Steven L. Brunton and J. Nathan Kutz

# 7 Data-driven methods for reduced-order modeling

**Abstract:** Data-driven mathematical methods are increasingly important for characterizing complex systems across the physical, engineering, and biological sciences. These methods aim to discover and exploit a relatively small subset of the full high-dimensional state space where low-dimensional models can be used to describe the evolution of the system. Emerging dimensionality reduction methods, such as the *dynamic mode decomposition* (DMD) and its Koopman generalization, have garnered attention due to the fact that they can (i) discover low-rank spatio-temporal patterns of activity, (ii) embed the dynamics in the subspace in an equation-free manner (i. e., the governing equations are unknown), unlike Galerkin projection onto proper orthogonal decomposition modes, and (iii) provide approximations in terms of linear dynamical systems, which are amenable to simple analysis techniques. The selection of observables (features) for the DMD/Koopman architecture can yield accurate low-dimensional embeddings for nonlinear partial differential equations (PDEs) while limiting computational costs. Indeed, a good choice of observables, including time delay embeddings, can often *linearize* the nonlinear manifold by making the spatio-temporal dynamics weakly nonlinear. In addition to DMD/Koopman decompositions, coarse-grained models for spatio-temporal systems can also be discovered using the *sparse identification of nonlinear dynamics* (SINDy) algorithm which allows one to construct reduced-order models in low-dimensional embeddings. These methods can be used in a nonintrusive, equation-free manner for improved computational performance on parametric PDE systems.

**Keywords:** reduced-order model, dynamic mode decomposition, Koopman theory, nonlinear system identification, sparse regression, system identification

**PACS:** 02.30.Hq, 02.30.Jr, 02.60.Cb, 02.30.Mv

## 7.1 Introduction

Data-driven modeling of complex systems is of increasing importance in modern scientific applications given the unprecedented rise of data collection on multi-scale, spatio-temporal systems. Enabled by emerging sensor technologies and high-performance computing platforms, the large-scale monitoring and collection of data

**Steven L. Brunton,** Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA
**J. Nathan Kutz,** Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA

on such systems has shifted our modeling paradigm by exploiting data-driven, machine learning approaches. Specifically, instead of positing empirical or approximate spatio-temporal models, typically characterized by partial differential equations (PDEs), the low-dimensional features extracted from time snapshots of the data can be directly used to construct *reduced-order models* (ROMs) for a variety of important tasks such as state-space reconstruction and diagnostics, as well as future state prediction and forecasting [14]. In this chapter, we present a diverse set of data-driven methods that can be used to construct ROMs directly from data. The methods presented can be used with traditional ROM architectures where the governing PDEs are known, or they can be used to discover unknown spatio-temporal dynamics directly from the data. Most of the methods are nonintrusive, minimizing the need for prohibitively expensive high-performance simulations. This also allows for accurate, low-fidelity models, enabling inexpensive Monte Carlo simulations. We present four methods for enabling data-driven ROMs: The *dynamic mode decomposition* (DMD) and the associated Koopman decomposition [78], the *sparse identification of nonlinear dynamics* (SINDy) algorithm [25], and the *Hankel alternative view of Koopman* (HAVOK) algorithm [22]. Each method can be used to advantage in a variety of situations, including when the governing PDE equations are known, only partially known, or unknown.

Consider a governing system of nonlinear PDEs of a single spatial variable $x$, which can be modeled as [14]

$$\mathbf{u}_t = \mathbf{L}(x)\mathbf{u} + \mathbf{N}(\mathbf{u}, x, t), \tag{7.1}$$

where $\mathbf{L}(x)$ is a linear operator and $\mathbf{N}(\cdot)$ prescribes the nonlinear terms in the evolution dynamics. Both $\mathbf{L}(x)$ and $\mathbf{N}(\cdot)$ may be unknown, or only partially known. As an example, the Burgers equation, $u_t = u_{xx} + vuu_x$, has $L = \partial^2/\partial x^2$ and $N(u) = vuu_x$. Associated with (7.1) are a set of initial and boundary conditions on a domain $x \in \mathcal{D}$. Historically, a number of analytic solution techniques have been devised to study (7.1) provided the right-hand side is known. Typically the aim of such methods is to reduce the PDE (7.1) to a set of ordinary differential equations (ODEs). The standard PDE methods of *separation of variables* and *similarity solutions* are constructed for this express purpose. Once in the form of an ODE, a broader variety of analytic methods can be applied along with a *qualitative theory* in the case of nonlinear behavior [61]. This again highlights the role that *asymptotics* can play in characterizing behavior.

For the general form of (7.1) where the right-hand side is known, separation of variables can often be used to yield a computational algorithm capable of producing low-rank approximations. Since the spatial solutions are not known a priori, it is typical to assume a set of basis modes which can be used for the low-rank approximation. Indeed, such assumptions on basis modes underly the critical ideas of the method of *eigenfunction expansions*. This yields a separation of variables solution ansatz of the

form

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{\Psi}(\mathbf{x})\mathbf{a}(t) = \sum_{k=1}^{r} \psi_k(\mathbf{x})a_k(t), \tag{7.2}$$

where $\mathbf{\Psi}(\mathbf{x}) \in \mathbb{C}^{n \times r}$ form a set of $r$ orthonormal basis modes and $\mathbf{x} \in \mathbb{R}^{n \times 1}$ represents the spatial discretization of $x$ in the governing PDE. The modal basis $\mathbf{\Psi}$ is often obtained via *proper orthogonal decomposition* (POD) [60, 14, 129]. This separation of variables solution approximates the true solution, provided $r$ is large enough. A fundamental assumption of reduced-order modeling is that there exists a low-rank truncation, or subspace, that accurately characterizes the evolution of the spatio-temporal system. More broadly, such approximations are based upon modal methods for building ROMs as discussed in Chapters 1 and 4 of Volume 1 of *Model order reduction* [12].

The orthogonality properties of the basis functions $\psi_k(x)$, which are the columns of $\mathbf{\Psi}$, enable us to make use of (7.2). Inserting the expansion (7.2) into the governing equations gives [14]

$$\frac{d\mathbf{a}}{dt} = \mathbf{\Psi}^T \mathbf{L} \mathbf{\Psi} \mathbf{a} + \mathbf{\Psi}^T \mathbf{N}(\mathbf{\Psi} \mathbf{a}). \tag{7.3}$$

The given form of $\mathbf{N}(\cdot)$ determines the mode-coupling that occurs between the various $r$ modes. Indeed, the hallmark feature of nonlinearity is the production of modal mixing from (7.3). Equation (7.3) is the canonical ROM identified as a Galerkin projection of the dynamics onto POD modes. It can be evaluated given full knowledge of the right-hand side of the governing PDE.

Equation (7.3) details how a low-rank subspace can be used to construct a Galerkin-POD-ROM model as a proxy, or surrogate, model for the high-fidelity model. In this reduction, the linear operator $\mathbf{\Psi}^T \mathbf{L} \mathbf{\Psi}$ can be computed once to produce an $r \times r$ matrix modeling the effects of the linear portion of the dynamics. What is more problematic is the evaluation of the nonlinear contribution $\mathbf{\Psi}^T \mathbf{N}(\mathbf{\Psi} \mathbf{a})$ in (7.3). Indeed, one of the primary challenges in producing the low-rank dynamical system is efficiently projecting the nonlinearity of the governing PDEs on the POD basis. This fact was recognized early on in the ROM community, and methods such as gappy POD [50, 142, 150] were proposed to more efficiently enable this hyperreduction task. More recently, the *empirical interpolation method* (EIM) [11], the *discrete EIM* (DEIM) [37], and the QR decomposition-based Q-DEIM [46], have provided a computationally efficient method for discretely (sparsely) sampling and evaluating the nonlinearity. These widely used hyperreduction methods ensure that the computational complexity of ROMs scale favorably with the rank of the approximation, even for complex nonlinearities.

Numerical schemes based on the Galerkin projection (7.3) are commonly used to perform simulations of the full governing system (7.1). Convergence to the true solution can be accomplished by judicious choice of both the modal basis elements $\mathbf{\Psi}$ and the total number of modes $r$. Interestingly, the separation of variables strategy, which is

rooted in *linear* PDEs, works for *nonlinear* and *nonconstant coefficient* PDEs, provided enough modal basis functions are chosen in order to capture the nonlinear mode mixing that occurs in (7.3). A good choice of modal basis elements allows for a smaller set of $r$ modes to be chosen to achieve a desired accuracy. The POD method is designed to specifically address the data-driven selection of a set of basis modes that are tailored to the particular dynamics, geometry, and parameters.

Unfortunately, the Galerkin-POD projection of the dynamics (7.3) is often unstable [34], requiring modification to stabilize the time-stepping scheme [4]. Moreover, the evaluation in (7.3) of the nonlinear term $\mathbf{\Psi}^T \mathbf{N}(\mathbf{\Psi a})$ renders the ROM scheme intrusive, i. e., to compute the nonlinear contribution in the low-rank subspace requires an expensive sampling of the high-fidelity model in order to build the low-rank subspace. Instead, one can directly approximate the nonlinearity via DMD which directly computes this contribution via nonintrusive methods [2]. Thus there is no recourse to high-fidelity and expensive computations to construct an approximation to the nonlinear contribution. If latent variables are present, i. e., important portions of the state-space have not been measured, then the Hankel alternative view of Koopman (HAVOK) algorithm, which helps to discover a proxy for the latent variable space, can be used instead of DMD. Finally, if the right-hand side is unknown, then the SINDy algorithm can be used to discover a low-rank, nonlinear model characterizing the evolution. The diversity of mathematical techniques highlights the emerging use of regression and machine learning strategies that can help model complex, spatio-temporal systems.

## 7.2 Data-driven reductions

Numerical linear algebra plays a central role in scientific computing [135, 77, 24]. Specifically, methods that have historically improved the efficiency of solving $\mathbf{Ax} = \mathbf{b}$ have always been of critical importance for tractable computations, especially at scale, where *scale* is a relative term associated with the limits of computing in a given era. From QR decomposition to lower-upper factorization [135], matrix decompositions have been the primary methods to enable improved computational efficiency. But perhaps the most important factorization technique is the *singular value decomposition* (SVD) [77], which plays a key role in data analysis and computation, including applications in reduced-order modeling through POD and DMD.

### 7.2.1 Singular value decomposition

The success of the SVD algorithm is largely due to the fact that by construction, it extracts the dominant, correlated features from any given data matrix. This often allows one to approximate the matrix with a principled low-rank approximation which

is guaranteed to be the best approximation in an $\ell_2$-sense. This mathematical architecture is so powerful and universal that it has been invented and used extensively in a wide range of fields [77]. Specifically, it is alternatively known as *principal component analysis* (PCA) in statistics (where to be precise, each column or row is scaled to have mean zero and unit variance), *POD* in the fluid dynamics community, *empirical mode decomposition* in atmospheric sciences, the *Hotelling transform*, *empirical eigenfunctions*, or *Karhunen-Loève decomposition*. Thus, from seemingly complex data from which a matrix is composed, a low-dimensional subspace can be computed on which the majority of the data resides.

The SVD of a matrix $\mathbf{X} \in \mathbb{C}^{n \times m}$ takes the form

$$\mathbf{X} = \mathbf{U\Sigma V}^*, \tag{7.4}$$

in terms of the following three matrices:

$$\mathbf{U} \in \mathbb{C}^{n \times n} \text{ is unitary,} \tag{7.5a}$$

$$\mathbf{V} \in \mathbb{C}^{m \times m} \text{ is unitary,} \tag{7.5b}$$

$$\mathbf{\Sigma} \in \mathbb{R}^{n \times m} \text{ is diagonal.} \tag{7.5c}$$

Additionally, it is assumed that the diagonal entries of $\mathbf{\Sigma}$ are nonnegative and ordered from largest to smallest so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, where $p = \min(m, n)$. The SVD of the matrix $\mathbf{X}$ thus shows that the matrix first applies a unitary transformation preserving the unit sphere via $\mathbf{V}^*$. This is followed by a stretching operation that creates an ellipse with principal semi-axes given by the matrix $\mathbf{\Sigma}$. Finally, the generated hyperellipse is rotated by the unitary transformation $\mathbf{U}$. Thus the image of a unit sphere under any $n \times m$ matrix is a hyperellipse. The following is the primary theorem concerning SVD [135].

**Theorem.** *Every matrix* $\mathbf{X} \in \mathbb{C}^{n \times m}$ *has an SVD* (7.4). *Furthermore, the singular values* $\{\sigma_j\}$ *are uniquely determined, and if* $\mathbf{X}$ *is square and* $\sigma_j$ *is distinct, the singular vectors* $\{\mathbf{u}_j\}$ *and* $\{\mathbf{v}_j\}$ *are uniquely determined up to complex signs (complex scalar factors of absolute value* 1*).*

The above theorem guarantees the existence of the SVD, but in practice, it still remains to be computed. This is a fairly straightforward process if one considers the following matrix products:

$$\mathbf{X}^*\mathbf{X} = (\mathbf{U\Sigma V}^*)^*(\mathbf{U\Sigma V}^*) = \mathbf{V\Sigma}^2\mathbf{V}^* \tag{7.6}$$

and

$$\mathbf{X}\mathbf{X}^* = (\mathbf{U\Sigma V}^*)(\mathbf{U\Sigma V}^*)^* = \mathbf{U\Sigma}^2\mathbf{U}^*. \tag{7.7}$$

Multiplying (7.6) and (7.7) on the right by $\mathbf{V}$ and $\mathbf{U}$, respectively, gives the two self-consistent eigenvalue problems

$$\mathbf{X}^*\mathbf{X}\mathbf{V} = \mathbf{V}\boldsymbol{\Sigma}^2, \tag{7.8a}$$

$$\mathbf{X}\mathbf{X}^*\mathbf{U} = \mathbf{U}\boldsymbol{\Sigma}^2. \tag{7.8b}$$

Thus if the normalized eigenvectors are found for these two equations, then the orthonormal basis vectors are produced for $\mathbf{U}$ and $\mathbf{V}$. Likewise, the square root of the eigenvalues of these equations produces the singular values $\sigma_j$.

