features while simultaneously associating the activity with periodic temporal behavior. With this decomposition, one can obtain a fully reduced dimensional surrogate model and avoid the evaluation of the nonlinear term in the online stage. This allows for a reduction in the computational cost and, at the same time, accurate approximations of the problem. We present a suite of numerical tests to illustrate our approach and to show the effectiveness of the method in comparison to existing approaches.

Key words. nonlinear dynamical systems, proper orthogonal decomposition, dynamic mode decomposition, data-driven modeling, reduced order modeling, dimensionality reduction

AMS subject classifications. 65L02, 65M02, 37M05, 62H25

DOI. 10.1137/16M1059308

1. Introduction.

Reduced order models (ROMs) are of growing importance in scientific computing as they provide a principled approach to approximating high-dimensional PDEs with low-dimensional models. Indeed, the dimensionality reduction provided by ROMs help to reduce the computational complexity and time needed to solve large-scale engineering systems [31, 4], enabling simulation-based scientific studies not possible even a decade ago. One of the primary challenges in producing the low-rank dynamical system is efficiently projecting the nonlinearity of the governing PDEs (inner products) [3, 9] on to the proper orthogonal decomposition (POD) [21, 13, 37] basis. This fact was recognized early on in the ROM community, and methods such as gappy POD [11, 38, 39] were proposed to more efficiently enable the task. More recently, the empirical interpolation method (EIM) [3] and the simplified discrete empirical interpolation method (DEIM) [9] for POD [21, 13, 37] have provided a computationally efficient method for discretely (sparsely) sampling and evaluating the nonlinearity. These broadly used and highly successful methods ensure that the computational complexity of ROMs scale favorably with the rank of the approximation, even for complex nonlinearities. Recently, DEIM has been investigated also in [29, 30]. As an alternative to the EIM/DEIM architecture, we propose using the recently developed dynamic mode decomposition (DMD) for producing low-rank approximations of the PDE nonlinearities. DMD provides a decomposition of data into spatio-temporal modes that correlates the data across spatial features (like POD) but also associates the correlated data to unique temporal Fourier modes, allowing for a computationally efficient regression of the nonlinear terms to a least-square fit lin-

[*] Submitted to the journal's Computational Methods in Science and Engineering section February 1, 2016; accepted for publication (in revised form) March 29, 2017; published electronically September 12, 2017.

http://www.siam.org/journals/sisc/39-5/M105930.html

Funding: The first author is supported by U.S. Department of Energy grant DE-SC0009324. The second author is supported by the Air Force Office of Scientific Research (FA9550-15-1-0385).

[†] Department of Scientific Computing, Florida State University, Tallahasee, FL 32306 (aalla@fsu.edu).

[‡] Department of Mathematics, University of Washington, Seattle, WA 98195 (kutz@uw.edu).

ear dynamics approximation. We demonstrate that the POD-DMD method produces a viable ROM architecture, scaling favorably in computational efficiency relative to POD-DEIM.

The POD-Galerkin method has been widely used in the scientific computing community. The primary challenge in producing the low-rank dynamical system is efficiently projecting the nonlinearity (inner products) to the POD basis, leading to numerous innovations in the ROM community for interpolating the projection. This ensures that the computational cost of evaluating the nonlinearity remains proportional to the rank of the reduced POD basis. The DEIM approach combines projection with interpolation by selecting interpolation indices to specify an interpolation-based projection for a nearly optimal ℓ_2 subspace approximating the nonlinearity. The EIM/DEIM are not the only methods developed to reduce the complexity of evaluating nonlinear terms; see, for instance, the missing point estimation (MPE) [2], the *best points* method [27], the gappy POD technique [11, 38, 39], or the so-called GNAT [7] method. However, they have been successful in a large number of diverse applications and models [9]. In any case, MPE, gappy POD, and EIM/DEIM all use a small selected set of spatial grid points to avoid evaluation of the expensive inner products required to evaluate nonlinear terms.

An alternative to these sparse sampling techniques for evaluating the nonlinear inner products is the DMD method. Like EIM/DEIM, it requires a singular value decomposition (SVD) to generate the approximation. We demonstrate that the DMD provides a highly efficient approximation for ROMs, performing a more rapid evaluation of the nonlinear terms in comparison to the EIM/DEIM methods. At its core, the DMD method can be thought of as an ideal combination of spatial dimensionality-reduction techniques, such as POD, with Fourier transforms in time. It also allows for further innovations that integrate the DMD with key concepts from multiresolution analysis [18] and sparsity/compression [6], allowing one to potentially generalize the proposed method to multiscale physics problems at greatly improved speeds. For the sake of completeness, we also study the DMD as a data-driven method for nonlinear dynamical systems and compare it with the proposed techniques. We refer to [5] for a comparison between POD and DMD for the shallow water equation.

The structure of the paper is as follows. In section 2 we recall the POD method and the DEIM applied to a general dynamical system. Section 3 explains the DMD and compares the DMD method used as equation-free or as a Galerkin projection method. The coupling between POD and DMD is explained in section 4. Finally, numerical tests are presented in section 5. Throughout the paper we use the following notation: all matrices and vectors are in bold letters. The basis functions are denoted by the matrix $\mathbf{\Psi}$ with different superscripts denoting how we computed the basis, e.g., $\mathbf{\Psi}^{\text{POD}}$ represents the basis functions from the POD method. The rank of the POD basis functions is ℓ, whereas the rank of the nonlinear term is k.

2. Problem formulation.

In what follows, we consider a system of ordinary differential equations:

$$\begin{cases} \mathbf{M}\dot{\mathbf{y}}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{f}(t, \mathbf{y}(t)), & t \in (0, T], \\ \mathbf{y}(0) = \mathbf{y_0}, \end{cases} \tag{2.1}$$

where $\mathbf{y_0} \in \mathbb{R}^n$ is a given initial data, $\mathbf{M}, \mathbf{A} \in \mathbb{R}^{n \times n}$ given matrices, and $\mathbf{f} : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$ a continuous function in both arguments and locally Lipschitz-type with respect to the second variable. It is well known that under these assumptions there exists a unique solution for (2.1).

This wide class of problems arises in many applications, especially from the numerical approximation of partial differential equations. In such cases, the dimension of the problem n is the number of spatial grid points used from discretization and it can be very large. The solution of system (2.1) may be very expensive and therefore it might be useful to simplify the complexity of the problem by means of reduced order modeling techniques.

2.1. The POD method and reduced order modeling.

One popular method for reducing the complexity of the system is the so-called POD. The idea is detailed here for completeness. We build a grid in time $0 := t_1 < t_2 < \cdots < t_m \leq T$. Let us assume we know the exact solution of (2.1) on the time grid points t_j, $j \in \{1, \ldots, m\}$. Our aim is to determine a POD basis of rank $\ell \ll n$ to describe the set of data collected in time by solving the following minimization problem:

$$\min_{\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_\ell \in \mathbb{R}^n} \sum_{j=1}^{m} \left\| \mathbf{y}(t_j) - \sum_{i=1}^{\ell} \langle \mathbf{y}(t_j), \boldsymbol{\psi}_i \rangle \boldsymbol{\psi}_i \right\|^2 \quad \text{such that } \langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle = \delta_{ij}, \tag{2.2}$$

where $\mathbf{y}(t_j)$ are the so-called snapshots, e.g., the solution of (2.1) at a given time t_j. This idea was introduced by Sirovich [35]. The associated norm is given by $\| \cdot \|^2 = \langle \cdot, \cdot \rangle$.

Solving (2.2) we look for an orthonormal basis $\{\boldsymbol{\psi}_i\}_{i=1}^{\ell}$ which minimizes the distance between the sequence $\mathbf{y}(t_j)$ with respect to its projection onto these unknown basis functions. The matrix $\mathbf{Y}$ contains the collection of snapshots $\mathbf{y}(t_j)$ as columns. It is useful to look for $\ell \ll \min\{m, n\}$ in order to reduce the dimension of the problem considered. The solution of (2.2) is given by the SVD of the snapshots matrix $\mathbf{Y} = \boldsymbol{\Psi}\boldsymbol{\Sigma}\mathbf{V}^T$, where we consider the first ℓ columns $\{\boldsymbol{\psi}_i\}_{i=1}^{\ell}$, of the orthogonal matrix $\boldsymbol{\Psi}$. We refer the interested reader to [37] for more details on the topic.

