J Nonlinear Sci (2015) 25:1307–1346
DOI 10.1007/s00332-015-9258-5

Journal of
Nonlinear
Science

A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition

Matthew O. Williams[1] · **Ioannis G. Kevrekidis**[2] · **Clarence W. Rowley**[3]

Received: 19 August 2014 / Accepted: 20 May 2015 / Published online: 5 June 2015

Abstract The Koopman operator is a *linear* but infinite-dimensional operator that governs the evolution of scalar observables defined on the state space of an autonomous dynamical system and is a powerful tool for the analysis and decomposition of nonlinear dynamical systems. In this manuscript, we present a data-driven method for approximating the leading *eigenvalues, eigenfunctions, and modes* of the Koopman operator. The method requires a data set of snapshot pairs and a dictionary of scalar observables, but does not require explicit governing equations or interaction with a "black box" integrator. We will show that this approach is, in effect, an extension of dynamic mode decomposition (DMD), which has been used to approximate the Koopman eigenvalues and modes. Furthermore, if the data provided to the method are generated by a Markov process instead of a deterministic dynamical system, the algorithm approximates the eigenfunctions of the Kolmogorov backward equation, which could be considered as the "stochastic Koopman operator" (Mezic in Nonlinear

Communicated by Oliver Junge.

✉ Matthew O. Williams
mow2@princeton.edu

Ioannis G. Kevrekidis
krevrekidis@princeton.edu

Clarence W. Rowley
cwrowley@princeton.edu

[1] Program in Applied and Computational Mathematics (PACM), Princeton University, Princeton, NJ 08544, USA

[2] Chemical and Biological Engineering Department & PACM, Princeton University, Princeton, NJ 08544, USA

[3] Mechanical and Aerospace Engineering Department, Princeton University, Princeton, NJ 08544, USA

Dynamics 41(1–3): 309–325, 2005). Finally, four illustrative examples are presented: two that highlight the quantitative performance of the method when presented with either deterministic or stochastic data and two that show potential applications of the Koopman eigenfunctions.

Keywords Data mining · Koopman spectral analysis · Set oriented methods · Spectral methods · Reduced order models

Mathematics Subject Classification Primary: 65P99 · 37M25 · Secondary: 47B33

1 Introduction

In many mathematical and engineering applications, a phenomenon of interest can be summarized in different ways. For instance, to describe the state of a two-dimensional incompressible fluid flow, one can either record velocity and pressure fields or stream function and vorticity (Hirsch 2007). Furthermore, these states can often be *approximated* using a low-dimensional set of proper orthogonal decomposition (POD) modes (Holmes et al. 1998), a set of dynamic modes (Schmid 2010; Schmid et al. 2012), or a finite collection of Lagrangian particles (Monaghan 1992). A mathematical example is the linear time-invariant (LTI) system provided by $\boldsymbol{x}(n+1) = \boldsymbol{A}\boldsymbol{x}(n)$, where $\boldsymbol{x}(n)$ is the system state at the nth timestep. Written as such, the evolution of $\boldsymbol{x}$ is governed by the eigenvalues of $\boldsymbol{A}$. One could also consider the invertible but nonlinear change in variables, $\boldsymbol{z}(n) = \boldsymbol{T}(\boldsymbol{x}(n))$, which generates a nonlinear evolution law for $\boldsymbol{z}$. Both approaches (i.e., $\boldsymbol{x}$ or $\boldsymbol{z}$) describe the same fundamental behavior, yet one description may be preferable to others. For example, solving an LTI system is almost certainly preferable to evolving a nonlinear system from a computational standpoint.

In general, one measures (or computes) the state of a system using a set of scalar *observables*, which are functions defined on state space, and watches how the values of these functions evolve in time. As we will show shortly, one can write an evolution law for the dynamics of this set of observables, and if they happen to be “rich enough,” reconstruct the original system state from the observations. Because the properties of this new dynamical system depend on our choice of variables (observables), it would be highly desirable if one could find a set of observables whose dynamics appear to be governed by a linear evolution law. If such a set could be identified, the dynamics would be completely determined by the spectrum of the evolution operator. Furthermore, this could enable the simple yet effective algorithms designed for linear systems, for example controller design (Todorov 2007; Stengel 2012) or stability analysis (Mauroy and Mezic 2013; Lehoucq et al. 1998), to be applied to nonlinear systems.