**Theorem** (Schmidt–Eckart–Young–Mirsky theorem [118, 47, 96]). *For any N so that* $0 \le N \le p = \min\{m, n\}$*, we can define the partial sum*

$$\mathbf{X}_N = \sum_{j=1}^{N} \sigma_j \mathbf{u}_j \mathbf{v}_j^* \,. \tag{7.9}$$

*And if* $N = \min\{m, n\}$*, we define* $\sigma_{N+1} = 0$*. Then*

$$\|\mathbf{X} - \mathbf{X}_N\|_2 = \sigma_{N+1} \,. \tag{7.10}$$

*Likewise, if using the Frobenius norm, then*

$$\|\mathbf{X} - \mathbf{X}_N\|_F = \sqrt{\sigma_{N+1}^2 + \sigma_{N+2}^2 + \cdots + \sigma_p^2} \,. \tag{7.11}$$

The interpretation of this theorem is critical as it gives a geometrical perspective for understanding the SVD. Geometrically, the SVD gives *the best approximation of a hyperellipsoid by a line segment*, i. e., simply take the line segment to be the longest axis, i. e., that associated with the singular value $\sigma_1$. Continuing this idea, what is the best approximation by a two-dimensional ellipse? Take the longest and second longest axes, i. e., those associated with the singular values $\sigma_1$ and $\sigma_2$. After $r$ steps, the total energy in $\mathbf{X}$ is completely captured. Thus the SVD gives an algorithm for a least-squares fit allowing us to project the matrix onto low-dimensional representations in a formal, algorithmic way.

The SVD provides a principled way to find a low-rank subspace on which to project the evolution dynamics of the PDE in (7.1). Specifically, the first $r$ modes of a low-rank projection form the POD basis in (7.2) desired for model reduction

$$\boldsymbol{\Psi} = \mathbf{U}_r. \tag{7.12}$$

These basis modes are used to project the dynamics onto the dominant, low-rank subspace of activity as shown in (7.3). Of course, to use these POD modes, the dynamics of the governing PDE must be known in advance. Moreover, the Galerkin projection of the dynamics onto POD modes in (7.3) may be, depending on the underlying problem, unstable, requiring modification to stabilize the time-stepping scheme. Such stability

issues have been considered extensively by Carlberg and co-workers [34], Amsallem and Farhat [4], and Kalashnikova et al. [67]. Regardless, POD reductions arising from the SVD computation of the basis $\mathbf{\Psi}$ form the underpinnings of many ROMs [60, 14]. A significant advantage of the maturity of POD-based reductions is the ability to produce rigorous error bounds. Indeed, there is a rich literature on how to use the error properties of POD/SVD to derive rigorous error bounds for simulation as well as optimal control across a diverse set of applications [75, 76, 143, 110, 59, 136]. Such rigorous bounds provide trust regions and certifiable models for many critical application areas. Details of POD-based ROMs can be found in Chapter 2 of the current volume on *Model order reduction* [13].

Due to tremendous advances and innovations, modern large-scale simulations and/or the data collection process can quickly produce volumes of data that traditional methods could not easily analyze and diagnose in real-time. This emerging *big data* era requires diagnostic tools that can scale to meet the rapidly growing information acquired from new monitoring technologies which are producing increasingly fine-scale spatial and temporal measurements. In such cases, one can exploit new techniques that are capable of extracting the dominant global features of the high-dimensional dynamics. Specifically, emerging *randomized linear algebra* algorithms [55, 85, 48] are critically enabling for scalable *big data* applications, supplementing the *method of snapshots* [122] for efficient computation of the SVD. Randomized algorithms exploit the fact that the data themselves have low-rank features. Indeed, the method scales with the intrinsic rank of the dynamics rather than the dimension of the measurements/sensor space. This is in contrast to standard SVD/PCA/POD reductions which do not scale well with the data size. One can think of the scalable methods as being critically enabling for producing real-time analysis of emerging, streaming *big data* sets. Moreover, the dominant features of the data can be used for an efficient compression of the data for storage or reduced-order modeling applications [3]. Figure 7.1 outlines the basic algorithmic architecture which can be used for producing scalable SVD decompositions.

## 7.2.2 Dynamic mode decomposition

An alternative to POD is the DMD reduction [117]. Unlike POD, the DMD algorithm not only correlates spatial activity, but also enforces that various low-rank spatial modes be correlated in time, essentially merging the favorable aspects of POD/PCA in space and the Fourier transform in time. DMD is a matrix factorization method based upon the SVD algorithm. However, in addition to performing a low-rank SVD approximation, it further performs an eigendecomposition on a best-fit linear operator that advances measurements forward in time in the computed subspaces in order to extract critical temporal features. Thus the DMD method provides a spatio-temporal decomposition of data into a set of dynamic modes that are derived from snapshots or mea-
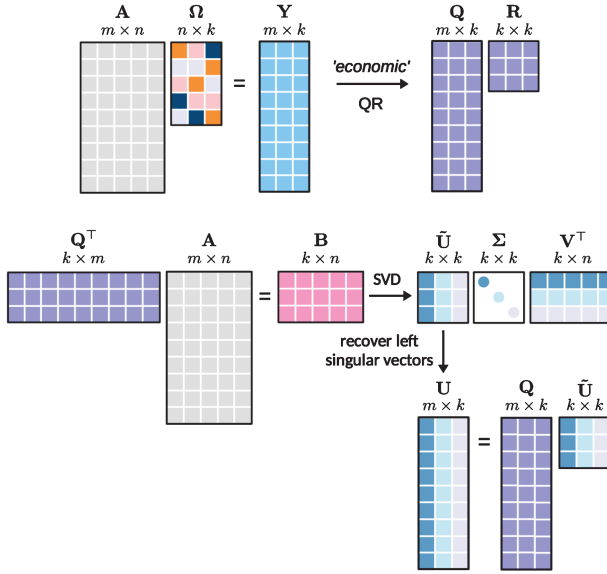
**Figure 7.1:** Illustration of the randomized matrix decomposition technique for scalable decompositions. The random sampling matrix **Ω** is used to produce a new matrix **Y** which can be decomposed using a QR decomposition. This leads to the construction of the matrix **B** which is used for approximating the left and right singular vector. *From Erichson et al.* [48].

surements of a given system in time, arranged as column state-vectors. The mathematics underlying the extraction of dynamic information from time-resolved snapshots is closely related to the idea of the Arnoldi algorithm, one of the workhorses of fast computational solvers. The DMD algorithm was originally designed to collect data at regularly spaced intervals of time. However, new innovations allow for both sparse spatial [27, 54] and temporal [139] collection of data as well as irregularly spaced collection times [6].

Like SVD, the DMD algorithm is based upon a regression. Thus there are a variety of algorithms that have been proposed in the literature for computing the DMD. A highly intuitive understanding of the DMD architecture was proposed by Tu et al. [138], which provides the *exact DMD* method.

**Definition: Exact dynamic mode decomposition** (Tu et al. 2014 [138]). Suppose we have a dynamical system (7.1) and two sets of measurement data

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{m-1} \\ | & | & & | \end{bmatrix}, \tag{7.13a}$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{u}'_1 & \mathbf{u}'_2 & \cdots & \mathbf{u}'_{m-1} \\ | & | & & | \end{bmatrix} \tag{7.13b}$$

so that $\mathbf{u}'_k = \mathbf{F}(\mathbf{u}_k)$, where $\mathbf{F}$ is the map corresponding to the evolution of (7.1) for time $\Delta t$. Exact DMD computes the leading eigendecomposition of the best-fit linear operator $\mathbf{A}$ relating the data $\mathbf{u}' \approx \mathbf{A}\mathbf{u}$ :

$$\mathbf{A} = \mathbf{X}'\mathbf{X}^{\dagger}. \tag{7.14}$$

The DMD modes, also called dynamic modes, are the eigenvectors of $\mathbf{A}$, and each DMD mode corresponds to a particular eigenvalue of $\mathbf{A}$.

The DMD framework takes an equation-free perspective where the original, non-linear dynamics may be unknown. Thus measurements of the system alone are used to approximate the dynamics and predict the future state. However, DMD can also be used when governing equations are known [2]. This DMD-Galerkin procedure represents a potential hybrid between the POD-Galerkin and DMD methods. The integration of DMD and POD can also be used for model reduction numerical schemes [146]. The DMD procedure constructs a proxy, locally linear dynamical system approximation to (7.1):

$$\mathbf{u}_{k+1} \approx \mathbf{A}\mathbf{u}_k, \tag{7.15}$$

whose well-known solution is

$$\mathbf{u}_k = \sum_{j=1}^{n} \boldsymbol{\phi}_j \lambda_j^k b_j = \boldsymbol{\Phi}\boldsymbol{\Lambda}^k \mathbf{b}, \tag{7.16}$$

where $\boldsymbol{\phi}_j$ and $\lambda_j$ are the eigenvectors and eigenvalues of the matrix $\mathbf{A}$, and the coefficients $b_j$ are the coordinates of the initial condition $\mathbf{u}_0$ in the eigenvector basis. The eigenvalues $\lambda$ of $\mathbf{A}$ determine the temporal dynamics of the system, at least in an asymptotic sense and for normal operators, i.e., transient dynamics are not well captured. It is often convenient to convert these eigenvalues to continuous time, $\omega = \log(\lambda)/\Delta t$, so the real parts of the eigenvalues $\omega$ determine growth and decay of the solution, and the imaginary parts determine oscillatory behaviors and their corresponding frequencies. The eigenvalues and eigenvectors are critically enabling for producing interpretable diagnostic features of the dynamics. It is important to note that the choice of the time step $\Delta t$ is critical in the DMD algorithm. The time step must be small enough to resolve the fastest time scales of relevance. A consequence of the linear model produced by the DMD algorithm is its inability to model transient phenomena over the snapshots sampled, aside from transient growth potentially produced by non-normal modes where eigenvalues are identical or nearly so.

The DMD algorithm produces a low-rank eigendecomposition of the matrix $\mathbf{A}$ that optimally fits the measured trajectory $\mathbf{u}_k$ for $k = 1, 2, \ldots, m$ snapshots in a least-squares sense so that $\|\mathbf{u}_{k+1} - \mathbf{A}\mathbf{u}_k\|_2$ is minimized across all points for $k = 1, 2, \ldots, m-1$. The optimality of the approximation holds only over the sampling window where $\mathbf{A}$ is constructed, and the approximate solution can be used to not only make future

state predictions, but also to derive dynamic modes critical for diagnostics. Indeed, in much of the literature where DMD is applied, it is primarily used as a diagnostic tool. This is much like POD analysis, where the POD modes are also primarily used for diagnostic purposes. Thus the DMD algorithm can be thought of as a modification of the SVD architecture which attempts to account for dynamic activity of the data. The eigendecomposition of the low rank space found from SVD enforces a Fourier mode time expansion which allows one to then make spatio-temporal correlations with the sampled data. Recently, DMD has also been rigorously connected to the spectral POD method [133].

Early variants of the DMD-computed eigenvalues that were biased by the presence of sensor noise [58, 44]. This was a direct result of the fact that the standard algorithms treated the data in a pairwise sense and favored the forward direction in time. Dawson et al. [44] and Hemati et al. [58] developed several methods for debiasing within the standard DMD framework. These methods have the advantage that they can be computed with essentially the same set of robust and fast tools as the standard DMD. As an alternative, the *optimized DMD* advocated by [38] treats all of the snapshots of the data at once. This avoids much of the bias of the original DMD but requires the solution of a potentially large nonlinear optimization problem. Askham and Kutz [6] recently showed that the optimized DMD algorithm could be rendered numerically tractable by leveraging the classical variable projection method [53]. Moreover, the optimized DMD method can be used to enforce all eigenvalues to have a real part less than or equal to zero. This ensures stability of solutions for future times as there are no growing modes. For input-output systems, DMD has also been modified through a postprocessing algorithm to generate a stable input-output model [15]. These methods show that DMD architectures can be imbued with advantageous stability properties for ROMs.

The variable projection algorithm is based upon the observation that the desired solutions of DMD are exponentials (7.16). Thus DMD is reformulated as an exponential data fitting (specifically, for inverse differential equations), an area of research that has been extensively developed and has many applications [52, 104]. The variable projection method leverages the special structure of the exponential data fitting problem, so that many of the unknowns may be eliminated from the optimization. An additional benefit of these tools is that the snapshots of data no longer need to be taken at regular intervals, i. e., the sample times do not need to be equispaced. The goal is then to rewrite the data matrix of snapshots as

$$\mathbf{X}^\top \approx \mathbf{\Phi}(\boldsymbol{\alpha})\mathbf{B} , \tag{7.17}$$

where $\mathbf{\Phi}(\boldsymbol{\alpha}) \in \mathbb{C}^{m \times r}$ with entries defined by $\Phi(\boldsymbol{\alpha})_{i,j} = \exp(\alpha_j t_i)$.