To concretely apply the POD method, the choice of the truncation parameter ℓ plays a crucial role. There are no a priori estimates which guarantee the ability to build a coherent reduced model, but one can focus on heuristic considerations, introduced by Sirovich [35], so as to have the following ratio close to one:

$$\mathcal{E}(\ell) = \frac{\sum_{i=1}^{\ell} \sigma_i^2}{\sum_{i=1}^{d} \sigma_i^2}, \tag{2.3}$$

where σ_i are the singular values of the matrix $\mathbf{Y}$ and d is the rank of the snapshot matrix $\mathbf{Y}$. This indicator is motivated by the fact that the error in (2.2) is given by the singular values we neglect:

$$\sum_{j=1}^{m} \left\| \mathbf{y}(t_j) - \sum_{i=1}^{\ell} \langle \mathbf{y}(t_j), \boldsymbol{\psi}_i \rangle \boldsymbol{\psi}_i \right\|^2 = \sum_{i=\ell+1}^{d} \sigma_i^2. \tag{2.4}$$

We note that the error (2.4) is strictly related to the computation of the snapshots and it is not related to the reduced dynamical system.

Let us assume that we have computed the POD basis functions $\boldsymbol{\psi}_j$. We construct the $n \times \ell$ matrix $\boldsymbol{\Psi}^{\text{POD}}$ whose columns are composed of the first ℓ POD modes. We make the following projection of the dynamics:

$$\mathbf{y}(t) \approx \boldsymbol{\Psi}^{\text{POD}} \mathbf{y}^{\ell}(t), \tag{2.5}$$

where $\mathbf{y}^{\ell}(t)$ are functions from $[0, T]$ to $\mathbb{R}^{\ell}$. We note that we are working with a Galerkin-type projection where we consider only a few basis functions whose support is nonlocal, unlike finite element basis functions. The reduced solution $\mathbf{y}^\ell(t) \in V^\ell \subset V$, where $V^\ell = \text{span}\{\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_\ell\}$.

Inserting the projection assumption (2.5) into the full model (2.1), and making use of the orthogonality of the POD basis functions, the reduced model takes the following form:

$$\left\{ \begin{array}{l} \mathbf{M}^\ell \dot{\mathbf{y}}^\ell(t) = \mathbf{A}^\ell \mathbf{y}^\ell(t) + (\boldsymbol{\Psi}^{POD})^T \mathbf{f}(t, \boldsymbol{\Psi}^{POD} \mathbf{y}^\ell(t)), \\ \mathbf{y}^\ell(0) = \mathbf{y}^\ell_\mathbf{0}, \end{array} \right. \tag{2.6}$$

where $(\mathbf{M}^\ell)_{ij} = \langle \mathbf{M}\boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle, (\mathbf{A}^\ell)_{ij} = \langle \mathbf{A}\boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle \in \mathbb{R}^{\ell \times \ell}$, and $\mathbf{y}^\ell_\mathbf{0} = (\boldsymbol{\Psi}^{\text{POD}})^T \mathbf{y}_0 \in \mathbb{R}^\ell$. We also note that $\mathbf{M}^\ell, \mathbf{A}^\ell \in \mathbb{R}^{\ell \times \ell}$. The system (2.6) is achieved following a Galerkin projection where the basis functions are computed by the POD method given by (2.2). If the dimension of the system is $\ell \ll n$, then a significant dimensionality reduction is accomplished. We note that an error analysis, in the infinite dimensional settings, for $\|\mathbf{y}(t) - \boldsymbol{\Psi}^{\text{POD}} \mathbf{y}^\ell(t)\|$ can be found in [17].

2.2. Discrete empirical interpolation method. For the results in this section we closely follow the presentation in [37]. The ROM introduced in (2.6) is a nonlinear system where the significant challenge with the POD-Galerkin approach is the computational complexity associated with the evaluation of the nonlinearity. To illustrate this issue, we consider the nonlinearity in (2.6):

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

To compute this inner product, the variable $\mathbf{y}^\ell(t) \in \mathbb{R}^\ell$ is first expanded to an n-dimensional vector $\boldsymbol{\Psi}^{\text{POD}} \mathbf{y}^\ell(t) \in \mathbb{R}^n$; then the nonlinearity $\mathbf{f}(t, \boldsymbol{\Psi}^{\text{POD}} \mathbf{y}^\ell(t))$ is evaluated and, at the end, we return back to the ROM. This is computationally expensive since it implies that the evaluation of the nonlinear term requires computing the full, high-dimensional model, and therefore the reduced model is not completely independent of the full dimension n. In many well-known example PDEs where POD-Galerkin is applied, this problem can be circumvented by explicitly expanding the nonlinearity in the POD modes. For example, a two POD mode expansion of a quadratic nonlinearity $\mathbf{y}^2 = (y_1^\ell)^2 \boldsymbol{\psi}_1^2 + 2 y_1^\ell y_2^\ell \boldsymbol{\psi}_1 \boldsymbol{\psi}_2 + (y_2^\ell)^2 \boldsymbol{\psi}_2^2$. Thus the nonlinearity and two mode expansion produce three distinct terms whose POD-Galerkin projection (inner products) can be computed offline. This trick is standard for the Navier–Stokes equation, for instance, where a simple quadratic nonlinearity is present. However, for more complex nonlinearities or sufficiently high rank truncation k, this method remains computationally expensive.

To avoid this computationally expensive, high-dimensional, evaluation, EIM [3] and [9] were introduced. The DEIM algorithm was built upon the mathematical framework of EIM but specifically tailored for POD basis functions and time or parameter dependent PDEs. The interested reader is referred to [9] for further information.

The computation of the POD basis functions for the nonlinear part is related to the set of the snapshots $\mathbf{f}(t_j, \mathbf{y}(t_j))$, where $\mathbf{y}(t_j)$ are already computed from (2.1). We denote with $\mathbf{U} \in \mathbb{R}^{n \times k}$ the POD basis function of rank k of the nonlinear part. The DEIM approximation of $\mathbf{f}(t, \mathbf{y}(t))$ is as follows:

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

where $\mathbf{S} \in \mathbb{R}^{n \times k}$ and $\mathbf{y}^{\text{DEIM}}(t) := \mathbf{S}^T \boldsymbol{\Psi}^{\text{POD}} \mathbf{y}^\ell(t)$. The role of the matrix S is to select interpolation points to evaluate the nonlinearity. The selection is made according to an LU decomposition algorithm with pivoting [9], or following the QR decomposition with pivoting [10]. Furthermore, if each component of the nonlinearity is independent from the other, then the matrix S can be moved into the nonlinearity.

Let us define $\mathbf{\Psi}^{\text{DEIM}} := \mathbf{U}(\mathbf{S}^T\mathbf{U})^{-1}$. We note that $\mathbf{\Psi}^{\text{DEIM}} \in \mathbb{R}^{n\times k}$. Therefore the reduced nonlinearity may be expressed as

$$(\mathbf{\Psi}^{\text{POD}})^T\mathbf{f}^{\text{DEIM}}(t,\mathbf{y}^{\text{DEIM}}(t)) = (\mathbf{\Psi}^{\text{POD}})^T\mathbf{\Psi}^{\text{DEIM}}\mathbf{f}(t,\mathbf{y}^{\text{DEIM}}),$$

where we only select a small (sparse) number of rows of $\mathbf{\Psi}^{\text{POD}}\mathbf{y}^\ell(t)$ by the matrix $\mathbf{S}$. As for the computational expense, the matrices

$$\mathbf{S}^T\mathbf{\Psi}^{\text{POD}} \in \mathbb{R}^{k\times\ell},\ (\mathbf{S}^T\mathbf{U})^{-1} \in \mathbb{R}^{k\times k} \text{ and } (\mathbf{\Psi}^{\text{POD}})^T\mathbf{\Psi}^{\text{DEIM}} \in \mathbb{R}^{\ell\times k}$$

can all be precomputed. All the precomputed quantities are independent of the full dimension n. Additionally, during the iteration process the nonlinearity needs only to be evaluated at the k interpolation points since $\mathbf{S}^T\mathbf{\Psi}^{POD}\mathbf{y}^\ell(t) \in \mathbb{R}^k$. Typically the dimension k is much smaller than the full dimension. This allows the ROM to be completely independent of the full dimension as follows:

$$\begin{cases} \mathbf{M}^\ell\dot{\mathbf{y}}^\ell(t) = \mathbf{A}^\ell\mathbf{y}^\ell(t) + (\mathbf{\Psi}^{\text{POD}})^T\mathbf{\Psi}^{\text{DEIM}}\mathbf{f}(t,\mathbf{y}^{\text{DEIM}}(t)), \\ \mathbf{y}^\ell(0) = \mathbf{y_0}^\ell. \end{cases} \tag{2.7}$$

We note that the only difference with respect to (2.6) is the low-rank approximation of the nonlinear term. The error between $\mathbf{f}(t,\mathbf{y}(t))$ and its DEIM approximation $\mathbf{f}^{\text{DEIM}}$ is given by

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

where different error performance is achieved depending on the selection of the interpolation points in $\mathbf{S}$ as shown in [10].