Mathematically, the evolution of observables of the system state is governed by the *Koopman operator* (Koopman and Neumann 1932; Koopman 1931; Budišić et al. 2012; Rowley et al. 2009), which is a *linear but infinite-dimensional* operator that is defined for a given dynamical system. Of particular interest here is the “slow” subspace of the Koopman operator, which is the span of the eigenfunctions associated with eigenvalues near the unit circle in discrete time (or near the imaginary axis in continuous time). These eigenvalues and eigenfunctions capture the long-term dynamics of observables that appear after the fast transients have subsided and could serve as a low-dimensional approximation of the otherwise infinite-dimensional operator when a spectral gap, which clearly delineates the "fast" and "slow" temporal dynamics, is present. In addition to the eigenvalues and eigenfunctions, the final element of Koopman spectral analysis is the set of *Koopman modes* for the *full-state* observable (i.e., the identity operator) (Budišić et al. 2012; Rowley et al. 2009) which are vectors that enable us to reconstruct the *state of the system* as a linear combination of the Koopman eigenfunctions. Overall, the "tuples" of Koopman eigenfunctions, eigenvalues, and modes enable us to: (a) transform state space so that the dynamics appear to be linear, (b) determine the temporal dynamics of the linear system, and (c) reconstruct the state of the original system from our new linear representation. In principle, this framework is quite broadly applicable and useful even for problems with multiple attractors that cannot be accurately approximated using models based on local linearization.

There are several algorithms in the literature that can computationally approximate subsets of these quantities. Three examples are numerical implementations of generalized Laplace analysis (GLA) (Budišić et al. 2012; Mauroy and Mezić 2012; Mauroy et al. 2013), the Ulam Galerkin method (Froyland et al. 2014; Bollt and Santitissadeekorn 2013), and dynamic mode decomposition (DMD) (Schmid 2010; Tu et al. 2014; Rowley et al. 2009). None of these techniques require explicit governing equations, so all, in principle, can be applied directly to data. GLA can approximate both the Koopman modes and eigenfunctions, but it requires knowledge of the eigenvalues to do so (Budišić et al. 2012; Mauroy and Mezić 2012; Mauroy et al. 2013). The Ulam Galerkin method has been used to approximate the eigenfunctions and eigenvalues (Froyland et al. 2014), though it is more frequently used to generate finite-dimensional approximations of the Perron–Frobenius operator, which is the adjoint of the Koopman operator. Finally, DMD has been used to approximate the Koopman modes and eigenvalues (Rowley et al. 2009; Tu et al. 2014), but not the Koopman eigenfunctions.

Even in pairs instead of triplets, approximations of these quantities are useful. DMD and its variants (Wynn et al. 2013; Chen et al. 2012; Jovanović et al. 2014) have been successfully used to analyze nonlinear fluid flows using data from both experiments and computation (Schmid 2010; Muld et al. 2012; Seena and Sung 2011). GLA and similar methods have been applied to extract meaningful spatio-temporal structures using sensor data from buildings and power systems (Eisenhower et al. 2010; Susuki and Mezić 2011, 2012, 2014). Finally, the Ulam Galerkin method has been used to identify coherent structures and almost invariant sets (Froyland et al. 2007; Froyland and Padberg 2009; Froyland 2005) based on the singular value decomposition of (a slight modification of) the Perron–Frobenius operator.