The preceding leads us to the following definition of the optimized DMD in terms of an exponential fitting problem. Suppose that $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{B}}$ solve

$$\text{minimize} \|\mathbf{X}^\top - \mathbf{\Phi}(\boldsymbol{\alpha})\mathbf{B}\|_F \quad \text{over } \boldsymbol{\alpha} \in \mathbb{C}^k, \mathbf{B} \in \mathbb{C}^{l \times n} . \tag{7.18}$$

The optimized DMD eigenvalues are then defined by $\lambda_i = \hat{\alpha}_i$ and the eigenmodes are defined by

$$\boldsymbol{\varphi}_i = \frac{1}{\|\hat{\mathbf{B}}^\top(:,i)\|_2} \hat{\mathbf{B}}^\top(:,i) \,, \tag{7.19}$$

where $\hat{\mathbf{B}}^\top(:,i)$ is the $i$-th column of $\hat{\mathbf{B}}^\top$. Details of the algorithm and code for computing the optimized DMD can be found in Askham and Kutz [6]. The improved and debiased decomposition (7.16) of this optimal DMD strategy are readily apparent in numerous examples. Moreover, a comparison of DMD variants shows how each method handles noise and takes on bias. Optimized DMD thus far outperforms all other variants at the cost of a nonlinear optimization.

A remarkable feature of the DMD algorithm is its modularity for mathematical enhancements. Specifically, the DMD algorithm can be engineered to exploit sparse sampling [27, 54], it can be modified to handle inputs and actuation [106], it can be used to more accurately approximate the Koopman operator when using judiciously chosen functions of the state-space [80], and it can easily decompose data into multiscale temporal features in order to produce a multiresolution DMD [79]. Few mathematical architectures are capable of seamlessly integrating such diverse modifications of the dynamical system. But since the DMD provides an approximation of a linear system, such modifications are easily constructed. Moreover, the DMD algorithm, unlike many other machine learning algorithms, is not data-intensive in comparison to most deep neural network architectures which require large labeled data sets. Thus a DMD approximation can always be achieved, especially as the first step in the algorithm is the SVD which is guaranteed to exist for any data matrix. However, for very large data sets, DMD can leverage randomized methods [55, 85, 48] to produce a scalable *randomized DMD* [49, 18].

DMD is closely related to the field of system identification, which identifies models from data, often for use with model-based controllers. Tu et al. [138] and Proctor et al. [106] established connections between DMD and several classical system identification approaches, including the eigensystem realization algorithm [64] and singular spectrum analysis (SSA) [20] in climate time-series analysis. Nearly all methods of system identification involve some form of regression of data onto dynamics, and the main distinction between the various techniques is the degree to which this regression is constrained. For example, DMD generates best-fit linear models.

## 7.2.3 Koopman theory and observable selection

Much of the challenge associated with predicting, estimating, controlling, and reducing complex systems arises from the inherent nonlinearity in the governing equations. Indeed, mathematical physics has a rich history in deriving coordinate transformations that simplify the dynamics and alleviate the challenge of nonlinearity. In 1931,

Koopman developed an alternative perspective to classical dynamical systems theory, showing that there is a linear, infinite-dimensional operator that acts on the Hilbert space of possible measurement functions of the system, advancing these measurements along the flow of the dynamics [71, 72]. Koopman's operator-theoretic perspective trades nonlinear dynamics for linear but infinite-dimensional dynamics, and was critical in Birkhoff's proof of the ergodic theorem [17, 97].

Recently, Koopman operator theory has seen a resurgence of interest [93, 29, 94], in large part because of the increasing availability of measurement data and improving computational capabilities. In 2005, Mezic showed that Koopman theory may be used to provide a modal decomposition of complex systems, providing direct relevance to engineering systems [93]. Since then, it has been shown that the DMD algorithm from fluid dynamics [117] actually approximates the Koopman operator [109], restricted to a set of linear measurements of the system; a more detailed treatment for fluid systems is given by Taira et al. [129].

The ability of Koopman analysis to transform nonlinear systems into a linear framework has tremendous promise to make complex systems amenable to optimal prediction, estimation, and control with simple techniques from linear systems theory. In a short time, Koopman theory has been extended to nonlinear estimation [125, 126] and control [106, 107], for example via model predictive control [73, 66], control in eigenfunction coordinates [65], and switching control [103]. However, Koopman theory appears to follow the principle of conservation of difficulty, in that finding the right nonlinear measurements that enable a tractable linear representation may be as challenging as solving the original problem. In a sense, obtaining Koopman embeddings may be seen as an expensive offline computation that enables fast and efficient online prediction, estimation, and control. In addition, the Koopman operator is one of two main candidates for analyzing a dynamical system using operator-based approaches, the other being the Perron–Frobenius operator. The Perron–Frobenius operator evolves probability density functions along the flow of the dynamics, while the Koopman operator evolves observable functions of the state. These two operators are adjoint to each other in appropriately defined function spaces and it should therefore theoretically not matter which one is used to study the system's behavior [70].

Before introducing the mathematical formulation of Koopman operator theory, we first consider the flow map $\mathbf{F}_{\Delta t}$ obtained by integrating the PDE in (7.1) for a short-time $\Delta t$, given by

$$\mathbf{u}_{k+1} = \mathbf{F}_{\Delta t}(\mathbf{u}_k). \tag{7.20}$$

The Koopman operator $\mathcal{K}$ is defined so that

$$\mathcal{K}_t g = g \circ \mathbf{F}_t, \tag{7.21}$$

where ∘ is the composition operator. For a discrete-time system with time step $\Delta t$, this becomes

$$\mathcal{K}_{\Delta t} g(\mathbf{u}_k) = g(\mathbf{F}_{\Delta t}(\mathbf{u}_k)) = g(\mathbf{u}_{k+1}). \tag{7.22}$$

In other words, the Koopman operator defines an infinite-dimensional linear dynamical system that advances the observation of the state $g_k = g(\mathbf{u}_k)$ to the next time step:

$$g(\mathbf{u}_{k+1}) = \mathcal{K}_{\Delta t} g(\mathbf{u}_k). \tag{7.23}$$

Note that this is true for *any* observable function $g$ and for any state $\mathbf{u}_k$.

Much of the challenge of modern Koopman theory is obtaining a finite-dimensional representation $\mathbf{K}$ of the infinite-dimensional operator $\mathcal{K}$. In practice, this amounts to discovering eigenfunctions of the Koopman operator, which are measurement functions that behave linearly when evolved forward in time. A discrete-time Koopman eigenfunction $\varphi(\mathbf{u})$ corresponding to eigenvalue $\lambda$ satisfies

$$\varphi(\mathbf{u}_{k+1}) = \mathcal{K}_{\Delta t} \varphi(\mathbf{u}_k) = \lambda \varphi(\mathbf{u}_k). \tag{7.24}$$

In continuous-time, a Koopman eigenfunction $\varphi(\mathbf{u})$ satisfies

$$\frac{d}{dt} \varphi(\mathbf{u}) = \mathcal{K} \varphi(\mathbf{u}) = \lambda \varphi(\mathbf{u}). \tag{7.25}$$

Obtaining Koopman eigenfunctions from data or from analytic expressions is a central applied challenge in modern dynamical systems. Discovering these eigenfunctions enables globally linear representations of strongly nonlinear systems. Applying the chain rule to the time derivative of the Koopman eigenfunction $\varphi(\mathbf{u})$ yields

$$\frac{d}{dt} \varphi(\mathbf{u}) = \nabla \varphi(\mathbf{u}) \cdot \dot{\mathbf{u}} = \nabla \varphi(\mathbf{u}) \cdot \mathbf{f}(\mathbf{u}). \tag{7.26}$$

Combined with (7.25), this results in a PDE for the eigenfunction $\varphi(\mathbf{u})$:

$$\nabla \varphi(\mathbf{u}) \cdot \mathbf{f}(\mathbf{u}) = \lambda \varphi(\mathbf{u}). \tag{7.27}$$

With this nonlinear PDE, it is possible to approximate the eigenfunctions, either by solving for the Laurent series or with data via regression, both of which are explored below. This formulation assumes that the dynamics are both continuous and differentiable. The discrete-time dynamics in (7.20) are more general, although in many examples the continuous-time dynamics have a simpler representation than the discrete-time map for long times. Koopman analysis has recently been extended to the continuous PDE formulation, rather than just the high-dimensional discretized ODE context, for example, showing that the Cole–Hopf transform is a Koopman embedding for Burgers' equation [80].

There are many approaches to obtain finite-dimensional approximations to the Koopman operator. DMD is a representation based on linear observables [109], which has been extended to nonlinear observables in the *extended* DMD (eDMD) [144] and the variational approach of conformation dynamics [99, 100]. In all of these cases, it is important that the measurements are chosen to form a Koopman-invariant subspace [23]; otherwise, the projection of the Koopman operator onto this subspace will result in spurious eigenvalues and eigenfunctions.

In eDMD, an augmented state is constructed:

$$\mathbf{y} = \mathbf{\Theta}^T(\mathbf{u}) = \begin{bmatrix} \theta_1(\mathbf{u}) \\ \theta_2(\mathbf{u}) \\ \vdots \\ \theta_p(\mathbf{u}) \end{bmatrix}. \tag{7.28}$$

The projection $\mathbf{\Theta}$ may contain the original state $\mathbf{u}$ as well as nonlinear measurements, so often $p \gg n$. Next, two data matrices are constructed, as in DMD:

$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_m \\ | & | & & | \end{bmatrix}, \quad \mathbf{Y}' = \begin{bmatrix} | & | & & | \\ \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_{m+1} \\ | & | & & | \end{bmatrix}. \tag{7.29a}$$

Finally, a best-fit linear operator $\mathbf{A_Y}$ is constructed that maps $\mathbf{Y}$ into $\mathbf{Y}'$:

$$\mathbf{A_Y} = \mathrm{argmin}_{\mathbf{A_Y}} \|\mathbf{Y}' - \mathbf{A_Y}\mathbf{Y}\|_2 = \mathbf{Y}'\mathbf{Y}^\dagger. \tag{7.30}$$

This regression may be written in terms of the data matrices $\mathbf{\Theta}(\mathbf{X})$ and $\mathbf{\Theta}(\mathbf{X}')$:

$$\mathbf{A_Y} = \mathrm{argmin}_{\mathbf{A_Y}} \|\mathbf{\Theta}^T(\mathbf{X}') - \mathbf{A_Y}\mathbf{\Theta}^T(\mathbf{X})\|_2 = \mathbf{\Theta}^T(\mathbf{X}')(\mathbf{\Theta}^T(\mathbf{X}))^\dagger. \tag{7.31}$$

The resulting nonlinear model for $u_k$ is given by the proxy eDMD variable $\mathbf{y}_{k+1} = \mathbf{A_Y}\mathbf{y}_k$. Because the augmented vector $\mathbf{y}$ may be significantly larger than the state $\mathbf{u}$, kernel methods are often employed to compute this regression [145]. In principle, the enriched library $\mathbf{\Theta}$ provides a larger basis in which to approximate the Koopman operator. It has been shown recently that in the limit of infinite snapshots, the eDMD operator converges to the Koopman operator projected onto the subspace spanned by $\mathbf{\Theta}$ [144, 70, 74]. However, if $\mathbf{\Theta}$ does not span a Koopman-invariant subspace, then the projected operator may not have any resemblance to the original Koopman operator, as all of the eigenvalues and eigenvectors may be different. In fact, it was shown that the eDMD operator will have spurious eigenvalues and eigenvectors unless it is represented in terms of a Koopman-invariant subspace [23]. Therefore, it is essential to use validation and cross-validation techniques to ensure that eDMD models are not overfit, as discussed below. For example, it was shown that eDMD cannot contain the original state $\mathbf{u}$ as a measurement and represent a system that has multiple fixed points,

periodic orbits, or other attractors, because these systems cannot be topologically conjugate to a finite-dimensional linear system [23]. Recently, researchers have been leveraging the representational power of deep neural networks to identify Koopman eigenfunctions and approximate Koopman operators [130, 149, 92, 141, 101, 83]. In the next section, we will discuss an alternative approach to obtain a Koopman-invariant subspace based on time delay coordinates [22].

## 7.2.4 Time-delay embeddings for Koopman embeddings

Instead of advancing instantaneous linear or nonlinear measurements of the state of a system directly, as in DMD, it may be possible to obtain intrinsic measurement coordinates for Koopman based on time-delayed measurements of the system [127, 22, 5, 43, 68]. This perspective is data-driven, relying on the wealth of information from previous measurements to inform the future. Unlike a linear or weakly nonlinear system, where trajectories may get trapped at fixed points or on periodic orbits, chaotic dynamics are particularly well suited to this analysis: Trajectories evolve to densely fill an attractor, so more data provide more information. The use of delay coordinates may be especially important for systems with long-term memory effects, where the Koopman approach has recently been shown to provide a successful analysis tool [128]. Interestingly, a connection between the Koopman operator and the Takens embedding was explored as early as in 2004 [95], where a stochastic Koopman operator is defined and a statistical Takens theorem is proven. One version of time-delay embeddings, the HAVOK, has been used successfully to diagnose a diverse set of dynamical systems [22]. More broadly, there are a number of analysis tools that can be applied to the Hankel matrix for analysis of dynamics [68].