3. Dynamic mode decomposition. DMD is an *equation-free*, data-driven method capable of providing accurate assessments of the spatio-temporal coherent structures in a given complex system, or short-time future estimates of such a system. It traces its origins to pioneering work of Bernard Koopman in 1931 [15], whose work was revived in a set of papers starting in 2004 [22, 23, 24]. Koopman theory is a dynamical systems tool that provides information about a nonlinear dynamical system via an associated infinite-dimensional linear system. Specifically, it provides a characterization that is readily interpretable in terms of standard methods of dynamical systems. Defining it as a data-driven algorithm, the DMD architecture for modeling complex flows was proposed in [33, 34], and it was shown quickly thereafter that the DMD method is actually a special case of Koopman theory in [32].

Given the connection between DMD and Koopman theory [22, 23, 32], we begin by defining the Koopman operator.

Definition 1 (Koopman operator [15]). *For a dynamical system,*

$$\frac{d\mathbf{y}}{dt} = \mathbf{N}(\mathbf{y}), \tag{3.1}$$

where $\mathbf{y} \in \mathcal{M}$*, an* n*-dimensional manifold. The Koopman operator* $\mathcal{K}$ *acts on a set of scalar observable functions* $g : \mathcal{M} \to \mathbb{C}$ *so that*

$$\mathcal{K}g(\mathbf{y}) = g\left(\mathbf{N}(\mathbf{y})\right)\ . \tag{3.2}$$

This shows that the Koopman operator is a *linear* operator that acts on scalar functions g. In a general setting, the Koopman operator can act on a set of observables g_j that are denoted by components of the vector $\mathbf{g}$. But as already mentioned, the DMD is a specific realization of the Koopman theory. Specifically, the observable is the state space itself so that one considers the linear observable $\mathbf{g}(\mathbf{y}) = \mathbf{y}$. In some cases, a better set of observables $\mathbf{g}(\mathbf{y})$ can be found for decomposing the data and performing a spectral analysis [24, 25, 26, 8].

When considering such a linear observable, the DMD algorithm determines the Koopman eigenvalues and modes directly from data. Specifically, one can use the recent formal definition of the DMD method [36].

Definition 2 (dynamic mode decomposition [36]). *Suppose we have a dynamical system* (3.1) *and two sets of data*

$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \mathbf{y}(t_0) & \mathbf{y}(t_1) & \cdots & \mathbf{y}(t_{m-1}) \\ | & | & & | \end{bmatrix}, \quad \mathbf{Y}' = \begin{bmatrix} | & | & & | \\ \mathbf{y}(t_1) & \mathbf{y}(t_2) & \cdots & \mathbf{y}(t_m) \\ | & | & & | \end{bmatrix} \tag{3.3}$$

with $\mathbf{y}(t_j)$ *an initial condition to* (3.1) *and* $\mathbf{y}(t_{j+1})$ *its corresponding output after some prescribed evolution time* τ *with there being* m *initial conditions considered. The DMD modes are eigenvectors of*

$$\mathbf{A}_\mathbf{y} = \mathbf{Y}'\mathbf{Y}^\dagger, \tag{3.4}$$

where † *denotes the Moore–Penrose pseudoinverse.*

The definition of DMD thus yields the matrix $\mathbf{A}_\mathbf{y}$, which is a finite dimensional approximation of the Koopman operator for a linear observable.

The definition of DMD produces a regression procedure whereby the data snapshots in time are used to produce the best-fit linear dynamical system to the nonlinear evolution $\mathbf{y}(t)$, denoted by the vector $\tilde{\mathbf{y}}(t)$, for the data $\mathbf{Y}$ and $\mathbf{Y}'$. The DMD procedure thus constructs the proxy, approximate linear evolution of (3.1)

$$\frac{d\tilde{\mathbf{y}}}{dt} = \mathbf{A}_\mathbf{y}\tilde{\mathbf{y}} \tag{3.5}$$

with $\tilde{\mathbf{y}}(0) = \tilde{\mathbf{y}}_0$ and whose solution is

$$\tilde{\mathbf{y}}(t) = \sum_{i=1}^{k} b_i \boldsymbol{\psi}_i \exp(\omega_i t)\,, \tag{3.6}$$

where $\boldsymbol{\psi}_i$ and ω_i are the eigenfunctions and eigenvalues of the matrix $\mathbf{A}_\mathbf{y}$.

The coefficients b_i of the vector $\mathbf{b}$ can be determined from the initial data. For example, at $t = t_0$ we have $\mathbf{y}(t_0) = \mathbf{y}_0$ so that (3.6) gives $\mathbf{b} = \boldsymbol{\Psi}^\dagger \mathbf{y}_0$, where $\boldsymbol{\Psi}$ is a matrix comprising the DMD modes $\boldsymbol{\psi}_i$. The ultimate goal in the DMD algorithm is to optimally construct the matrix $\mathbf{A}_\mathbf{y}$ so that the true and approximate solutions remain optimally close in a least-square sense, i.e., $\|\mathbf{y}(t) - \tilde{\mathbf{y}}(t)\| \ll 1$. Of course, the optimality of the approximation holds only over the sampling window where $\mathbf{A}_\mathbf{y}$ is constructed, but the approximate solution can be used not only to make future state predictions but also to decompose the dynamics into various time-scales since the ω_k are prescribed. Moreover, the DMD typically makes use of low-rank structure so that the total number of modes, $k \ll n$, allows for dimensionality reduction of the dynamical system.

In effect, the least-square regression of the nonlinear dynamical system to the linear system (3.5) allows us to approximate the governing equation (2.1) in the following manner:

$$\mathbf{M}\dot{\mathbf{y}}(t) = \mathbf{A}\mathbf{y}(t) + \mathbf{f}(t, \mathbf{y}(t)) \approx \mathbf{A}\mathbf{y}(t) + \mathbf{A}_{\mathbf{y}}\mathbf{y}(t), \tag{3.7}$$

where the DMD algorithm constructs the matrix $\mathbf{A}_{\mathbf{y}}$ approximating the nonlinearity over the snapshots collected.

In practice, the matrix $\mathbf{A}_{\mathbf{y}}$ is, in general, highly ill-conditioned and when the state dimension n is large, the aforementioned matrix may be even intractable to analyze directly. Instead, DMD circumvents the eigendecomposition of $\mathbf{A}_{\mathbf{y}}$ by considering a rank-reduced representation in terms of a POD-projected matrix $\tilde{\mathbf{A}}_{\mathbf{y}}$. The DMD algorithm proceeds as follows [36]:

1. First, take the SVD of $\mathbf{Y} \in \mathbb{C}^{n\times m}$:

   $$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{3.8}$$

   where $*$ denotes the conjugate transpose, $\mathbf{U} \in \mathbb{C}^{n\times k}$, $\mathbf{\Sigma} \in \mathbb{C}^{k\times k}$, and $\mathbf{V} \in \mathbb{C}^{m\times k}$. Here k is the rank of the reduced SVD approximation to $\mathbf{Y}$. The left singular vectors $\mathbf{U}$ are POD modes.

   The SVD reduction in (3.8) could also be exploited at this stage in the algorithm to perform a low-rank truncation of the data. Specifically, if a low-dimensional structure is present in the data, the singular values of $\mathbf{\Sigma}$ will decrease sharply to zero with perhaps only a limited number of dominant modes. A principled way to truncate noisy data would be to use the recent hard-thresholding algorithm of Gavish and Donoho [12].
2. Next, compute $\tilde{\mathbf{A}}_{\mathbf{y}}$, the $k \times k$ projection of the full matrix $\mathbf{A}_{\mathbf{y}}$ onto POD modes:

   $$\begin{aligned} &\mathbf{A}_{\mathbf{y}} = \mathbf{Y}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^* \\ \Longrightarrow\quad &\tilde{\mathbf{A}}_{\mathbf{y}} = \mathbf{U}^*\mathbf{A}_{\mathbf{y}}\mathbf{U} = \mathbf{U}^*\mathbf{Y}'\mathbf{V}\mathbf{\Sigma}^{-1}. \end{aligned} \tag{3.9}$$

3. Compute the eigendecomposition of $\tilde{\mathbf{A}}_{\mathbf{y}}$:

   $$\tilde{\mathbf{A}}_{\mathbf{y}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}, \tag{3.10}$$

   where columns of $\mathbf{W}$ are eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix containing the corresponding eigenvalues λ_i.
4. Finally, we may reconstruct eigendecomposition of $\mathbf{A}_{\mathbf{y}}$ from $\mathbf{W}$ and $\mathbf{\Lambda}$. In particular, the eigenvalues of $\mathbf{A}_{\mathbf{y}}$ are given by $\mathbf{\Lambda}$ and the eigenvectors of $\mathbf{A}_{\mathbf{y}}$ (DMD modes) are given by columns of $\mathbf{\Psi}$:

   $$\mathbf{\Psi} = \mathbf{Y}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{W}. \tag{3.11}$$

Note that (3.11) from [36] differs from the formula $\mathbf{\Psi} = \mathbf{U}\mathbf{W}$ from [34], although these will tend to converge if $\mathbf{Y}$ and $\mathbf{Y}'$ have the same column spaces. As a pseudo-algorithm, it can be summarized in Algorithm 1.