In this manuscript, we present a data-driven method that approximates the leading Koopman *eigenfunctions, eigenvalues,* **and** *modes* from a data set of successive "snapshot" pairs and a dictionary of observables that spans a subspace of the space of scalar observables. There are many possible ways to choose this dictionary, and it could be comprised of polynomials, Fourier modes, spectral elements, or other sets of functions of the full-state observable. We will argue that this approach is an extension of DMD that can produce better approximations of the Koopman eigenfunctions; as such, we refer to it as *extended dynamic mode decomposition* (EDMD). One regime where the behavior of both EDMD and DMD can be formally analyzed and contrasted is in the limit of large data. Following the definition of DMD in Tu et al. (2014), we consider the case where this large data set consists of a distribution of snapshot pairs rather than a single time series. In this regime, we will show that the numerical approximation of the Koopman eigenfunctions generated by EDMD converges to the numerical approximation we would obtain from a Galerkin method (Boyd 2013) in that the residual is orthogonal to the subspace spanned by the elements of the dictionary. With finite amounts of data, we will demonstrate the effectiveness of EDMD on two deterministic examples: one that highlights the quantitative accuracy of the method and other a more practical application.

Because EDMD is an entirely data-driven procedure, it can also be applied to data from stochastic systems without any algorithmic changes. If the underlying system is a Markov process, we will show that EDMD approximates the eigenfunctions of the Kolmogorov backward equation (Givon et al. 2004; Bagheri 2014) which has been called the *stochastic Koopman* operator (SKO) (Mezić 2005). Once again, we will demonstrate the effectiveness of the EDMD procedure when the amount of data is limited, by applying it to two stochastic examples: the first to test the accuracy of the method and the second to highlight a potential application of EDMD as a nonlinear manifold learning technique. In the latter example, we highlight two forms of model reduction: reduction that occurs when the *dynamics of the system state* are constrained to a low-dimensional manifold and reduction that occurs when the evolution of *statistical moments* of the stochastic dynamical system are effectively low dimensional.

In the remainder of the manuscript, we will detail the EDMD algorithm and show (when mathematically possible) or demonstrate through examples that it accurately approximates the leading Koopman eigenfunctions, eigenvalues, and modes for both deterministic and stochastic sets of data. In particular, in Sect. 2, the EDMD algorithm will be presented, and we will prove that it converges to a Galerkin approximation of the Koopman operator given a sufficiently large amount of data. In Sect. 3, we detail three choices of dictionary that we have found to be effective in a broad set of applications. In Sect. 4, we will demonstrate that the EDMD approximation can be accurate even with finite amounts of data and can yield useful parameterizations of common dynamical structures in problems with multiple basins of attraction *when the underlying system is deterministic*. In Sect. 5, we experiment by applying EDMD to stochastic data and show it approximates the eigenfunctions of the SKO for Markov processes. Though the interpretation of the eigenfunctions now differs, we demonstrate that they can still be used to accomplish useful tasks such as the parameterization of nonlinear manifolds. Finally, some brief concluding remarks are given in Sect. 6.

2 Dynamic Mode Decomposition and the Koopman Operator

Our ambition in this section is to establish the connection between the Koopman operator and what we call EDMD. To accomplish this, we will define the Koopman operator in Sect. 2.1. Using this definition, we will outline the EDMD algorithm in

Sect. 2.2 and then show how it can be used to approximate the Koopman eigenvalues, eigenfunctions, and modes. Next, in Sect. 2.3, we will prove that the EDMD method almost surely converges to a Galerkin method in the limit of large data. Finally, in Sect. 2.4, we will highlight the connection between the EDMD algorithm and standard DMD.

2.1 The Koopman Operator

Because the Koopman operator is central to all that follows, we will define it along with the properties relevant to EDMD in this subsection. No new mathematical results are presented here; our only objective is to include for completeness the terms and definitions we will require later in the paper. In this manuscript, we consider the case where the dynamics of interest are governed by an autonomous, discrete-time dynamical system $(\mathcal{M}, n, \boldsymbol{F})$, where $\mathcal{M} \subseteq \mathbb{R}^N$ is the state space, $n \in \mathbb{Z}$ is (discrete) time, and $\boldsymbol{F} : \mathcal{M} \to \mathcal{M}$ is the evolution operator. Unlike $\boldsymbol{F}$, which acts on $\boldsymbol{x} \in \mathcal{M}$, the Koopman operator, $\mathcal{K}$, acts on functions of state space, $\psi \in \mathcal{F}$ with $\psi : \mathcal{M} \to \mathbb{C}$. The action of the Koopman operator is