The time-delay measurement scheme is shown schematically in Figure 7.2, as illustrated on the Lorenz system for a single time-series measurement of the first variable, $x(t)$. If the conditions of the Takens embedding theorem are satisfied [131], it is possible to obtain a diffeomorphism between a delay-embedded attractor and the attractor in the original coordinates. We then obtain eigentime-delay coordinates from a time series of a single measurement $x(t)$ by taking the SVD of the Hankel matrix $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} x(t_1) & x(t_2) & \cdots & x(t_{m_c}) \\ x(t_2) & x(t_3) & \cdots & x(t_{m_c+1}) \\ \vdots & \vdots & \ddots & \vdots \\ x(t_{m_o}) & x(t_{m_o+1}) & \cdots & x(t_m) \end{bmatrix} = \mathbf{\Psi}_{\text{TD}} \mathbf{\Sigma} \mathbf{V}^*, \tag{7.32}$$

where $m_c$ is the number of snapshots and $m_o$ is the total number of delays. The columns of $\mathbf{\Psi}_{\text{TD}}$ and $\mathbf{V}$ from the SVD are arranged hierarchically by their ability to model the columns and rows of $\mathbf{H}$, respectively. Often, $\mathbf{H}$ may admit a low-rank approximation by the first $r$ columns of $\mathbf{\Psi}_{\text{TD}}$ and $\mathbf{V}$. Note that the Hankel matrix in (7.32) is
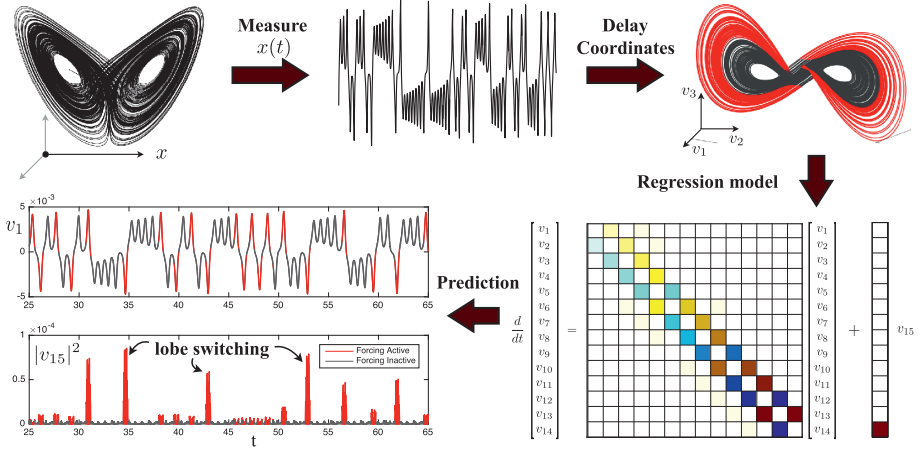
**Figure 7.2:** Schematic of the Hankel alternative view of Koopman (HAVOK) algorithm [22], as illustrated on the Lorenz 63 system. A time series $x(t)$ is stacked into a Hankel matrix **H**. The SVD of **H** yields a hierarchy of *eigen*time series that produce a delay-embedded attractor. A best-fit linear regression model is obtained on the delay coordinates **v**; the linear fit for the first $r - 1$ variables is excellent, but the last coordinate $v_r$ is not well modeled as linear. Instead, $v_r$ is an input that forces the first $r - 1$ variables. Rare forcing events correspond to lobe switching in the chaotic dynamics. *From Brunton and Kutz* [24]*, modified from* [22].

the basis of the eigensystem realization algorithm [64] in linear system identification and SSA [20] in climate time-series analysis.

The low-rank approximation to (7.32) provides a *data-driven* measurement system that is approximately invariant to the Koopman operator for states on the attractor. By definition, the dynamics map the attractor into itself, making it *invariant* to the flow. In other words, the columns of **U** form a Koopman-invariant subspace. We may rewrite (7.32) with the Koopman operator $\mathcal{K} \triangleq \mathcal{K}_{\Delta t}$:

$$
\mathbf{H} = \begin{bmatrix}
x(t_1) & \mathcal{K}x(t_1) & \cdots & \mathcal{K}^{m_c-1}x(t_1) \\
\mathcal{K}x(t_1) & \mathcal{K}^2x(t_1) & \cdots & \mathcal{K}^{m_c}x(t_1) \\
\vdots & \vdots & \ddots & \vdots \\
\mathcal{K}^{m_o-1}x(t_1) & \mathcal{K}^{m_o}x(t_1) & \cdots & \mathcal{K}^{m-1}x(t_1)
\end{bmatrix}.
\tag{7.33}
$$

The columns of (7.32) are well approximated by the first $r$ columns of $\mathbf{\Psi}_{\text{TD}}$. The first $r$ columns of **V** provide a time series of the magnitude of each of the columns of $\mathbf{\Psi}_{\text{TD}}\mathbf{\Sigma}$ in the data. By plotting the first three columns of **V**, we obtain an embedded attractor for the Lorenz system (Figure 7.2).

The connection between eigentime-delay coordinates from (7.32) and the Koopman operator motivates a linear regression model on the variables in **V**. Even with an approximately Koopman-invariant measurement system, there remain challenges

to identifying a linear model for a chaotic system. A linear model, however detailed, cannot capture multiple fixed points or the unpredictable behavior characteristic of chaos with a positive Lyapunov exponent [23]. Instead of constructing a closed linear model for the first $r$ variables in $\mathbf{V}$, we build a linear model on the first $r - 1$ variables and recast the last variable, $v_r$, as a forcing term:

$$\frac{d}{dt}\mathbf{v}(t) = \mathbf{A}\mathbf{v}(t) + \mathbf{B}v_r(t), \tag{7.34}$$

where $\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_{r-1} \end{bmatrix}^T$ is a vector of the first $r - 1$ eigentime-delay coordinates. Other work has investigated the splitting of dynamics into deterministic linear and chaotic stochastic dynamics [93].

In all of the examples explored in [22], the linear model on the first $r - 1$ terms is accurate, while no linear model represents $v_r$. Instead, $v_r$ is an input forcing to the linear dynamics in (7.34), which approximates the nonlinear dynamics. The statistics of $v_r(t)$ are non-Gaussian, with long tails corresponding to rare-event forcing that drives lobe switching in the Lorenz system; this is related to rare-event forcing distributions observed and modeled by others [86, 113, 87].

# 7.3 Data-driven model discovery

For many modern complex systems of interest, such as in materials science, neuroscience, epidemiology, climate science, and finance, there is a basic lack of physical laws and governing equations. Even when governing equations are available, for example in fluid turbulence, protein folding, and combustion, the equations are so complex that they are not readily amenable to analysis. With increasingly complex systems, and the emergence of powerful computing architectures and big data, the paradigm is shifting to data-driven methods to discover governing equations [19, 119, 25, 111].

## 7.3.1 SINDy: sparse identification of nonlinear dynamics

Discovering ROMs from data is a central challenge in modern computational physics. Typically, the form of a candidate model is either constrained via prior knowledge of the governing equations, as in Galerkin projection [98, 9, 34], or a handful of heuristic models are tested and parameters are optimized to fit data. Alternatively, best-fit linear models may be obtained using DMD. Simultaneously identifying the nonlinear structure and parameters of a model from data is considerably more challenging, as there are combinatorially many possible model structures.

The SINDy algorithm [25] bypasses the intractable combinatorial search through all possible model structures, leveraging the fact that many dynamical systems

$$\frac{d}{dt}\mathbf{a} = \mathbf{f}(\mathbf{a}) \tag{7.35}$$

have dynamics $\mathbf{f}$ with only a few active terms in the space of possible right-hand side functions; for example, the Lorenz equations (Figure 7.3) only have a few linear and quadratic interaction terms per equation. Here, $\mathbf{a} \in \mathbb{R}^r$ is a low-dimensional state, for example obtained via POD/SVD [25, 81], or constructed from physically realizable measurements, such as lift, drag, and the derivative of lift for aerodynamic systems [82].
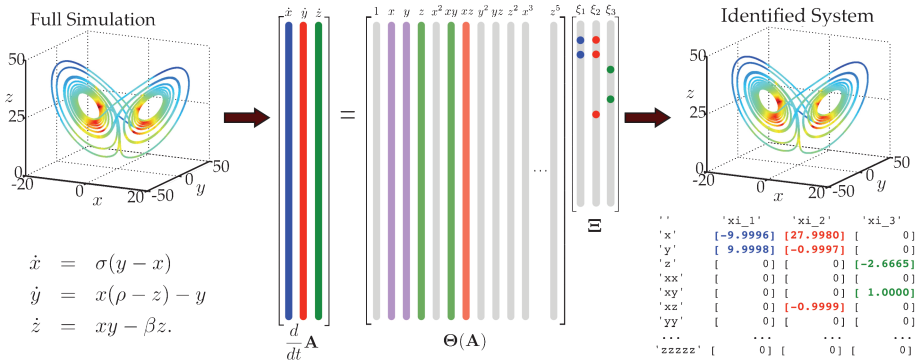


**Figure 7.3:** Schematic of the sparse identification of nonlinear dynamics (SINDy) algorithm [25], as illustrated on the Lorenz 63 system. *From Brunton and Kutz* [24]*, modified from* [25].

We then seek to approximate $\mathbf{f}$ by a generalized linear model in a set of candidate basis functions $\theta_k(\mathbf{a})$

$$\mathbf{f}(\mathbf{a}) \approx \sum_{k=1}^{p} \theta_k(\mathbf{a})\xi_k = \mathbf{\Theta}(\mathbf{a})\mathbf{\Xi}, \tag{7.36}$$

with the fewest nonzero terms in $\mathbf{\Xi}$. It is possible to solve for the relevant terms that are active in the dynamics using sparse regression [132, 155, 57, 63] that penalizes the number of terms in the dynamics and scales well to large problems.

First, time-series data are collected from (7.35) and formed into a data matrix:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}(t_1) & \mathbf{a}(t_2) & \cdots \mathbf{a}(t_m) \end{bmatrix}^T. \tag{7.37}$$

A similar matrix of derivatives is formed:

$$\dot{\mathbf{A}} = \begin{bmatrix} \dot{\mathbf{a}}(t_1) & \dot{\mathbf{a}}(t_2) & \cdots \dot{\mathbf{a}}(t_m) \end{bmatrix}^T. \tag{7.38}$$

In practice, this may be computed directly from the data in **A** using a numerical differencing scheme, for instance. However, for noisy data, the total-variation regularized derivative tends to provide numerically robust derivatives [36]. Alternatively, it is possible to formulate the SINDy algorithm for discrete-time systems $\mathbf{a}_{k+1} = \mathbf{F}(\mathbf{a}_k)$, as in the DMD algorithm, and avoid derivatives entirely.

A library of candidate nonlinear functions $\boldsymbol{\Theta}(\mathbf{A})$ may be constructed from the data in **A**:

$$\boldsymbol{\Theta}(\mathbf{A}) = \begin{bmatrix} \mathbf{1} & \mathbf{A} & \mathbf{A}^2 & \cdots & \mathbf{A}^d & \cdots & \sin(\mathbf{A}) & \cdots \end{bmatrix}. \tag{7.39}$$

Here, the matrix $\mathbf{A}^d$ denotes a matrix with column vectors given by all possible time series of $d$-th-degree polynomials in the state **a**. In general, this library of candidate functions is only limited by one's imagination, but polynomials, trigonometric functions, and other well-known functions are a good starting point. The matrix $\boldsymbol{\Theta}$ tends to be ill-conditioned as the library elements, such as polynomials, are often not orthogonal. Indeed, they can often be nearly aligned over the time course where the library is evaluated. Despite the high condition number, the sparse selection advocated below is able to identify the correct dynamics provided the noise level is sufficiently small.

The dynamical system in (7.35) may now be represented in terms of the data matrices in (7.38) and (7.39) as

$$\dot{\mathbf{A}} = \boldsymbol{\Theta}(\mathbf{A})\boldsymbol{\Xi}. \tag{7.40}$$

Each column $\boldsymbol{\xi}_k$ in $\boldsymbol{\Xi}$ is a vector of coefficients determining the active terms in the $k$-th row in (7.35). A parsimonious model will provide an accurate model fit in (7.40) with as few terms as possible in $\boldsymbol{\Xi}$. Such a model may be identified using a convex $\ell_1$-regularized sparse regression:

$$\boldsymbol{\xi}_k = \mathrm{argmin}_{\boldsymbol{\xi}_k'} \|\dot{\mathbf{A}}_k - \boldsymbol{\Theta}(\mathbf{A})\boldsymbol{\xi}_k'\|_2 + \lambda \|\boldsymbol{\xi}_k'\|_1. \tag{7.41}$$

Here, $\dot{\mathbf{A}}_k$ is the $k$-th column of $\dot{\mathbf{A}}$ and $\lambda$ is a sparsity-promoting regularization weight, typically chosen by simple hyperparameter tuning. Sparse regression, such as the LASSO [132] or the sequential thresholded least-squares (STLS) algorithm used in SINDy [25], improves the numerical robustness of this identification for noisy overdetermined problems, in contrast to earlier methods [140] that used compressed sensing [45, 30, 32, 31, 33, 10, 137]. We advocate STLS to select active terms; there are recent guarantees on the convergence of this algorithm [152], and it has also been formalized in a general sparse regression framework called SR3 [154].

The sparse vectors $\boldsymbol{\xi}_k$ may be synthesized into a dynamical system:

$$\dot{a}_k = \boldsymbol{\Theta}(\mathbf{a})\boldsymbol{\xi}_k. \tag{7.42}$$

Note that $x_k$ is the $k$-th element of **a** and $\boldsymbol{\Theta}(\mathbf{a})$ is a row vector of symbolic functions of **a**, as opposed to the data matrix $\boldsymbol{\Theta}(\mathbf{A})$.