3.1. Applications of the DMD method. In this section, we propose two different applications of the DMD method to ROMs. Our first application concerns the interpolation of a parametrized function which is compared with the DEIM approach. The second one is related to the reduction of dynamical systems and considers the DMD method as a Galerkin projection strategy.

Algorithm 1. Exact DMD.

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

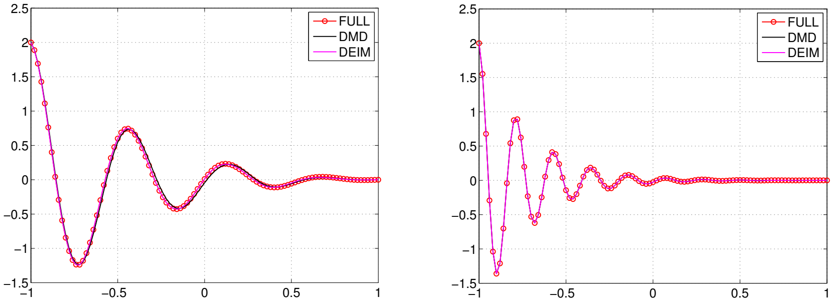

FIG. 3.1. *Test 1: DMD and DEIM interpolation with $\mu = 1.2$ (left) and $\mu = 3.1$ (right) for $k = 20$.*

Test 1: Interpolation of parametrized functions. Let us consider the following nonlinear parametrized functions:

$$s(x;\mu) = (1-x)\cos(3\pi\mu(x+1))e^{-(1+x)\mu}, \tag{3.12}$$

where $s : \Omega \times \mathcal{D} \to \mathbb{R}$, $x \in \Omega = [-1, 1]$ and $\mu \in \mathcal{D} = [1, \pi]$. This nonlinear function is from [9]. Let us discretize the space domain $[x_1, \dots, x_n] \in \mathbb{R}^n$ with x_i equidistant in Ω. With compact notation we define $\mathbf{f} : \mathcal{D} \to \mathbb{R}^\mathbf{n}$ by

$$\mathbf{f}(\mu) = [s(x_1;\mu), \dots, s(x_n;\mu)] \in \mathbb{R}^n \tag{3.13}$$

for $\mu \in \mathcal{D}$. This example uses 51 snapshots of $\mathbf{f}(\mu_j)$ to build the POD basis functions where μ_i are equidistributed points in $[1, \pi]$ and $n = 101$.

The purpose of this subsection is to show that the DMD might also be used as an interpolation method as shown in Figure 3.1. As we can see DMD is able to reconstruct the parametrized functions in $\mu = \{1.2, 3.1\}$ which is not included in the snapshot set. If we look more closely into these approximations and compare it with the DEIM interpolation method we can see that the DMD method is always much faster than DEIM (Figure 3.2, left) and the error, at the beginning, is comparable up to the first 20 modes. The error is computed with respect to the Frobenius norm.

Test 2: DMD-Galerkin approximation. Although DMD is a well-known equation-free method, it also works in a Galerkin projection framework. In this subsection, we compare the performances of POD and DMD integrated with the Galerkin method. We note that DMD basis functions in the DMD-Galerkin projection are computed following Algorithm 1 and then orthonormalized. Let us consider the following one-dimensional linear advection equation:

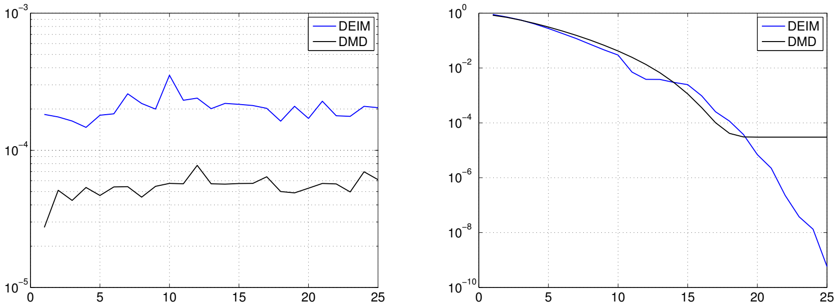

FIG. 3.2. *Test* 1*: CPU time (left) and relative error (right) for* $k = 1, \dots, 25$.

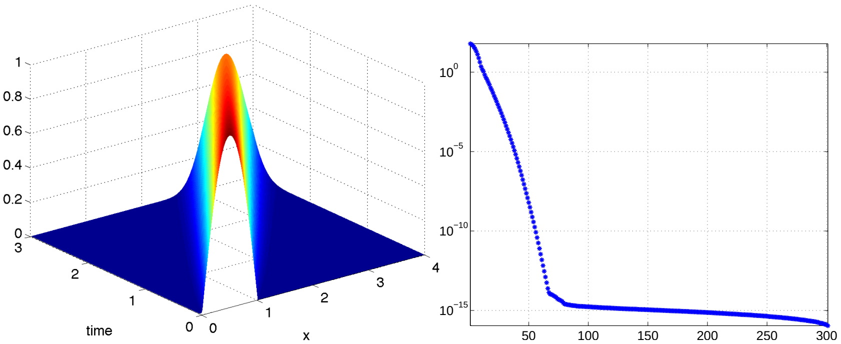

FIG. 3.3. *Test* 2*: Solution of* (3.14) *(left) and singular values (right).*

$$\begin{aligned} y_t(x,t) + \theta y_x(x,t) &= 0, && (x,t) \in [a,b] \times [0,T], \\ y(x,0) &= y_0(x), && x \in [a,b], \\ y(a,t) = 0 &= y(b,t), && t \in [0,T], \end{aligned} \tag{3.14}$$

where $a = 0, b = 4, T = 3, \theta = 1, y_0(x) = \sin(\pi x)$ if $0 \leq x \leq 1$ and 0 elsewhere. In order to lead (3.14) to our general formulation (2.1) we utilize a finite difference discretization with a spatial step $\Delta x = 0.01$. The dimension of the problem is $n = 399$. We note that in this case the mass matrix $\mathbf{M}$ is the identity matrix. In order to apply POD and DMD, we need to compute the snapshot set which is given by the temporal discretization of (3.14) with an implicit Euler scheme and a temporal step size $\Delta t = 0.01$. The solution of (3.14) builds the snapshot set and it is visualized on the left side of Figure 3.3. We also show the decay of the singular values of the snapshot set on the right of Figure 3.3.

POD-Galerkin has already been explained in section 2. The DMD-Galerkin approach assumes that our solution can be written as $\mathbf{y}(t) \approx \mathbf{\Psi}^{\text{DMD}}\mathbf{y}^{\text{DMD}}(t)$. This assumption is very similar to (2.5) but considers different basis functions. The reduced problem has the same form of (2.6). Figure 3.4 shows the results of model order reduction with POD (top), with DMD considered as a Galerkin projection method (middle), and DMD as an equation-free method. The first column refers to approximations of rank 5, the second of rank 10, and the third of rank 15. As expected, if we increase the rank of the basis functions, we can easily see that the approximation gets better and better. It is well known that advection problems have a high variability

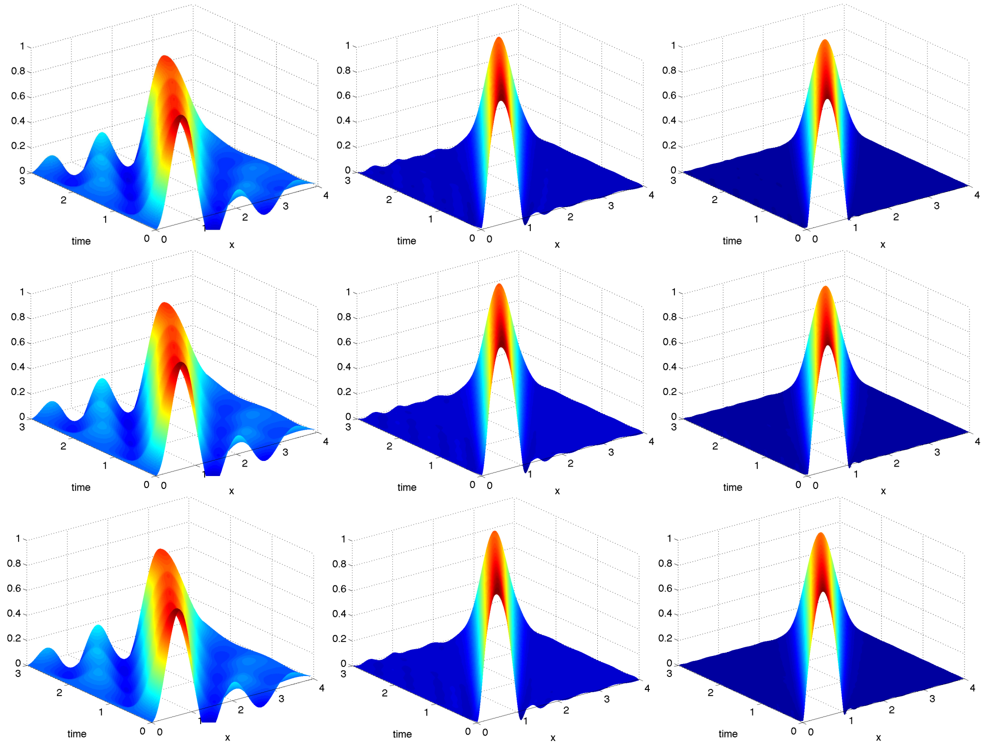

FIG. 3.4. *Test* 2: *Reduced approximation with* 5 *basis functions (first column),* 10 *basis functions (second column), and* 15 *basis functions (third column). POD approximation (top), DMD-Galerkin (middle), DMD (data-driven) (bottom).*

during time evolution and it is difficult to capture the dynamics with only a few basis functions.