$$\mathcal{K}\psi = \psi \circ \boldsymbol{F}, \tag{1}$$

where $\circ$ denotes the composition of ψ with $\boldsymbol{F}$. We stress once again that *the Koopman operator maps functions of state space to functions of state space* and not states to states (Koopman and Neumann 1932; Koopman 1931; Budišić et al. 2012).

In essence, the Koopman operator defines a new dynamical system, $(\mathcal{F}, n, \mathcal{K})$, that governs the evolution of *observables*, $\psi \in \mathcal{F}$, in discrete time. In what follows, we assume that $\mathcal{F} = L^2(\mathcal{M}, \rho)$, where ρ is a positive, single-valued analytic function with $\|\rho\|_{\mathcal{M}} = \int_{\mathcal{M}} \rho(\boldsymbol{x})\ d\boldsymbol{x} = 1$, but not necessarily an invariant measure of the underlying dynamical system. This assumption, which has been made before in the literature (Budišić et al. 2012; Koopman 1931), is required so that the inner products in the Galerkin-like method we will present can be taken. Because it acts on functions, $\mathcal{K}$ is infinite dimensional even when $\boldsymbol{F}$ is finite dimensional, but provided that $\mathcal{F}$ is a vector space, it is also *linear even when* $\boldsymbol{F}$ *is nonlinear*. The infinite-dimensional nature of the Koopman operator is potentially problematic, but if it can, practically, be truncated without too great a loss of accuracy (e.g., if the system has multiple timescales), then the result would be a linear and finite-dimensional approximation. Therefore, the promise of the Koopman approach is to take the tools developed for linear systems and apply them to the dynamical system defined by the Koopman operator, thus obtaining a linear approximation of a nonlinear system without directly linearizing around a particular fixed point.

The dynamical system defined by $\boldsymbol{F}$ and the one defined by $\mathcal{K}$ are two different parameterizations of the same fundamental behavior. The link between these parameterizations is the identity operator or "full-state observable," $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$ and $\{(\mu_k, \varphi_k, \boldsymbol{v}_k)\}_{k=1}^{N_K}$, the set of N_K "tuples" of Koopman eigenvalues, eigenfunctions, and modes required to reconstruct the full state. Note that N_K could (and often will) be infinite. Although $\boldsymbol{g}$ is a *vector-valued observable*, each component of it is a scalar-valued observable, i.e., $g_i \in \mathcal{F}$ where g_i is the ith component of $\boldsymbol{g}$. Assuming g_i is in the span of our set of N_K eigenfunctions, $g_i = \sum_{k=1}^{N_K} v_{ik}\varphi_k$ with $v_{kj} \in \mathbb{C}$. Then, $\boldsymbol{g}$ can be obtained by "stacking" these weights into vectors (i.e., $\boldsymbol{v}_j = [v_{1j}, v_{2j}, \ldots, v_{Nj}]^T$). As a result,

$$\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{x}) = \sum_{k=1}^{N_K} \boldsymbol{v}_k \varphi_k(\boldsymbol{x}), \tag{2}$$

where $\boldsymbol{v}_k$ is the kth *Koopman mode* and φ_k is the kth *Koopman eigenfunction*. In doing this, we have assumed that each of the scalar observables that comprise $\boldsymbol{g}$ are in the subspace of $\mathcal{F}$ spanned by our N_K eigenfunctions, but we have not assumed that the eigenfunctions form a basis for $\mathcal{F}$. In some applications, the Koopman operator will have a continuous spectrum, which as shown in Mezić (2005) would introduce an additional term in (2). The system state at future times can be obtained either by directly evolving $\boldsymbol{x}$ or by evolving the full-state observable through Koopman:

$$\boldsymbol{F}(\boldsymbol{x}) = (\mathcal{K}\boldsymbol{g})(\boldsymbol{x}) = \sum_{k=1}^{N_K} \boldsymbol{v}_k (\mathcal{K}\varphi_k)(\boldsymbol{x}) = \sum_{k=1}^{N_K} \mu_k \boldsymbol{v}_k \varphi_k(\boldsymbol{x}). \tag{3}$$

This representation of $\boldsymbol{F}(\boldsymbol{x})$ is particularly advantageous because the dynamics associated with each eigenfunction are determined by its corresponding eigenvalue.