The result of the SINDy regression is a parsimonious model that includes only the most important terms required to explain the observed behavior. The sparse regression procedure used to identify the most parsimonious nonlinear model is a convex procedure. The alternative approach, which involves regression onto every possible sparse nonlinear structure, constitutes an intractable brute-force search through the combinatorially many-candidate model forms. SINDy bypasses this combinatorial search with modern convex optimization and machine learning. It is interesting to note that for discrete-time dynamics, if $\Theta(\mathbf{A})$ consists only of linear terms, and if we remove the sparsity promoting term by setting $\lambda = 0$, then this algorithm reduces to DMD [117, 109, 138, 78]. If a least-squares regression is used, as in DMD, then even a small amount of measurement error or numerical round-off will lead to every term in the library being active in the dynamics, which is nonphysical. A major benefit of the SINDy architecture is the ability to identify parsimonious models that contain only the required nonlinear terms, resulting in interpretable models that avoid overfitting.

### 7.3.1.1 Extensions and applications

Because SINDy is formulated in terms of linear regression in a nonlinear library, it is highly extensible. The SINDy framework has been recently generalized by Loiseau and Brunton [81] to incorporate known physical constraints and symmetries in the equations by implementing a constrained sequentially thresholded least-squares optimization. In particular, energy-preserving constraints on the quadratic nonlinearities in the Navier–Stokes equations were imposed to identify fluid systems [81], where it is known that these constraints promote stability [86, 9, 34]. This work also showed that polynomial libraries are particularly useful for building models of fluid flows in terms of POD coefficients, yielding interpretable models that are related to classical Galerkin projection [25, 81]. Loiseau et al. [82] also demonstrated the ability of SINDy to identify dynamical systems models of high-dimensional systems, such as fluid flows, from a few physical sensor measurements, such as lift and drag measurements on the cylinder. SINDy has also been applied to identify models in nonlinear optics [123] and plasma physics [40]. For actuated systems, SINDy has been generalized to include inputs and control [26], and these models are highly effective for model predictive control [66]. It is also possible to extend the SINDy algorithm to identify dynamics with rational function nonlinearities [88], with integral terms [116], and based on highly corrupt and incomplete data [134]. SINDy was also recently extended to incorporate information criteria for objective model selection [89], and to identify models with hidden variables using delay coordinates [22]. Finally, the SINDy framework was generalized to include partial derivatives, enabling the identification of PDE models [111, 115], which will be explored below.

More generally, the use of sparsity-promoting methods in dynamics is quite recent [140, 114, 102, 84, 28, 105, 8, 7, 21, 90, 91]. Other techniques for dynami-

cal system discovery include methods to discover equations from time series [39], equation-free modeling [69], empirical dynamic modeling [124, 148], modeling emergent behavior [108], the nonlinear autoregressive model with exogenous inputs (NARMAX) [51, 153, 16, 121], and automated inference of dynamics [120, 41, 42]. Broadly speaking, these techniques may be classified as system identification, where methods from statistics and machine learning are used to identify dynamical systems from data. Nearly all methods of system identification involve some form of regression of data onto dynamics, and the main distinction between the various techniques is the degree to which this regression is constrained. For example, DMD generates best-fit linear models. Recent nonlinear regression techniques have produced nonlinear dynamic models that preserve physical constraints, such as conservation of energy. Yao and Bollt previously formulated the dynamical system identification problem as a similar linear inverse problem [147], although sparsity-promoting regression was not used, so the resulting models included all terms in $\Theta$. In addition, SINDy is closely related to NARMAX [16], which identifies the structure of models from time-series data through an orthogonal least-squares procedure.

### 7.3.2  Model discovery for PDEs

A major extension of the SINDy modeling framework generalized the library to include partial derivatives, enabling the identification of PDEs [111, 115]. The resulting algorithm, called the PDE functional identification of nonlinear dynamics (PDE-FIND), shown in Figure 7.4, has been demonstrated to successfully identify several canonical PDEs from classical physics, purely from noisy data. These PDEs include Navier–Stokes, Kuramoto–Sivashinsky, Schrödinger, reaction diffusion, Burgers, Korteweg–de Vries (KdV), and the diffusion equation for Brownian motion [111].

PDE-FIND is similar to SINDy, in that it is based on sparse regression in a library constructed from measurement data. PDE-FIND is outlined below for PDEs in a single variable, although the theory is readily generalized to higher dimensional PDEs. The spatial time-series data are arranged into a single column vector $\Upsilon \in \mathbb{C}^{mn}$, representing data collected over $m$ time points and $n$ spatial locations. Additional inputs, such as a known potential for the Schrödinger equation, or the magnitude of complex data, is arranged into a column vector $\mathbf{Q} \in \mathbb{C}^{mn}$. Next, a library $\Theta(\Upsilon, \mathbf{Q}) \in \mathbb{C}^{mn \times D}$ of $D$ candidate linear and nonlinear terms and partial derivatives for the PDE is constructed. Derivatives are taken either using finite differences for clean data, or when noise is added, with polynomial interpolation. The candidate linear and nonlinear terms and partial derivatives are then combined into a matrix $\Theta(\Upsilon, \mathbf{Q})$ which takes the form

$$\Theta(\Upsilon, \mathbf{Q}) = \begin{bmatrix} \mathbf{1} & \Upsilon & \Upsilon^2 & \dots & \mathbf{Q} & \dots & \Upsilon_x & \Upsilon\Upsilon_x & \dots \end{bmatrix}. \tag{7.43}$$

Each column of $\Theta$ contains all of the values of a particular candidate function across all of the $mn$ space-time grid points on which data are collected. The time derivative
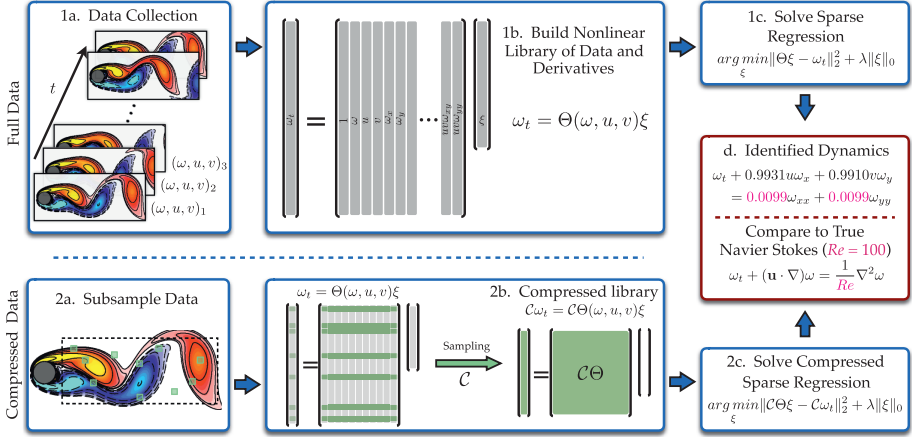
**Figure 7.4:** Schematic of PDE-FIND [111], as illustrated on the fluid flow past a circular cylinder. *From Rudy et al.* [111].

$\Upsilon_t$ is also computed and reshaped into a column vector. As an example, a column of $\boldsymbol{\Theta}(\boldsymbol{\Upsilon}, \mathbf{Q})$ may be $qu_x^2$.

The PDE evolution can be expressed in this library as follows:

$$\boldsymbol{\Upsilon}_t = \boldsymbol{\Theta}(\boldsymbol{\Upsilon}, \mathbf{Q})\boldsymbol{\xi}. \tag{7.44}$$

Each entry in $\boldsymbol{\xi}$ is a coefficient corresponding to a term in the PDE, and for canonical PDEs, the vector $\boldsymbol{\xi}$ is *sparse*, meaning that only a few terms are active.

If the library $\boldsymbol{\Theta}$ has a sufficiently rich column space that the dynamics are in its span, then the PDE should be well represented by (7.44) with a sparse vector of coefficients $\boldsymbol{\xi}$. To identify the few active terms in the dynamics, a sparsity-promoting regression is employed, as in SINDy. Importantly, the regression problem in (7.44) may be poorly conditioned. Errors in computing the derivatives will be magnified by numerical errors when inverting $\boldsymbol{\Theta}$. Thus a least-squares regression radically changes the qualitative nature of the inferred dynamics.

In general, we seek the sparsest vector $\boldsymbol{\xi}$ that satisfies (7.44) with a small residual. Instead of an intractable combinatorial search through all possible sparse vector structures, a common technique is to relax the problem to a convex $\ell_1$-regularized least-squares [132]; however, this tends to perform poorly with highly correlated data. Instead, we use ridge regression with hard thresholding, which we call sequential threshold ridge regression. For a given tolerance and threshold $\lambda$, this gives a sparse approximation to $\boldsymbol{\xi}$.

We iteratively refine the tolerance of Algorithm 1 to find the best predictor based on the selection criteria,

$$\hat{\boldsymbol{\xi}} = \text{argmin}_{\boldsymbol{\xi}} \left\| \boldsymbol{\Theta}(\boldsymbol{\Upsilon}, \mathbf{Q})\boldsymbol{\xi} - \boldsymbol{\Upsilon}_t \right\|_2^2 + \epsilon \kappa(\boldsymbol{\Theta}(\boldsymbol{\Upsilon}, \mathbf{Q})) \|\boldsymbol{\xi}\|_0, \tag{7.45}$$

where $\kappa(\mathbf{\Theta})$ is the condition number of the matrix $\mathbf{\Theta}$, providing stronger regularization for ill-posed problems. Penalizing $\|\boldsymbol{\xi}\|_0$ discourages overfitting by selecting from the optimal position in a Pareto front. While in general this problem is *NP*-hard we are restricting it to solutions generated via the STRidge algorithm, which promotes hard thresholding. Such hard thresholding has been recently shown to be a proxy for the $\ell_0$-norm [154].

As in the SINDy algorithm, it is important to provide sufficiently rich training data to disambiguate between several different models. For example, if only a single traveling wave from the KdV equation is analyzed, the method incorrectly identifies the standard linear advection equation, as this is the simplest equation that describes a single traveling wave. However, if two traveling waves of different amplitudes are analyzed, the KdV equation is correctly identified, as it describes the different amplitude-dependent wave speeds [111].

The PDE-FIND algorithm can also be used to identify PDEs based on Lagrangian measurements that follow the path of individual particles. For example, it is possible to identify the diffusion equation describing Brownian motion of a particle based on a single long time-series measurement of the particle position. In this example, the time series is broken up into several short sequences, and the evolution of the distribution of these positions is used to identify the diffusion equation [111].

# 7.4 Data-driven ROMs

The methods detailed in the previous sections can be integrated with traditional model reduction architectures. In what follows, we highlight how such methods can be used in a data-driven way to construct ROM models in a nonintrusive, efficient manner.

## 7.4.1 Application of DMD and Koopman to ROM models

DMD provides an alternative approach to computing the projection of the nonlinearity onto the rank-$r$ POD subspace in (7.3). Specifically, instead of using POD modes and gappy sampling for approximation of the nonlinear, low-rank contribution to the dynamics, DMD is used to directly compute a time evolution of the nonlinearity $\mathbf{\Psi}^T\mathbf{N}(\mathbf{\Psi}\mathbf{a})$ from snapshot data. Like the DEIM interpolation procedure [37], the DMD algorithm will proceed by constructing a snapshot matrix of the nonlinearity:

$$\mathbf{X}_{\text{NL}} = \begin{bmatrix} \vert & \vert & & \vert \\ \mathbf{N}_1 & \mathbf{N}_2 & \cdots & \mathbf{N}_m \\ \vert & \vert & & \vert \end{bmatrix}, \tag{7.46}$$

where the columns $\mathbf{N}_k = \mathbf{N}(\mathbf{u}(t_k), \mathbf{x}, t_k) \in \mathbb{C}^n$ are evaluations of the nonlinearity at time $t_k$.

Following (7.16), a DMD of the matrix $\mathbf{X}_{\mathrm{NL}}$ gives a low-rank approximation of the form

$$\mathbf{N}(\mathbf{u}(t), \mathbf{x}, t) = \mathbf{\Phi}_{\mathrm{NL}} \exp(\mathbf{\Omega}_{\mathrm{NL}} t) \mathbf{b}_{\mathrm{NL}}. \tag{7.47}$$

This low-rank approximation is achieved directly with further recourse to gappy interpolation for projecting back the DMD modes. The approximation can be used to modify (7.3) so as to achieve the following low-rank model:

$$\frac{d\mathbf{a}}{dt} = \mathbf{\Psi}^T \mathbf{L} \mathbf{\Psi} \mathbf{a} + \mathbf{\Psi}^T \mathbf{\Phi}_{\mathrm{NL}} \exp(\mathbf{\Omega}_{\mathrm{NL}} t) \mathbf{b}_{\mathrm{NL}}. \tag{7.48}$$

This integration of POD and DMD methods has been shown to provide performance increases in comparison to POD alone [2]. Moreover, the technique can be integrated with randomized linear algebra decomposition methods to achieve further enhancements in computational speed and scalability. Alla and Kutz further show that the POD-DMD integration competes well with POD with DEIM in terms of accuracy, while significantly outperforming it in terms of computation time. The DMD algorithm itself is faster than POD with DEIM and POD-DMD, but suffers from poor accuracy. One can also envision using a DMD-DMD reduction whereby a projection-based reduction with DMD-Galerkin is performed along with a hyperreduction with DMD. DMD-based ROM models have also recently been successfully demonstrated in a number of technical applications [1, 56]. A more detailed analysis of interpolation methods can be found in Chapter 7 of Volume 1 of *Model order reduction* [12].