The error analysis in the left panel of Figure 3.5 confirms our heuristic expectations. Here we compute the relative error with respect to the Frobenius norm where we consider as truth the solution of the governing equations approximated by a finite difference scheme. The error decays as soon as we increase the dimension of the reduced model, in particular, the POD method always performs better. On the other hand, it is, in general, hard to see a significant difference between the DMD-Galerkin and the DMD data-driven method as shown in Figure 3.4. This phenomena has been observed for both linear and nonlinear problems (see section 5). In fact, this error analysis brings us to the idea of working with a data-driven method for the approximation of nonlinear dynamical systems. For the sake of clarity, we also show the first two modes in Figure 3.5, where we can clearly observe that the DMD basis functions oscillate more than the POD modes.

4. Coupling POD and DMD for nonlinear problem.

This section focuses on the approximation of a nonlinear problem by means of model order reduction. As discussed in section 2, the use of POD basis functions does not lead to a surrogate model which is independent of the full dimension of the problem (see (2.6)) unless all the nonlinear terms are exapended into POD modes and their inner products computed. We advocate an alternative method to EIM/DEIM by working with the DMD algorithm for evaluating the nonlinear term in (2.1). As already discussed,

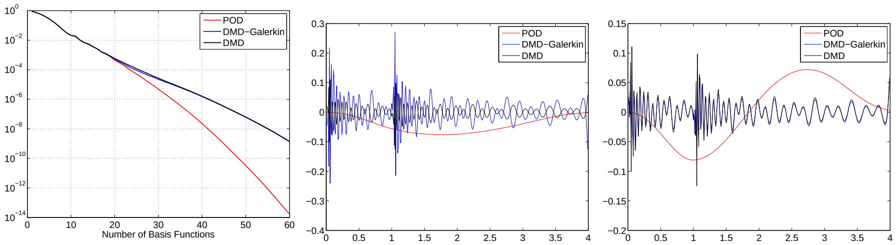

FIG. 3.5. *Error analysis with respect to the Frobenius norm (left), first mode (middle), and second mode (right).*

the snapshot measurements used in DMD approximate the dynamics and predict the future state. The use of DMD, in this work, concerns the approximate of the nonlinearity $\mathbf{f}(t,\mathbf{y}(t))$ of the dynamical system (2.1).

To begin with, let us collect snapshots from the system $\{\mathbf{y}(t_0),\dots,\mathbf{y}(t_m)\}$ for some given time instances $\{t_0,\dots,t_m\}$ and compute the POD basis functions of rank ℓ. Then, we need to collect snapshots for the nonlinearity $\{\mathbf{f}(t_0,\mathbf{y}(t_0)),\dots,\mathbf{f}(t_m,\mathbf{y}(t_m))\}$ and divide them into two different sets as explained in section 3. We apply the DMD algorithm (see Algorithm 1) to the nonlinear measurements. The DMD approximation of the nonlinearity reads

$$\mathbf{f}^{\text{DMD}}(t,y(t)) = \sum_{i=1}^{k} b_i \boldsymbol{\psi}_i^{\text{DMD}} \exp(\omega_i t)\,, \tag{4.1}$$

where $\boldsymbol{\psi}_i^{\text{DMD}}$ are the DMD basis functions of rank k related to the nonlinear function $\mathbf{f}(t,\mathbf{y}(t))$ and ω_i are the eigenvalues of the linear matrix $\tilde{A}_{\mathbf{y}}$. The weighting coefficients b_i for each DMD mode can be determined by projecting on the initial evaluation of the nonlinear function $\mathbf{f}^{\text{DMD}}(t_0,y(t_0))$ and taking the pseudoinverse. With compact notation we obtain

$$\mathbf{f}^{\text{DMD}}(t,y(t)) \approx \boldsymbol{\Psi}^{\text{DMD}} \operatorname{diag}(e^{\omega^{\text{DMD}} t})\mathbf{b}, \tag{4.2}$$

where $\mathbf{b} = (\boldsymbol{\Psi}^{\text{DMD}})^{\dagger}\mathbf{f}(t_0,\mathbf{y}(t_0)) \in \mathbb{R}^k$, $\operatorname{diag}(e^{\omega^{\text{DMD}} t})\mathbf{b} \in \mathbb{R}^k$ represents the reduced approximation of the data in terms of the DMD modes. As we can see from (4.2) the nonlinearity is approximated by a DMD representation and no further evaluation of the nonlinearity is required. This circumvents the DEIM selection of the interpolation points. If we plug the approximation of the nonlinearty (4.2) into the POD system (2.6) we get the following reduced system:

$$\begin{cases} \mathbf{M}^{\ell}\dot{\mathbf{y}}^{\ell}(t) = \mathbf{A}^{\ell}\mathbf{y}^{\ell}(t) + (\boldsymbol{\Psi}^{\text{POD}})^T \boldsymbol{\Psi}^{\text{DMD}} \operatorname{diag}(e^{\omega^{\text{DMD}} t})\mathbf{b}, \\ \mathbf{y}^{\ell}(0) = \mathbf{y_0}^{\ell}. \end{cases} \tag{4.3}$$

Let us analyze the dimension of the new reduced dynamical system (4.3). The matrix $\mathbf{M}^{\ell}, \mathbf{A}^{\ell} \in \mathbb{R}^{\ell\times\ell}$ has the same dimension as the POD system. The quantity $(\boldsymbol{\Psi}^{\text{POD}})^T \boldsymbol{\Psi}^{\text{DMD}} \in \mathbb{R}^{\ell\times k}$ is independent of the dimension of the full system, and $\operatorname{diag}(e^{\omega^{\text{DMD}} t})\mathbf{b} \in \mathbb{R}^k$. Even for this formulation we are able to build a surrogate model which does not depend on the dimension of the original system. Moreover, in this formulation we do not have to evaluate the nonlinearity further, which gives an important speedup in the efficiency of the formulation. We also note that the efficiency

Algorithm 2. POD-DMD.

```
Require: Snapshots $\{\mathbf{y}_(t_0),\dots,\mathbf{y}(t_m)\}$, $\ell$ number of POD modes, $k$ number of DMD modes
Compute the POD basis function $\{\boldsymbol{\psi}_i\}_{i=1}^{\ell}$ of rank of $\ell$
Compute nonlinear snapshots $\{\mathbf{f}(t_0,\mathbf{y}(t_0)),\dots,\mathbf{f}(t_m,\mathbf{y}(t_m))\}$
Set $\mathbf{Y} = [\mathbf{f}(t_0,\mathbf{y}(t_0)),\dots,\mathbf{f}(t_{m-1},\mathbf{y}(t_{m-1}))]$
Set $\mathbf{Y'} = [\mathbf{f}(t_1,\mathbf{y}(t_1)),\dots,\mathbf{f}(t_m,\mathbf{y}(t_m)]$
Compute DMD modes following Algorithm 1
Set and integrate (4.3)
Project back full solution
```

of the DEIM method is related to the nonlinear term since it is not always clear if the components of the nonlinear term are independent. As in the DEIM case some quantities can be precomputed offline. Of course, this method is closely related to the snapshot set and approximates the nonlinear term with a linear regression operator. The algorithm is summarized in Algorithm 2.

The POD-DMD method has one significant advantage: computational speed. Indeed, as we will show, the POD-DMD algorithm is significantly faster than the POD-DEIM method in approximating the nonlinear terms in the model reduction. Indeed, the computational efficiency for this task is improved by an order of magnitude or more. The drawback of the method is that the DMD modes produced for the low-rank projection are not orthogonal, which is in contrast to POD modes which give an optimal orthonormal basis with some guaranteed convergence properties [16]. We hope to fix this problem in future work by potentially orthogonalizing the DMD modes. However, reorthogonalization of the DMD modes is nontrivial. Specifically, one can easily use, for instance, Gram–Schimdt to orthogonalize the DMD modes of the matrix $\boldsymbol{\Psi}^{\text{DMD}}$. But the new orthogonal modes would no longer be associated with the DMD spectra ω_j. Thus the DMD approximation (4.2) no longer holds. To date, reorthogonalization remains an open theoretical challenge in the DMD community.