Figure 1 shows a commutative diagram that acts as a visual summary of this section. The top row shows the direct evolution of states, $\boldsymbol{x} \in \mathcal{M}$, governed by $\boldsymbol{F}$; the bottom row shows the evolution of observables, $\psi \in \mathcal{F}$, governed by the Koopman operator. Although $\boldsymbol{F}$ and $\mathcal{K}$ act on different spaces, they encapsulate the same dynamics. For example, once given a state $\boldsymbol{x}$, to compute $(\mathcal{K}\psi)(\boldsymbol{x})$ one could either take the observable ψ, apply $\mathcal{K}$, and evaluate it at $\boldsymbol{x}$ (the bottom route), or use $\boldsymbol{F}$ to compute $\boldsymbol{F}(\boldsymbol{x})$ and then evaluate ψ at this updated position (the top route). Similarly, to compute $\boldsymbol{F}(\boldsymbol{x})$, one could either apply $\boldsymbol{F}$ to $\boldsymbol{x}$ (the top route) or apply $\mathcal{K}$ to the full-state observable and evaluate $(\mathcal{K}\boldsymbol{g})(\boldsymbol{x})$ (the bottom route). As a result, one can either choose to work with a finite-dimensional, nonlinear system or an infinite-dimensional, linear system depending upon which "path" is simpler/more useful for a given problem.

We should note that for autonomous systems of ordinary differential equations (ODEs), such as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}), \tag{4}$$

there is a semigroup of Koopman operators, $\mathcal{K}_{\Delta t}$, each of which is associated with the flow map for the time interval Δt (Budišić et al. 2012; Santitissadeekorn and Bollt 2007). The infinitesimal generator of this semigroup is

$$\hat{\mathcal{K}}\psi = \boldsymbol{f} \cdot \nabla\psi, \tag{5}$$

which we will refer to as the *continuous-time* Koopman operator. If φ_k is the kth eigenfunction of $\hat{\mathcal{K}}$ associated with the eigenvalue λ_k, then the future value of some vector-valued observable $\boldsymbol{g}$ can be written as