## 7.4.2 Application of SINDy for ROMs

The SINDy algorithm can also be used to construct ROM architectures (7.2) from data alone, i. e., no governing equations are known a priori. As an example, the flow past a cylinder (Figure 7.5) provides a model with a rich history in fluid mechanics and dynamical systems [98]. It has long been theorized that turbulence is the result of a series of Hopf bifurcations that occur as the flow velocity increases [112], giving rise to more rich and intricate structures in the fluid. After 15 years, the first Hopf bifurcation was discovered in a fluid system, in the transition from a steady laminar wake to laminar periodic vortex shedding at Reynolds number 47 [62, 151]. This discovery led to a long-standing debate about how a Hopf bifurcation, with cubic nonlinearity, can be exhibited in a Navier–Stokes fluid with quadratic nonlinearities. After 15 more years, this was resolved using a separation of time scales and a mean-field model by Noack et al. [98]. It was shown that coupling between oscillatory modes and the base flow gives rise to a slow manifold, resulting in algebraic terms that approximate cubic nonlinearities on slow time scales.
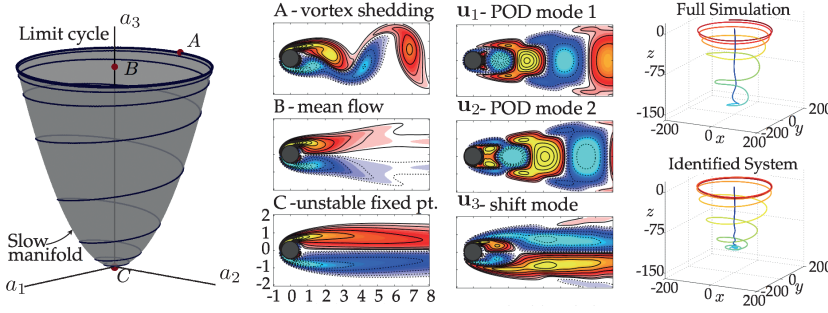
**Figure 7.5:** The vortex shedding past a cylinder is a prototypical example in fluid dynamics, with relevance to many applications, including drag reduction behind vehicles. Vortex shedding is the result of a Hopf bifurcation. However, because the Navier–Stokes equations have quadratic nonlinearity, it is necessary to employ a mean-field model with a separation of time scales, where a fast mean-field deformation is slave to the slow vortex shedding dynamics. The parabolic slow manifold is shown (left), with the unstable fixed point (C), mean flow (B), and vortex shedding (A). A POD basis and shift mode are used to reduce the dimension of the problem (middle right). The identified dynamics closely match the true trajectory in POD coordinates, and they capture the quadratic nonlinearity and time scales associated with the mean-field model. *From Brunton, Proctor and Kutz* [25].

This example provides a compelling test case for the proposed ROM-SINDy algorithm, since the underlying form of the dynamics took nearly three decades for experts in the community to uncover. Because the state dimension is large, it is advantageous to reduce the dimension of the system. POD provides a low-rank basis resulting in a hierarchy of orthonormal modes that, when truncated, capture the most energy of the original system for the given rank truncation. The first two most energetic POD modes capture a significant portion of the energy, and steady-state vortex shedding is a limit cycle in these coordinates. An additional mode, called the shift mode, is included to capture the transient dynamics connecting the unstable steady state with the mean of the limit cycle [98].

In the dominant POD coordinate system (rank $r = 3$), the mean-field model $\dot{\mathbf{a}} = \mathbf{f}(\mathbf{a})$ for the cylinder dynamics is discovered by SINDy to be [25]:

$$\dot{a}_1 = \mu a_1 - \omega a_2 + A a_1 a_3, \tag{7.49a}$$

$$\dot{a}_2 = \omega a_1 + \mu a_2 + A a_2 a_3, \tag{7.49b}$$

$$\dot{a}_3 = -\lambda(a_3 - a_1^2 - a_2^2). \tag{7.49c}$$

Note that the governing equations for $\mathbf{a}(t)$ in (7.49) are closely related to the slow-manifold formulation of Noack et al. [98] formulated using the standard Galerkin-POD projection. Specifically, it discovers the correct model $\dot{\mathbf{a}} = \mathbf{f}(\mathbf{a})$ with quadratic nonlinearities and reproduces a parabolic slow manifold. The $a_3$ variable corresponds to the shift-mode of Noack et al. [98], and if $\lambda$ is large, so that the $a_3$-dynamics are fast, then the mean flow rapidly corrects to be on the slow manifold $a_3 = a_1^2 + a_2^2$ given

by the amplitude of vortex shedding. When substituting this algebraic relationship into equations (7.49a) and (7.49b), we recover the Hopf normal form on the slow manifold. Note that derivative measurements are not available, but are computed from the state variables. When the training data do not include trajectories that originate from the slow manifold, the algorithm incorrectly identifies cubic nonlinearities, and fails to identify the slow manifold. This model was subsequently improved by Loiseau and Brunton [81] to incorporate energy-conserving constraints and to include higher-order terms to model the effect of truncated POD modes.

## 7.4.3 Application of time-delay embeddings for ROMs

Time-delay embedding for building ROMs can be used in a completely data-driven architecture where the governing equations are unknown, or for building a Koopman operator for a known governing evolution equation [22]. Indeed, one can use time-delay embedding with the SINDy architecture when short time-delay embeddings are used, or for producing a direct Koopman approximation when long time-delay embeddings are used. Champion et al. [35] highlight the various architectures possible. The short-time and long-time embedding possibilities are detailed here.

### 7.4.3.1 Short time-delay embedding

For a short time-delay embedding, the time-shifted data can provide a more accurate assessment of the true rank of the underlying system. Such time-delay embedding was used by Tu et al. [138] in order to ensure that the data were not rank-deficient. Indeed, without time-shifting the data, the DMD approximation does not capture the correct complex eigenvalue pairs associated with the periodic (Fourier) time dynamics.

Figure 7.6 shows the effects of the time-delay embedding as illustrated on the simple Van der Pol oscillator. In the top left panel of this figure, the singular values of $\mathbf{H}$ given by (7.32) for a short time-delay embedding is shown. Specifically, the data were delayed by five time steps. For this delay, the rank of the matrix $\mathbf{H}$ is dominated by two modes. The time dynamics of the first three modes are shown in the middle panel of the figure, illustrating the strongly nonlinear Van der Pol oscillations. A reduced model can then be constructed from the first two modes so that $\mathbf{\Psi}_{\text{TD}}$ spans a rank-2 subspace. Importantly, the dominant nonlinear time-series data can then be used with the SINDy architecture to discover the governing equations and build a dynamical ROM model.
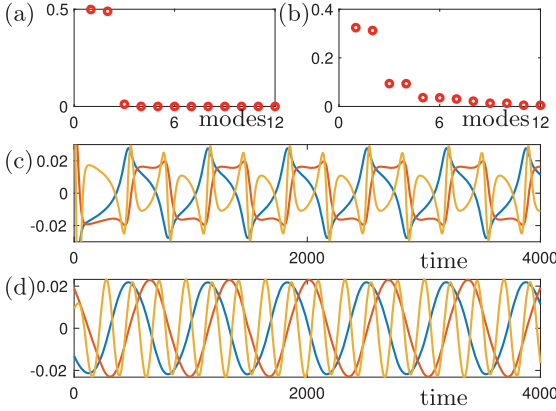
**Figure 7.6:** Time-delay embedding of the Van der Pol oscillator with time steps of 0.01. (a) With a short time-delay embedding of five time steps, the SVD produces a dominant low-rank (two-mode) truncation whose time-dynamic modes are illustrated in (c). (b) With a long time-delay embedding of several hundred time steps, the SVD produces a low-rank truncation of approximately a dozen modes whose time dynamic modes are illustrated in (d). Note that the short time-delay modes are strongly nonlinear oscillatory modes while the long time delay produces nearly perfect sinusoidal modes. Details can be found in Champion, Brunton, and Kutz [35].

### 7.4.3.2 Long time-delay embedding

For long time-delay embeddings, the nonlinear dynamics can be made to be approximately linear, thus providing an approximation to the Koopman operator and a linear ROM. The long time-delay embedding is especially useful in a data-driven architecture where the governing equations are unknown. Moreover, the time-delay embedding can significantly improve upon the DMD algorithm for producing an approximate dynamical system for forecasting.

Figure 7.6 shows the effects of the time-delay embedding as illustrated on the simple Van der Pol oscillator. In the top right panel of this figure, the singular values of **H** for a long-time delay embedding are shown. Specifically, the data were delayed by several hundred time steps which spanned more than a period of the nonlinear oscillations. Unlike the short time-delay embedding, the rank increases from two to about a dozen. The time dynamics of the first three of these dozen modes (i. e., the first three columns of the **V** matrix of (7.32)) are shown in the bottom panel. Note that the time modes with the long delay are now approximately sinusoidal, thus being ideal for a DMD/Koopman approximation. In this case, the SINDy architecture is unnecessary.

## 7.5 Conclusion and outlook

ROMs continue to play a critically enabling role in emulation and simulation strategies. Indeed, ROMs are making many intractable computations tractable by providing a surrogate model that can be computed at a fraction of the cost and with improved memory constraints. For emerging models in multiscale dynamical systems, such as in biology, atmospheric dynamics, and molecular dynamics simulations, ROMs provide a scalable mathematical framework, where it is possible to obtain accurate statistical estimates of the properties of the high-fidelity model from low-fidelity models.

Data-driven approaches to ROMs are also playing an increasingly important role in developing scalable and nonintrusive emulators. Thus the governing equations, which may be unknown or only partially known, can be approximated by a suite of emerging mathematical methods. Table 7.1 highlights the various methods that are available for producing data-driven ROMs. They are compared to the standard Galerkin-POD architecture. Importantly, for each ROM architecture, two things must be prescribed in the underlying separation of variable strategy (7.2): (i) the subspace on which the ROM is to be constructed, and (ii) the manner of extracting the dynamical evolution in this subspace. Of course, such reductions do not guarantee the construction of a stable ROM model, as recently highlighted by Carlberg et al. [34]. Thus for each ROM model strategy, care must be taken in order to produce a stable, low-rank emulator. Indeed, both POD-Galerkin and POD-DMD algorithms, for instance, must be modified in order to promote a stable time-stepping ROM.

If the governing evolution equations (7.1) are known, then a Galerkin-POD (or Petrov–Galerkin-POD) provides a simple projective method for producing a ROM. One can also use the DMD algorithm in this architecture (POD-DMD) for more rapid evaluation of the nonlinear terms. For unknown governing equations where the full state

**Table 7.1:** Model reduction algorithms and their subspaces. Included is one example reference highlighting the method.

| **Data-driven ROM algorithms** | |
| --- | --- |
| **ROM model $\mathbf{u}(\mathbf{x}, t) = \mathbf{\Psi}(\mathbf{x})\mathbf{a}(t)$** | |
| Galerkin-POD [14] | $\dot{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{L}\mathbf{\Psi}\mathbf{a} + \mathbf{\Psi}^T \mathbf{N}(\mathbf{\Psi}\mathbf{a})$ |
| DMD [78] | $\mathbf{u} = \mathbf{\Phi} \exp(\mathbf{\Omega}t)\mathbf{b}$ |
| POD-DMD [2] | $\dot{\mathbf{a}} = \mathbf{\Psi}^T \mathbf{L}\mathbf{\Psi}\mathbf{a} + \mathbf{\Psi}^T \mathbf{\Phi}_{\text{NL}} \exp(\mathbf{\Omega}_{\text{NL}}t)\mathbf{b}_{\text{NL}}$ |
| POD-SINDy [25] | $\dot{\mathbf{a}} = \mathbf{f}(\mathbf{a})$ dynamics on subspace $\mathbf{\Psi}$ |
| HAVOK-SINDy [35] | $\dot{\mathbf{a}} = \mathbf{f}(\mathbf{a})$ dynamics on subspace $\mathbf{\Psi}_{\text{TD}}$ (short delay) |
| HAVOK-Koopman [22] | $\dot{\mathbf{a}} = \mathbf{K}\mathbf{a}$ dynamics on subspace $\mathbf{\Psi}_{\text{TD}}$ (long delay) |
| **Basis elements (rank $r$)** | |
| POD modes | $\mathbf{X} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \ldots \ \mathbf{u}_m] = \mathbf{\Psi}\mathbf{\Sigma}\mathbf{V}^*$ |
| DMD modes | $\mathbf{X} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \ldots \ \mathbf{u}_m] = \mathbf{\Phi} \exp(\mathbf{\Omega}t)\mathbf{b}$ |
| nonlinear DMD modes | $\mathbf{N} = [\mathbf{N}_1 \ \mathbf{N}_2 \ \ldots \ \mathbf{N}_m] = \mathbf{\Phi}_{\text{NL}} \exp(\mathbf{\Omega}_{\text{NL}}t)\mathbf{b}_{\text{NL}}$ |
| Time-delay Koopman modes | $\mathbf{H} = \mathbf{\Psi}_{\text{TD}}\mathbf{\Sigma}\mathbf{V}^*$ |

space is sampled, DMD can be used to produce a low-rank, best-fit linear model for the dynamics. An alternative to DMD is the POD-SINDy algorithm, which discovers a low-rank, nonlinear dynamical system approximating the dynamics of the system. Time-delay embeddings allow for some flexibility in building a ROM depending upon the scenario. Time-delay embeddings also allow one to handle latent variables when the full state measurements are unknown or unavailable. For a long time-delay embedding with known or unknown governing equations, one can augment the DMD algorithm by producing a time-delay coordinate system which helps make the dynamics linearly dominant (HAVOK-Koopman). A short time delay can be used to determine the rank of the underlying dynamics and potentially build a SINDy model (HAVOK-SINDy). Alternatively, a long time-delay embedding can discover the intrinsic rank and linearize the dynamics in the time-delay coordinates. For more details on DMD, its variants, and its broad applications, please see [78]. For a broader overview of data-driven methods and machine learning applied to dynamics, please see [24].