5. Numerical tests.

In this section we present our numerical tests. In our numerical computations we use the finite difference method to reduce a partial differential equation into the form (2.1) and integrate the system with a semi-implicit scheme. All the numerical simulations reported in this paper are performed on a MacBook Pro with an Intel Core i5, 2.2 GHz, and 8 GB RAM using MATLAB R2013a.

In the following numerical examples we build different surrogate models, such as POD, POD-DEIM, and POD-DMD, and compare their performances in terms of CPU time and the error with respect to a reference solution computed by the finite difference approach. For the sake of completeness we also utilize DMD as a data driven method. The tests consider three types of equations.

Test 3: Semilinear parabolic equation. Let us consider the following equation:

$$
\begin{aligned}
y_t(x,t) - \theta\Delta y(x,t) + \mu\left(y(x,t) - y^3(x,t)\right) &= 0, && (x,t)\in\Omega\times[0,T],\\
y(x,0) &= y_0(x), && x\in\Omega,\\
y(x,t) &= 0, && t\in\partial\Omega\times[0,T],
\end{aligned}
\tag{5.1}
$$

where Δ is the Laplace operator, $\Omega = [0,1]\times[0,1], T = 3, \theta = 0.1, \mu = 1, x = (x_1,x_2), y_0(x) = 0.1$ if $0.1 \leq x_1x_2 \leq 0.6$ and 0 elsewhere. The POD basis vectors are built upon 100 equidistant snapshots. The finite difference discretization yields a system of ODEs of the same form as (2.1) with $n = 10000$. The solution of this equation generates a stationary solution $y(x,t) \equiv 1$ for large t, as shown in Figure 5.1. Figure 5.2 shows a similar decay for the singular values of the snapshot set and of the nonlinear term in (5.1).

The complexity of problem (5.1) is reduced by model order reduction. When dealing with model order reduction, it is relevant to consider the CPU time of the simulation and the error. In general it is important to have a trade-off between the two quantities. Figure 5.3 considers the CPU time on the left. The DMD is always the fastest method since it is an equation-free method. Among the equation-based approaches, the POD-DMD approximation is computationally faster than any other approximation for any dimension of the reduced system. The strength of POD-DMD is the fact we do not have to evaluate the nonlinearity after collecting snapshots. We note that the number of POD, DEIM, and DMD are always the same in Figure 5.3, e.g., $\ell = k$. In the middle of Figure 5.3 we compute the relative error with respect to the Frobenius norm. It is clear that POD provides the best approximation. As expected, all the methods decrease their error when increasing the number of basis functions. A more appropriate comparison is given in the right panel of Figure 5.3. There we look at the CPU time for a given fixed error tolerance. As shown, the DMD method is faster than the POD-DMD approach.

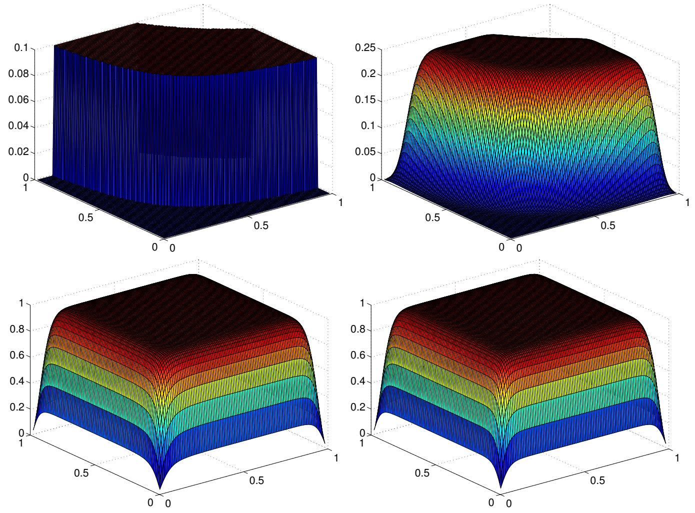

Fig. 5.1. *Test* 3*: Solution of* (5.1) *at time* $t = \{0, 0.1\}$ *(top) and* $t = \{1.5, 3\}$ *(bottom).*

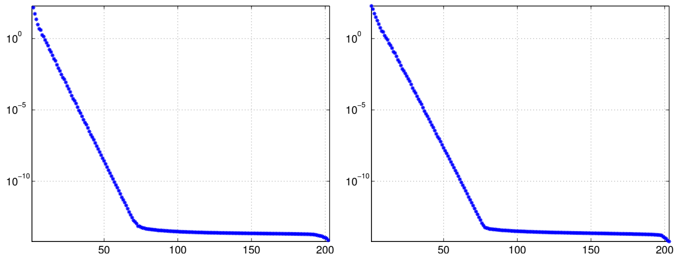

Fig. 5.2. *Test* 3*: Singular values of the solution (left) and the nonlinearity (right) of* (5.1).

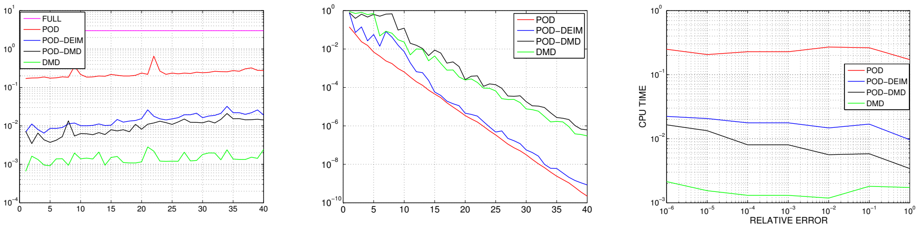

Fig. 5.3. *Test* 3: *CPU time (left) and relative error in the Frobenius norm where the number of POD modes and DEIM/DMD points are the same* $\ell = k = \{1, \ldots, 40\}$ *(middle); CPU time for a given error (right).*

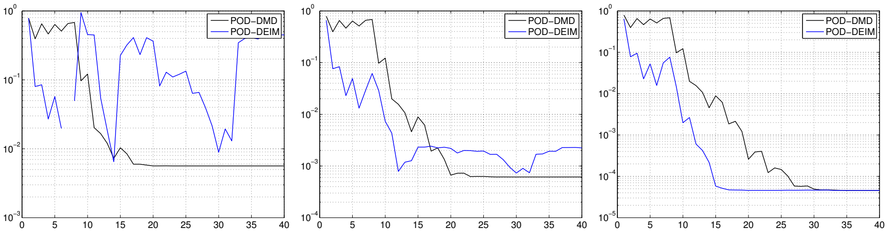

Fig. 5.4. *Test* 3: *Relative error for* 5 *POD basis functions (left),* 10 *POD basis (middle), and* 15 *POD basis (right). The x-axis show the number of DEIM/DMD modes utilized* $k = 1, \ldots, 40$.

Since POD-DMD is faster than other methods it is natural to look at the performance with a different number of basis functions. Figure 5.4 shows the error for a fixed number of basis functions $\ell = \{5, 10, 15\}$ and $k \in [1, 40]$. As we can see POD-DMD performs better than POD-DEIM when $\ell = 5, 10$. In this case increasing the number of DMD basis functions lead to more accurate solutions of POD-DEIM. Moreover, we can observe a monotone decay of the error for the POD-DMD approach.

Test 4: Burgers' equation. Let us consider the following one-dimensional Burgers equation:

$$\begin{aligned} y_t(x,t) - \theta y_{xx}(x,t) + y(x,t) y_x(x,t) = 0, &\qquad (x,t) \in [a,b] \times [0,T], \\ y(x,0) = y_0(x), &\qquad x \in [a,b], \\ y(a,t) = 0 = y(b,t), &\qquad t \in [0,T], \end{aligned} \tag{5.2}$$

where $a = 0, b = 1, T = 1, \theta = 0.01, y_0(x) = \operatorname{sgn}(x)$. The dimension of the semi-discrete problem is $n = 101$ in (2.1). In Figure 5.5 we visualize the full solution of (5.2) and the decay of the singular values. We note that the decay is very similar for y and its nonlinearity. The results of the model reduction are shown in Figure 5.6. In the left panel, the POD approximation with 20 basis function is demonstrated. POD-DEIM is visualized in the middle panel, and finally, the POD-DMD approach is in the right panel. With a 20-rank truncation of the DEIM and DMD approximation, it is difficult to see any differences in the solution.