The diversity of strategies is important in modern complex systems simulations where often the equations are only partially known, but where rich measurement data may be available. Thus data-driven strategies can bridge the gap between measurement space and model space. Table 7.1 gives a summary of the various current techniques. It is envisioned that refinement and innovations using the various strategies will greatly aid in modeling the challenge problems in many fields where high-dimensional, multiscale physics are prevalent. Figure 7.7 gives a summary of the decision space necessary when considering an appropriate ROM. One can either em-
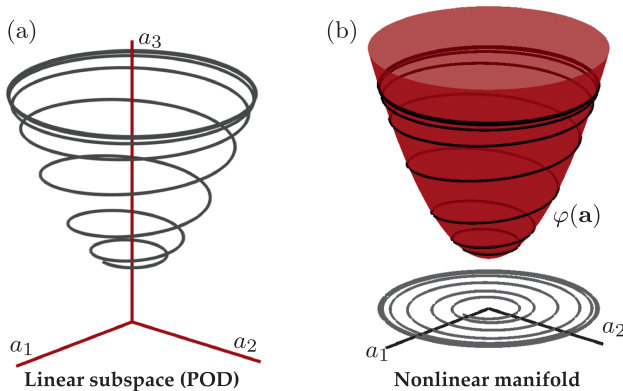


(a) $a_3$ $a_1$ $a_2$ **Linear subspace (POD)**

(b) $\varphi(\mathbf{a})$ $a_1$ $a_2$ **Nonlinear manifold**

**Figure 7.7:** Low-order modeling of fluid flows begins with an appropriate coordinate system that captures the few dominant flow mechanisms that are dynamically relevant. It is most common to embed high-dimensional fluid data in a linear subspace, for example using POD (a). However, for the flow past a cylinder, it is clear that the data live on a low-dimensional manifold in the embedding space (b). Both approaches have been explored extensively, for example by Noack et al. [98] and Loiseau et al. [82]. After an appropriate coordinate system is obtained, there are several choices for model construction.

bed in a linear space or in a nonlinear space (manifold), and then determine the appropriate nonlinear dynamics. This can be done in a variety of ways depending on wether the underlying governing equations are known, or if only measurement data are available.

# Bibliography

[1]   I. Abraham, G. De La Torre, and T. D. Murphey, Model-based control using Koopman operators. arXiv preprint arXiv:1709.01568, 2017.

[2]   A. Alla and J. N. Kutz, Nonlinear model order reduction via dynamic mode decomposition, *SIAM Journal on Scientific Computing*, **39** (5) (2017), B778–B796.

[3]   A. Alla and J. N. Kutz, Randomized model order reduction, *Advances in Computational Mathematics*, **45** (3) (2019), 1251–1271.

[4]   D. Amsallem and C. Farhat, Stabilization of projection-based reduced-order models, *International Journal for Numerical Methods in Engineering*, **91** (4) (2012), 358–377.

[5]   H. Arbabi and I. Mezić, Ergodic theory, dynamic mode decomposition and computation of spectral properties of the Koopman operator, *SIAM Journal on Applied Dynamical Systems*, **16** (4) (2017), 2096–2126.

[6]   T. Askham and J. N. Kutz, Variable projection methods for an optimized dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, **17** (1) (2018), 380–416.

[7]   Z. Bai, S. L. Brunton, B. W. Brunton, J. N. Kutz, E. Kaiser, A. Spohn, and B. R. Noack, Data-driven methods in fluid dynamics: Sparse classification from experimental data, in *Invited chapter for Whither Turbulence and Big Data in the 21st Century*, 2015.

[8]   Z. Bai, Z. Berger T. Wimalajeewa, G. Wang, M. Glauser, and P. K. Varshney, Low-dimensional approach for reconstruction of airfoil data via compressive sensing, *AIAA Journal*, **53** (4) (2014), 920–933.

[9]   M. J. Balajewicz, E. H. Dowell, and B. R. Noack, Low-dimensional modelling of high-reynolds-number shear flows incorporating constraints from the navier–stokes equation, *Journal of Fluid Mechanics*, **729** (2013), 285–308.

[10]  R. G. Baraniuk, Compressive sensing, *IEEE Signal Processing Magazine*, **24** (4) (2007), 118–120.

[11]  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. Mathématique*, **339** (9) (2004), 667–672.

[12]  P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira (eds.), *Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms*, De Gruyter, Berlin, 2020.

[13]  P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira (eds.), *Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms*, De Gruyter, Berlin, 2020.

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

[15]  P. Benner, C. Himpe, and T. Mitchell, On reduced input-output dynamic mode decomposition, *Advances in Computational Mathematics*, **44** (6) (1751–1768), 2018.

[16]  S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, John Wiley & Sons, 2013.

[17]   G. D. Birkhoff and B. O. Koopman, Recent contributions to the ergodic theory, *Proceedings of the National Academy of Sciences*, **18** (3) (1932), 279–282.

[18]   D. A. Bistrian and I. M. Navon, Randomized dynamic mode decomposition for nonintrusive reduced order modelling, *International Journal for Numerical Methods in Engineering*, **112** (1) (2017), 3–25.

[19]   J. Bongard and H. Lipson, Automated reverse engineering of nonlinear dynamical systems, *Proceedings of the National Academy of Sciences*, **104** (24) (2007), 9943–9948.

[20]   D. S. Broomhead and R. Jones, Time-series analysis, *Proceedings of the Royal Society of London A*, **423** (1864), 103–121.

[21]   B. W. Brunton, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Sparse sensor placement optimization for classification, *SIAM Journal on Applied Mathematics*, **76** (5) (2016), 2099–2122.

[22]   S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, Chaos as an intermittently forced linear system, *Nature Communications*, **8** (19) (2017), 1–9.

[23]   S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control, *PLoS ONE*, **11** (2) (2016), e0150171.

[24]   S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, 2018.

[25]   S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proceedings of the National Academy of Sciences*, **113** (15) (2016), 3932–3937.

[26]   S. L. Brunton, J. L. Proctor, and J. N. Kutz, Sparse identification of nonlinear dynamics with control (SINDYc), *IFAC-PapersOnLine*, **49** (18) (2016), 710–715.

[27]   S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz, Compressed sensing and dynamic mode decomposition, *Journal of Computational Dynamics*, **2** (2) (2015), 165–191.

[28]   S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz, Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems, *SIAM Journal on Applied Dynamical Systems*, **13** (4) (2014), 1716–1732.

[29]   M. Budišić, R. Mohr, and I. Mezić, Applied Koopmanism a), *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **22** (4): 047510, 2012.

[30]   E. J. Candès, Compressive sensing, in *Proceedings of the International Congress of Mathematics*, 2006.

[31]   E. J. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, *IEEE Transactions on Information Theory*, **52** (2) (2006), 489–509.

[32]   E. J. Candès, J. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements. *Communications in Pure and Applied Mathematics*, **8**(1207–1223), 59.

[33]   E. J. Candès and T. Tao, Near optimal signal recovery from random projections: Universal encoding strategies?, *IEEE Transactions on Information Theory*, **52** (12) (2006), 5406–5425.

[34]   K. Carlberg, M. Barone, and H. Antil, Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction, *Journal of Computational Physics*, **330** (2017), 693–734.

[35]   K. P. Champion, S. L. Brunton, and J. N. Kutz, Discovery of nonlinear multiscale systems: Sampling strategies and embeddings, *SIAM Journal on Applied Dynamical Systems*, **18** (1) (2019), 312–333.

[36]   R. Chartrand, Numerical differentiation of noisy, nonsmooth data, *ISRN Applied Mathematics*, **2011**, 2011.

[37]   S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM Journal on Scientific Computing*, **32** (5) (2010), 2737–2764.

[38] K. K. Chen, J. H. Tu, and C. W. Rowley, Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses, *Journal of Nonlinear Science*, **22** (6) (2012), 887–915.

[39] J. P. Crutchfield and B. S. McNamara, Equations of motion from a data series, *Complex Systems*, **1** (1987), 417–452.

[40] M. Dam, M. Brøns, J. J. Rasmussen, V. Naulin, and J. S. Hesthaven, Sparse identification of a predator-prey system from simulation data of a convection model, *Physics of Plasmas*, **24** (2) (2017), 022310.

[41] B. C. Daniels and I. Nemenman, Automated adaptive inference of phenomenological dynamical models, *Nature Communications*, **6**, 2015.

[42] B. C. Daniels and I. Nemenman, Efficient inference of parsimonious phenomenological models of cellular dynamics using s-systems and alternating regression, *PLoS ONE*, **10** (3) (2015), e0119821.

[43] S. Das and D. Giannakis, Delay-coordinate maps and the spectra of Koopman operators, *Journal of Statistical Physics*, (2019), 1–39.

[44] S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley, Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition, *Experiments in Fluids*, **57** (3) (2016), 1–19.

[45] D. L. Donoho, Compressed sensing, *IEEE Transactions on Information Theory*, **52** (4) (2006), 1289–1306.

[46] Z. Drmac and S. Gugercin, A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions, *SIAM Journal on Scientific Computing*, **38** (2) (2016), A631–A648.

[47] C. Eckart and G. Young, The approximation of one matrix by another of lower rank, *Psychometrika*, **1** (3) (1936), 211–218.

[48] N. B. Erichson and C. Donovan, Randomized low-rank dynamic mode decomposition for motion detection, *Computer Vision and Image Understanding*, **146** (2016), 40–50.

[49] N. B. Erichson, L. Mathelin, J. N. Kutz, and S. L. Brunton, Randomized dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, **18** (4) (2019), 1867–1891.

[50] R. Everson and L. Sirovich, Karhunen–Loeve procedure for gappy data, *Journal of the Optical Society of America. A, Online*, **12** (8) (1995), 1657–1664.

[51] B. Glaz, L. Liu, and P. P. Friedmann, Reduced-order nonlinear unsteady aerodynamic modeling using a surrogate-based recurrence framework, *AIAA Journal*, **48** (10) (2010), 2418–2429.

[52] G. H. Golub and R. J. LeVeque, Extensions and uses of the variable projection algorithm for solving nonlinear least squares problems, in *Proceedings of the 1979 Army Numerical Analsysis and Computers Conference*, 1979.

[53] G. H. Golub and V. Pereyra, The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate, *SIAM Journal on Numerical Analysis*, **10** (2) (1973), 413–432.

[54] F. Gueniat, L. Mathelin, and L. Pastur, A dynamic mode decomposition approach for large and arbitrarily sampled systems, *Physics of Fluids*, **27** (2) (2015), 025113.

[55] N. Halko, P.-G. Martinsson, and J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Review*, **53** (2) (2011), 217–288.

[56] S. Hanke, S. Peitz, O. Wallscheid, S. Klus, J. Böcker, and M. Dellnitz, Koopman operator based finite-set model predictive control for electrical drives. arXiv preprint arXiv:1804.00854, 2018.

[57] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, vol. 2, Springer, 2009.

[58] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, De-biasing the dynamic mode decomposition for applied Koopman spectral analysis, *Theoretical and Computational Fluid Dynamics*, **31** (4) (2017), 349–368.

[59] M. Hinze and S. Volkwein, Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control, in *Dimension Reduction of Large-Scale Systems*, pp. 261–306, Springer, 2005.

[60] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, 2nd paperback edition, Cambridge University Press, Cambridge, 2012.

[61] P. Holmes and J. Guckenheimer, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Applied Mathematical Sciences, vol. 42, Springer-Verlag, Berlin, Heidelberg, 1983.

[62] C. P. Jackson, A finite-element study of the onset of vortex shedding in flow past variously shaped bodies, *Journal of Fluid Mechanics*, **182** (1987), 23–45.

[63] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, Springer, 2013.

[64] J. N. Juang and R. S. Pappa, An eigensystem realization algorithm for modal parameter identification and model reduction, *Journal of Guidance, Control, and Dynamics*, **8** (5) (1985), 620–627.

[65] E. Kaiser, J. N. Kutz, and S. L. Brunton, Data-driven discovery of Koopman eigenfunctions for control. arXiv preprint arXiv:1707.01146, 2017.

[66] E. Kaiser, J. N. Kutz, and S. L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, *Proceedings of the Royal Society A*, **474** (2219) (2018), 20180335.

[67] I. Kalashnikova, M. F. Barone, S. Arunajatesan, and B. G. van Bloemen Waanders, Construction of energy-stable projection-based reduced order models, *Applied Mathematics and Computation*, **249** (2014), 569–596.

[68] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz, Time-delay observables for Koopman: Theory and applications, *SIAM Journal on Applied Dynamical Systems*, **19** (2) (2020), 886–917.

[69] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos, Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis, *Communications in Mathematical Sciences*, **1** (4) (2003), 715–762.

[70] S. Klus, P. Koltai, and C. Schütte, On the numerical approximation of the Perron-Frobenius and Koopman operator. arXiv preprint arXiv:1512.05997, 2015.

[71] B. O. Koopman, Hamiltonian systems and transformation in Hilbert space, *Proceedings of the National Academy of Sciences*, **17** (5) (1931), 315–318.

[72] B. O. Koopman and J. v. Neumann, Dynamical systems of continuous spectra, *Proceedings of the National Academy of Sciences of the United States of America*, **18** (3) (1932), 255.

[73] M. Korda and I. Mezić, Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control, *Automatica*, **93** (2018), 149–160.

[74] M. Korda and I. Mezić, On convergence of extended dynamic mode decomposition to the Koopman operator, *Journal of Nonlinear Science*, **28** (2) (2018), 687–710.

[75] K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems, *Numerische Mathematik*, **90** (1) (2001), 117–148.

[76] K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, *SIAM Journal on Numerical Analysis*, **40** (2) (2002), 492–515.

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

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

[79]    J. N. Kutz, X. Fu, and S. L. Brunton, Multiresolution dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, **15** (2) (2016), 713–735.

[80]    J. N. Kutz, J. L. Proctor, and S. L. Brunton, Applied Koopman theory for partial differential equations and data-driven modeling of spatio-temporal systems, *Complexity*, (2018), 2018.

[81]    J.-C. Loiseau and S. L. Brunton, Constrained sparse Galerkin regression, *Journal of Fluid Mechanics*, **838** (2018), 42–67.

[82]    J.-C. Loiseau, B. R. Noack, and S. L. Brunton, Sparse reduced-order modeling: sensor-based dynamics to full-state estimation, *Journal of Fluid Mechanics*, **844** (2018), 459–490.

[83]    B. Lusch, J. N. Kutz, and S. L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, *Nature Communications*, **9** (1) (2018), 4950.

[84]    A. Mackey, H. Schaeffer, and S. Osher, On the compressive spectral method, *Multiscale Modeling & Simulation*, **12** (4) (2014), 1800–1827.

[85]    M. W. Mahoney et al., Randomized algorithms for matrices and data, *Foundations and Trends in Machine Learning*, **3** (2) (2011), 123–224.

[86]    A. J. Majda and J. Harlim, Physics constrained nonlinear regression models for time series, *Nonlinearity*, **26** (1) (2012), 201.

[87]    A. J. Majda and Y. Lee, Conceptual dynamical models for turbulence, *Proceedings of the National Academy of Sciences*, **111** (18) (2014), 6548–6553.

[88]    N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Inferring biological networks by sparse identification of nonlinear dynamics, *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, **2** (1) (2016), 52–63.

[89]    N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, Model selection for dynamical systems via sparse regression and information criteria, *Proceedings of the Royal Society A*, **473** (2204) (2017), 1–16.

[90]    K. Manohar, S. L. Brunton, and J. N. Kutz, Environmental identification in flight using sparse approximation of wing strain, *Journal of Fluids and Structures*, **70** (2017), 162–180.

[91]    K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, Data-driven sparse sensor placement, *IEEE Control Systems Magazine*, **38** (3) (2018), 63–86.

[92]    A. Mardt, L. Pasquali, H. Wu, and F. Noé, VAMPnets: Deep learning of molecular kinetics, *Nature Communications*, **9** (5) (2018).

[93]    I. Mezić, Spectral properties of dynamical systems, model reduction and decompositions, *Nonlinear Dynamics*, **41** (1-3) (2005), 309–325.

[94]    I. Mezic, Analysis of fluid flows via spectral properties of the Koopman operator, *Annual Review of Fluid Mechanics*, **45** (2013), 357–378.

[95]    I. Mezić and A. Banaszuk, Comparison of systems with complex behavior, *Physica D: Nonlinear Phenomena*, **197** (1) (2004), 101–133.

[96]    L. Mirsky, Symmetric gauge functions and unitarily invariant norms, *Quarterly Journal of Mathematics*, **11** (1) (1960), 50–59.

[97]    C. C. Moore, Ergodic theorem, ergodic theory, and statistical mechanics, *Proceedings of the National Academy of Sciences*, **112** (7) (2015), 1907–1911.

[98]    B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, *Journal of Fluid Mechanics*, **497** (2003), 335–363.

[99]    F. Noé and F. Nuske, A variational approach to modeling slow processes in stochastic dynamical systems, *Multiscale Modeling & Simulation*, **11** (2) (2013), 635–655.

[100]   F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, Variational approach to molecular kinetics, *Journal of Chemical Theory and Computation*, **10** (4) (2014), 1739–1752.

[101]  S. E. Otto and C. W. Rowley, Linearly recurrent autoencoder networks for learning dynamics, *SIAM Journal on Applied Dynamical Systems*, **18** (1) (2019), 558–593.

[102]  V. Ozoliņš, R. Lai, R. Caflisch, and S. Osher, Compressed modes for variational problems in mathematics and physics, *Proceedings of the National Academy of Sciences*, **110** (46) (2013), 18368–18373.

[103]  S. Peitz and S. Klus, Koopman operator-based model reduction for switched-system control of pdes, *Automatica*, **106** (2019), 184–191.

[104]  V. Pereyra and G. Scherer, *Exponential Data Fitting and Its Applications*, Bentham Science Publishers, 2010.

[105]  J. L. Proctor, S. L. Brunton, B. W. Brunton, and J. N. Kutz, Exploiting sparsity and equation-free architectures in complex systems (invited review), *The European Physical Journal Special Topics*, **223** (13) (2014), 2665–2684.

[106]  J. L. Proctor, S. L. Brunton, and J. N. Kutz, Dynamic mode decomposition with control, *SIAM Journal on Applied Dynamical Systems*, **15** (1) (2016), 142–161.

[107]  J. L. Proctor, S. L. Brunton, and J. N. Kutz, Generalizing Koopman theory to allow for inputs and control, *SIAM Journal on Applied Dynamical Systems*, **17** (1) (2018), 909–930.

[108]  A. J. Roberts, *Model Emergent Dynamics in Complex Systems*, SIAM, 2014.

[109]  C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, Spectral analysis of nonlinear flows, *Journal of Fluid Mechanics*, **645** (2009), 115–127.

[110]  C. W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, **15** (03) (2005), 997–1013.

[111]  S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Data-driven discovery of partial differential equations, *Science Advances*, **3** (2017), e1602614.

[112]  D. Ruelle and F. Takens, On the nature of turbulence, *Communications in Mathematical Physics*, **20** (1971), 167–192.

[113]  T. P. Sapsis and A. J. Majda, Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems, *Proceedings of the National Academy of Sciences*, **110** (34) (2013), 13705–13710.

[114]  H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher, Sparse dynamics for partial differential equations, *Proceedings of the National Academy of Sciences of the United States of America*, **110** (17) (2013), 6634–6639.

[115]  H. Schaeffer, Learning partial differential equations via data discovery and sparse optimization, *Proceedings of the Royal Society A. Mathematical, Physical and Engineering Sciences*, **473** (2197) (2017), 20160446.

[116]  H. Schaeffer and S. G. McCalla, Sparse model selection via integral terms, *Physical Review E*, **96** (2) (2017), 023302.

[117]  P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *Journal of Fluid Mechanics*, **656** (August 2010), 5–28.

[118]  E. Schmidt, Zur theorie der linearen und nichtlinearen integralgleichungen. 1. teil: Entwicklung willkürlicher funktionen nach systemen vorgeschriebener, *Mathematische Annalen*, **63** (1907), 433–476.

[119]  M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, *Science*, **324** (5923) (2009), 81–85.

[120]  M. D. Schmidt, R. R. Vallabhajosyula, J. W. Jenkins, J. E. Hood, A. S. Soni, J. P. Wikswo, and H. Lipson, Automated refinement and inference of analytical models for metabolic networks, *Physical Biology*, **8** (5) (2011), 055011.

[121]  O. Semeraro, F. Lusseyran, L. Pastur, and P. Jordan, Qualitative dynamics of wave packets in turbulent jets, *Physical Review Fluids*, **2** (9) (2017), 094605.

[122] L. Sirovich, Method of snapshots, *Quarterly of Applied Mathematics*, **45** (3) (1987), 561–571.

[123] M. Sorokina, S. Sygletos, and S. Turitsyn, Sparse identification for nonlinear optical communication systems: SINO method, *Optics Express*, **24** (26) (2016), 30433–30443.

[124] G. Sugihara, R. May, H. Ye, C.-h. Hsieh, E. Deyle, M. Fogarty, and S. Munch, Detecting causality in complex ecosystems, *Science*, **338** (6106) (2012), 496–500.

[125] A. Surana, Koopman operator based observer synthesis for control-affine nonlinear systems, in *55th IEEE Conference on Decision and Control (CDC)*, pp. 6492–6499, 2016.

[126] A. Surana and A. Banaszuk, Linear observer synthesis for nonlinear systems using Koopman operator framework, *IFAC-PapersOnLine*, **49** (18) (2016), 716–723.

[127] Y. Susuki and I. Mezić, A prony approximation of Koopman mode decomposition, in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. pp. 7022–7027, IEEE, 2015.

[128] A. Svenkeson, B. Glaz, S. Stanton, and B. J. West, Spectral decomposition of nonlinear systems with memory, *Physical Review E*, **93** (Feb 2016), 022211.

[129] K. Taira, S. L. Brunton, S. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, Modal analysis of fluid flows: An overview, *AIAA Journal*, **55** (12) (2017), 4013–4041.

[130] N. Takeishi, Y. Kawahara, and T. Yairi, Learning Koopman invariant subspaces for dynamic mode decomposition, in *Advances in Neural Information Processing Systems*, pp. 1130–1140, 2017.

[131] F. Takens, Detecting strange attractors in turbulence, *Lecture Notes in Mathematics*, **898** (1981), 366–381.

[132] R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society, Series B, Methodological*, (1996), 267–288.

[133] A. Towne, O. T. Schmidt, and T. Colonius, Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis, *Journal of Fluid Mechanics*, **847** (2018), 821–867.

[134] G. Tran and R. Ward, Exact recovery of chaotic systems from highly corrupted data, *Multiscale Modeling & Simulation*, **15** (3) (2017), 1108–1129.

[135] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, vol. 50, Siam, (1997).

[136] F. Tröltzsch and S. Volkwein, Pod a-posteriori error estimates for linear-quadratic optimal control problems, *Computational Optimization and Applications*, **44** (1) (2009), 83.

[137] J. A. Tropp and A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Transactions on Information Theory*, **53** (12) (2007), 4655–4666.

[138] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, On dynamic mode decomposition: theory and applications, *Journal of Computational Dynamics*, **1** (2) (2014), 391–421.

[139] J. H. Tu, C. W. Rowley, J. N. Kutz, and J. K. Shang, Spectral analysis of fluid flows using sub-nyquist-rate piv data, *Experiments in Fluids*, **55** (9) (2014), 1805.

[140] W. X. Wang, R. Yang, Y. C. Lai, V. Kovanis, and C. Grebogi, Predicting catastrophes in nonlinear dynamical systems by compressive sensing, *Physical Review Letters*, **106** (2011), 154101–1–154101–4.

[141] C. Wehmeyer and F. Noé, Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics, *Journal of Chemical Physics*, **148** (241703) (2018), 1–9.

[142] K. Willcox, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, *Computers & Fluids*, **35** (2) (2006), 208–226.

[143] K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA Journal*, **40** (11) (2002), 2323–2330.

[144] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, A data-driven approximation of the Koopman operator: extending dynamic mode decomposition, *Journal of Nonlinear Science*, **6** (2015), 1307–1346.

[145] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, A kernel approach to data-driven Koopman spectral analysis, *Journal of Computational Dynamics*, **2** (2) (2015), 247–265.

[146] M. O. Williams, P. J. Schmid, and J. N. Kutz, Hybrid reduced-order integration with proper orthogonal decomposition and dynamic mode decomposition, *Multiscale Modeling & Simulation*, **11** (2) (2013), 522–544.

[147] C. Yao and E. M. Bollt, Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems, *Physica D: Nonlinear Phenomena*, **227** (1) (2007), 78–99.

[148] H. Ye, R. J. Beamish, S. M. Glaser, S. C. H. Grant, C.-h. Hsieh, L. J. Richards, J. T. Schnute, and G. Sugihara, Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling, *Proceedings of the National Academy of Sciences*, **112** (13) (2015), E1569–E1576.

[149] E. Yeung, S. Kundu, and N. Hodas, Learning deep neural network representations for Koopman operators of nonlinear dynamical systems, in *2019 American Control Conference (ACC)*. pp. 4832–4839, IEEE, 2019.

[150] B. Yildirim, C. Chryssostomidis, and G. E. Karniadakis, Efficient sensor placement for ocean measurements using low-dimensional concepts, *Ocean Modelling*, **27** (3) (2009), 160–173.

[151] Z. Zebib, Stability of viscous flow past a circular cylinder, *Journal of Engineering Mathematics*, **21** (1987), 155–165.

[152] L. Zhang and H. Schaeffer, On the convergence of the SINDy algorithm, *Multiscale Modeling & Simulation*, **17** (3) (2019), 948–972.

[153] W. Zhang, B. Wang, Z. Ye, and J. Quan, Efficient method for limit cycle flutter analysis based on nonlinear aerodynamic reduced-order models, *AIAA Journal*, **50** (5) (2012), 1019–1028.

[154] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, A unified framework for sparse relaxed regularized regression: Sr3, *IEEE Access*, **7** (2018), 1404–1423.

[155] H. Zou and T. Hastie, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society, Series B, Statistical Methodology*, **67** (2) (2005), 301–320.