The CPU time is expressed on the left side of Figure 5.7 and we can see, as already discussed in the previous example, that the CPU time of POD-DMD is always below the other two approximations but we lose some accuracy, as expected. The relative error is represented in the middle of Figure 5.7, where one can see that it decays when

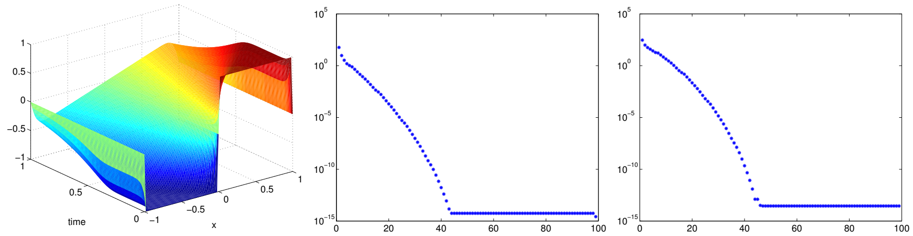

FIG. 5.5. *Test* 4*: Full approximation (left), and singular values of the solution (middle) and of the nonlinearity (right) of* (5.2)*.*

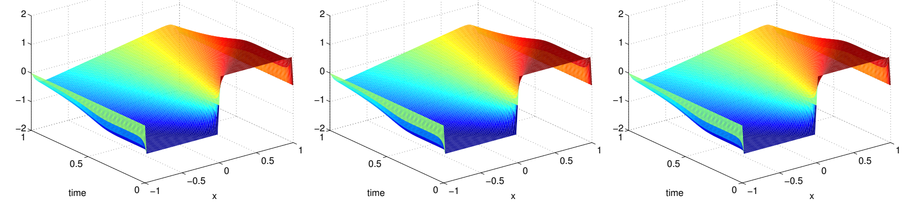

FIG. 5.6. *Test* 4*: Approximation with* 20 *POD basis functions (left), approximation with* 20 *POD basis functions and* 20 *DEIM (middle), and approximation with* 20 *POD basis functions and* 20 *DMD (right).*

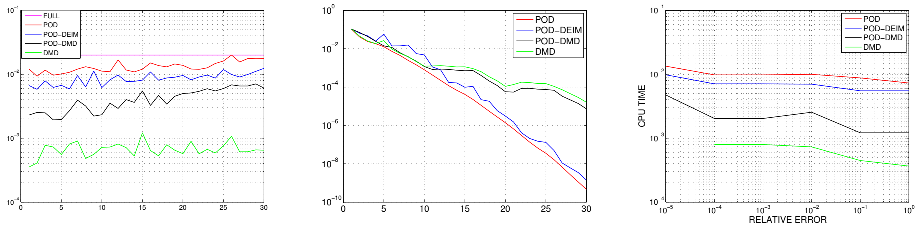

FIG. 5.7. *Test* 4*: CPU time (left) and relative error in the Frobenius norm where the number of POD modes and DEIM/DMD points are the same with* $\ell = k = \{1, \dots, 30\}$ *(middle); CPU time for a given error (right).*

we increase the dimension of the surrogate model. We also show the DMD as a data-driven method which turns out to be faster than the other with a similar accuracy to POD-DMD. In the right panel of Figure 5.7 we show the CPU time for a given error tolerance. It turns out that DMD and POD-DMD are faster than the other methods.

It is difficult to directly compare POD-DMD and POD-DEIM since the meaning of the rank in DEIM is different from DMD. For this reason we also compute the error varying the rank k for a fixed number of POD basis functions. Figure 5.8 shows that POD-DMD always has a monotone decay. POD-DEIM is in general more accurate, especially for large k. Since POD-DMD is always faster, this is not a big issue, and in fact, one could work with a low-dimensional structure of the POD basis functions and consider a larger number of DMD basis functions as shown in Figure 5.7 in the right panel.

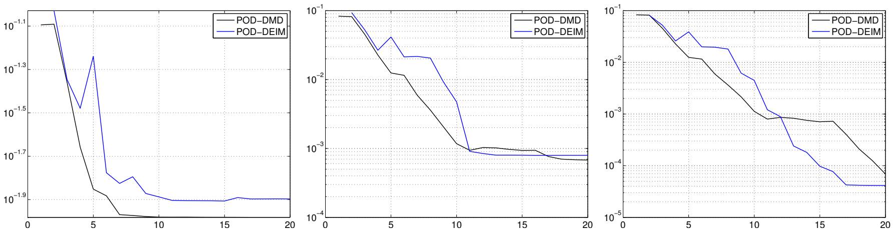

FIG. 5.8. *Test* 4*: Relative error for* 5 *POD basis functions (left),* 10 *POD basis functions (middle), and* 15 *POD basis functions (right). The x-axis show the number of DEIM/DMD modes utilized* $k = 1, \dots, 30$.

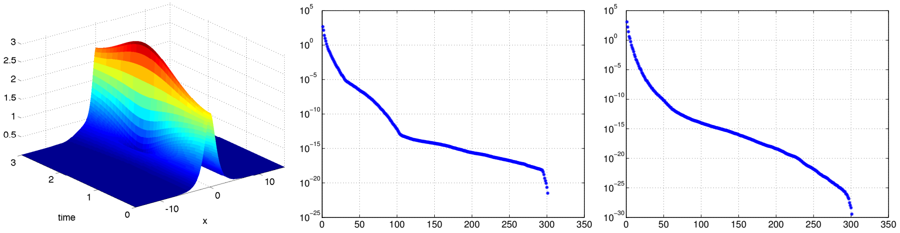

FIG. 5.9. *Test* 5*: Full approximation (left), and singular values of the solution (middle) and of the nonlinearity (right) of* (5.3).

Test 5: Nonlinear Schrödinger equation. Let us consider the following Schrödinger equation:

$$
\begin{aligned}
y_t - i\theta y_{xx} - i\theta |y|^2 y &= 0, && (x,t) \in [-L, L] \times [0, T], \\
y(x,0) &= y_0(x), && x \in [-L, L], \\
y(-L,t) = 0 &= y(L,t), && t \in [0, T],
\end{aligned}
\tag{5.3}
$$

where $L = 15, T = 3, \theta = 0.5$, and $y_0(x) = 2sech(x)$. The dimension of the correspondent semidiscrete problem (2.1) is $n = 301$. The solution of (5.3) is shown in Figure 5.9 on the left. The singular values of the solution are shown in the middle panel, whereas the singular values of the nonlinearity are in the right panel. It is well known that Schrödinger's equation generates wave functions in its solution and therefore it is difficult to capture this behavior with only a few modes.

Approximation by means of the model order reduction technique is shown in Figure 5.10; we can see it is hard to distinguish any difference between the reduced solutions. In this case POD-DEIM always performs better than POD-DMD but is more expensive. Again, we emphasize the speedup of the DMD and POD-DMD methods with respect to the other methods and performance of these methods with a fixed error tolerance (see Figure 5.11.) Finally, we show in Figure 5.12 the error decay with a fixed number of POD basis functions and varying the rank k for the nonlinear term.

6. Conclusions and future work.

In order to make model reduction methods such as POD computationally efficient, innovative methods for evaluating the nonlinear terms of the governing equations (2.1) must be used. Previous successful techniques used sparse sampling to evaluate the nonlinearity. Indeed, the discrete empirical interpolation method identifies through a greedy algorithm a limited number

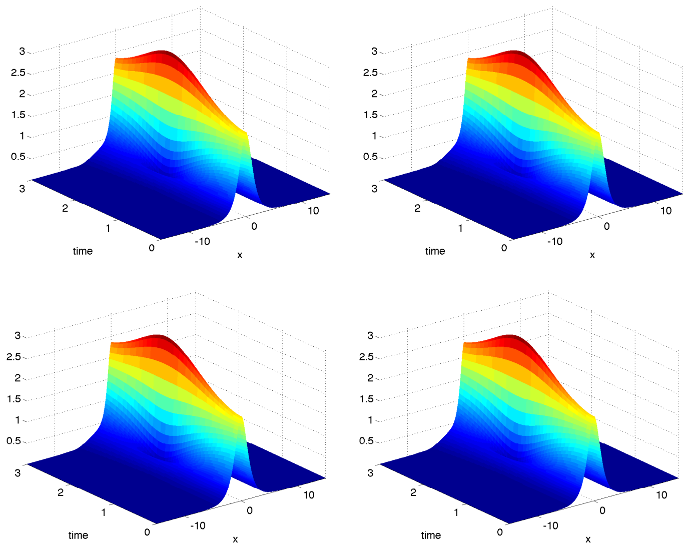

Fig. 5.10. *Test* 5*: Approximation with* 20 *POD basis functions (top left), approximation with* 20 *POD basis functions and* 10 *DEIM (top right), approximation with* 20 *POD basis functions and* 20 *DMD (bottom left), and approximation with* 20 *DMD basis functions as data-driven method (bottom right).*

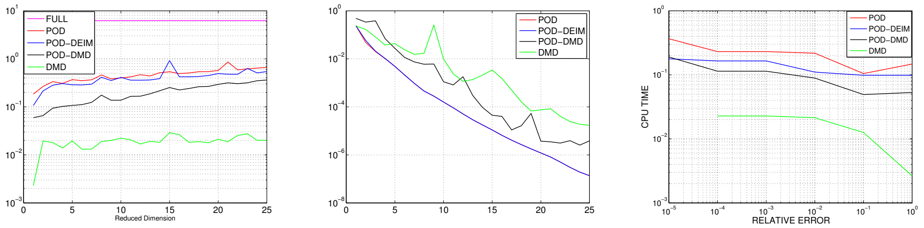

Fig. 5.11. *Test* 5*: CPU time (left) and relative error in the Frobenius norm. Number of POD modes and DEIM/DMD points are the same with* $\ell = k = \{1, \dots, 25\}$ *(middle), CPU time for a given error (right).*

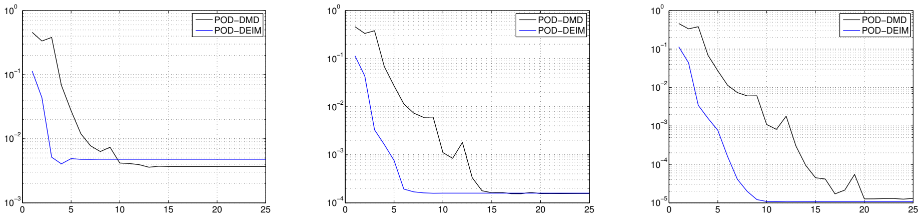

Fig. 5.12. *Test* 5*: Relative error for* 5 *POD basis functions (left),* 10 *POD basis functions (middle), and* 15 *POD basis functions (right) for* $k = \{1, \dots, 25\}$.

of spatial sampling locations that can allow for reconstruction of the nonlinear terms in a low-dimensional manner. Such sparse sampling of the nonlinearity is directly related to compressive sensing strategies whereby a small number of sensors can be used to characterize the dynamics of the high-dimensional nonlinear system. In this paper we present a new model order reduction approach for nonlinear dynamical systems. The method couples the POD method for the projection of the system and the DMD algorithm for the approximation of the nonlinear term. In particular, DMD provides a significant reduction of the system in terms of the CPU time since the nonlinearity is never evaluated online. The method is effective for nonlinear dynamical systems where POD approximations are relevant.

The DMD method is a fairly robust technique, but it can fail when the data sampled exhibit strong transient growth and decay phenomena and/or intermittency. Of course, such scenarios can also be problematic for POD methods in general as they rely on correlations in the sampled data for producing POD modes. Thus, generally speaking, the POD-DMD method advocated should be robust in dynamical systems where standard POD-DEIM methods are applied to take advantage of low-dimensional dynamics. Future work will focus on error estimation of the proposed method in both the DMD and POD-DMD projection techniques. Special focus will be given to improving the error estimates. We also intend to use recent innovations in the DMD method, specifically around multiresolution analysis [18] and compression [6, 1], to more effectively construct approximations to the nonlinear dynamics.

Acknowledgments. The second author would like to thank Steven Brunton, Joshua Proctor, Bing Brunton, Matthew Williams, Jonathan Tu, and Clancy Rowley for invaluable discussions related to the dynamic mode decomposition and Koopman operator theory.

REFERENCES

[1] A. Alla and J. N. Kutz, *Randomized Model Order Reduction*, https://arxiv.org/pdf/1611.02316.pdf, 2016.

[2] P. Astrid, *Fast reduced order modeling technique for large scale LTV systems*, in Proceedings of the 2004 American Control Conference, Vol. 1, 2004, pp. 762–767.

[3] 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*, C. R. Math., 339 (2004), pp. 667–672.

[4] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531.

[5] D. A. Bistrian and I. M. Navon, *An improved algorithm for the shallow water equations model reduction: Dynamic mode decomposition vs POD*, Internat. J. Numer. Methods Fluids, 78 (2015), pp. 552–580.

[6] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Compressive sampling and dynamic mode decomposition*, J. Comput. Dynam., 2 (2015), pp. 165–191.

[7] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, *The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows*, J. Comput. Phys., 242 (2013), pp. 623–647.

[8] A. Cammilleri, F. Guéniat, J. Carlier, L. Pastur, E. Mémin, F. Lusseyran, and G. Artana, *POD-spectral decomposition for fluid flow analysis and model reduction*, Theoret. Comput. Fluid Dyn., 27 (2013), pp. 787–815.

[9] S. Chaturantabut and D. Sorensen, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764.

[10] Z. Drmac and S. Gugercin, *A new selection operator for the discrete empirical interpolation method: Improved a priori error bound and extensions*, SIAM J. Sci. Comput., 38 (2016), pp. A631–A648.

[11] R. Everson and L. Sirovich, *Karhunen-Loéve procedure for gappy data*, J. Opt. Soc. Amer. A, 12 (1995), pp. 1657–1664.

[12] M. Gavish and D. L. Donoho, *The optimal hard threshold for singular values is* $4/\sqrt{3}$, IEEE Trans. Inform. Theory, 60 (2014), pp. 5040–5053.

[13] P. J. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, 2nd ed., Cambridge Monogr. Mech., Cambridge University Press, Cambridge, England, 2012.

[14] H. Hotelling, *Analysis of a complex of statistical variables into principal components*, J. Educ. Psychol., 24 (1933), pp. 417–441.

[15] B. O. Koopman, *Hamiltonian systems and transformation in Hilbert space*, Proc. Natl. Acad. Sci. USA, 17 (1931), pp. 315–318.

[16] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for parabolic problems*, Numer. Math., 90 (2001), pp. 117–148.

[17] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics*, SIAM J. Numer. Anal., 40 (2002), pp. 492–515.

[18] J. N. Kutz, X. Fu, and S. L. Brunton, *Multi-resolution dynamic mode decomposition*, SIAM J. Appl. Dyn. Syst., 15 (2016), pp. 713–735.

[19] J. N. Kutz, *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*, Oxford University Press, New York, 2013.

[20] E. N. Lorenz, *Empirical Orthogonal Functions and Statistical Weather Prediction*, Technical report, Massachusetts Institute of Technology, Cambridge, MA, 1956.

[21] J. L. Lumley, *Stochastic Tools in Turbulence*, Academic Press, New York, 1970.

[22] I. Mezić and A. Banaszuk, *Comparison of systems with complex behavior*, Phys. D, 197 (2004), pp. 101–133.

[23] I. Mezić, *Spectral properties of dynamical systems, model reduction and decompositions*, Nonlinear Dynam., 41 (2005), pp. 309–325.

[24] I. Mezić, *Analysis of fluid flows via spectral properties of the Koopman operator*, Ann. Rev. Fluid Mech. 45 (2013), pp. 357–378.

[25] J. N. Kutz, J. L. Proctor, and S. L. Brunton, *Koopman Theory for Partial Differential Equations*, preprint, arXiv:1607.07076, 2016.

[26] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, SIAM, Philadelphia, 2016.

[27] N. C. Nguyen, A. T. Patera, and J. Peraire, *A "best points" interpolation method for efficient approximation of parametrized functions*, Internat. J. Numer. Methods Engrg., 73 (2008), pp. 521–543.

[28] K. Pearson, *On lines and planes of closest fit to systems of points in space*, Philos. Mag., 2 (1901), pp. 559–572.

[29] B. Peherstorfer, D. Butnaru, K. Willcox, and H. J. Bungartz, *Localized discrete empirical interpolation method*, SIAM J. Sci. Comput., 36 (2014), pp. 168–192.

[30] B. Peherstorfer and K. Willcox, *Online adaptive model reduction for nonlinear systems via low-rank updates*, SIAM J. Sci. Comput., 37 (2015), pp. 2123–2150.

[31] A. Quarteroni and G. Rozza, eds., *Reduced Order Methods for Modeling and Computational Reduction*, Springer, New York, 2014.

[32] C. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson, *Spectral analysis of nonlinear flows*, J. Fluid Mech., 641 (2009), pp. 115–127.

[33] P. J. Schmid and J. Sesterhenn, *Dynamic mode decomposition of numerical and experimental data*, in Proceedings of the 61st Annual Meeting of the APS Division of Fluid Dynamics, American Physical Society, 2008.

[34] P. Schmid, *Dynamic mode decomposition of numerical and experimental data*, J. Fluid Mech., 656 (2010), pp. 5–28.

[35] L. Sirovich, *Turbulence and the dynamics of coherent structures.* Quart. Appl. Math., 45 (1987), pp. 561–590.

[36] J. Tu, C. Rowley, D. Luchtenburg, S. Brunton, and J. N. Kutz, *On dynamic mode decomposition: Theory and applications*, J. Comput. Dyn., 1 (2014), pp. 391–421.

[37] S. Volkwein, *Model Reduction Using Proper Orthogonal Decomposition*, Lecure Notes, University of Konstanz, 2013.

[38] K. Willcox, *Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition*, Comput. Fluids, 35 (2006), pp. 208–226.

[39] B. Yildirim, C. Chryssostomidis, and G. E. Karniadakis, *Efficient sensor placement for ocean measurements using low-dimensional concepts*, Ocean Modeling, 273 (2009), pp. 160–173.