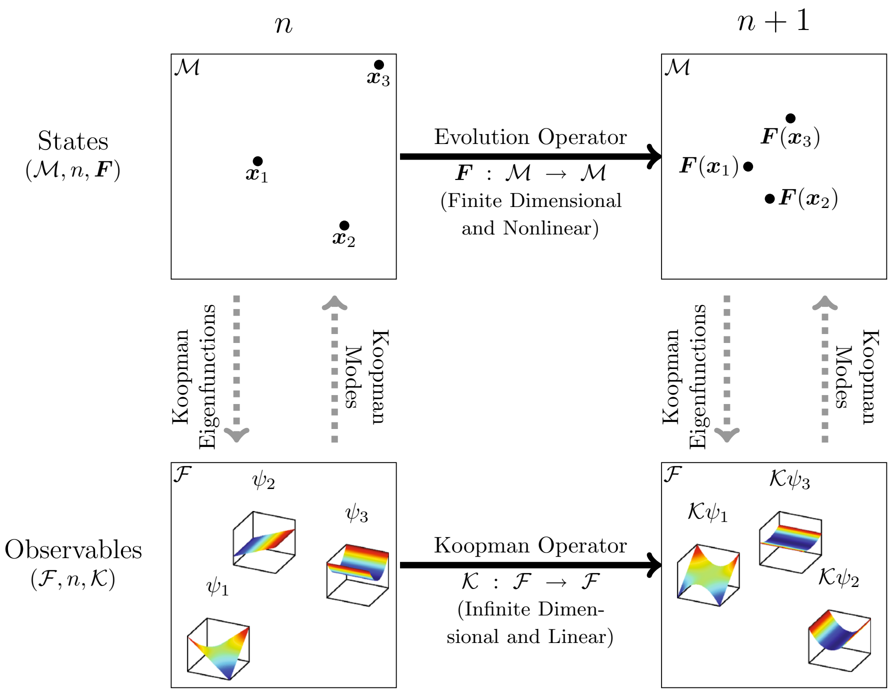

n
$n+1$
States
$(\mathcal{M}, n, \boldsymbol{F})$
$\mathcal{M}$
$\boldsymbol{x}_3$
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
Evolution Operator
$\boldsymbol{F} : \mathcal{M} \to \mathcal{M}$
(Finite Dimensional and Nonlinear)
$\mathcal{M}$
$\boldsymbol{F}(\boldsymbol{x}_3)$
$\boldsymbol{F}(\boldsymbol{x}_1)$
$\boldsymbol{F}(\boldsymbol{x}_2)$
Koopman Eigenfunctions
Koopman Modes
Koopman Eigenfunctions
Koopman Modes
Observables
$(\mathcal{F}, n, \mathcal{K})$
$\mathcal{F}$
ψ_2
ψ_3
ψ_1
Koopman Operator
$\mathcal{K} : \mathcal{F} \to \mathcal{F}$
(Infinite Dimensional and Linear)
$\mathcal{F}$
$\mathcal{K}\psi_3$
$\mathcal{K}\psi_1$
$\mathcal{K}\psi_2$

Fig. 1 A cartoon of the Koopman operator and how it relates to the underlying dynamical system. The *top path* updates the state, $\boldsymbol{x} \in \mathcal{M}$, using the evolution operator $\boldsymbol{F}$. The *bottom path* updates the observables, $\psi \in \mathcal{F}$, using the Koopman operator, $\mathcal{K}$. Here, both dynamical systems are autonomous, so the (discrete) time, $n \in \mathbb{Z}$, does not appear explicitly. The connection between the states and observables is through the *full-state observable* $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$. By writing $\boldsymbol{g}$ in terms of the Koopman eigenfunctions, we substitute the complex evolution of $\boldsymbol{x}$ with the straightforward, linear evolution of the φ_i. To reconstruct $\boldsymbol{x}$, we superimpose the Koopman eigenfunctions evaluated at a point, which satisfy $(\mathcal{K}\varphi_i)(x_i) = \mu_i \varphi_i(\boldsymbol{x}_i)$, using the Koopman modes as shown in (3). As a result, these two "paths" commute, and one can either solve a finite dimensional but nonlinear problem (the *top path*) or an infinite dimensional but linear problem (the *bottom path*) if one can compute the Koopman eigenvalues, eigenfunctions, and modes

$$\boldsymbol{g}(\boldsymbol{x}(t + \Delta t)) = (\mathcal{K}_{\Delta t}\boldsymbol{g})(\boldsymbol{x}(t)) = \sum_{k=1}^{N_K} e^{\lambda_k \Delta t} \boldsymbol{v}_k \varphi_k(\boldsymbol{x}(t)), \tag{6}$$

which is similar in form to (3), but now "fast" and "slow" are determined by the real part of λ_k rather than its absolute value.

In what follows, we will not have access to the "right-hand side" function, $\boldsymbol{f}$, and therefore cannot approximate $\hat{\mathcal{K}}$ directly. However, if the discrete-time dynamical system $\boldsymbol{F}$ is the flow map associated with $\boldsymbol{f}$ for a fixed time interval Δt (i.e., $\mathcal{K} = \mathcal{K}_{\Delta t}$), then (3) and (6) are equivalent with $\mu_k = e^{\lambda_k \Delta t}$. As a result, although we will be approximating the Koopman operator associated with discrete-time dynamical systems, we will often present our results in terms of λ_k rather than μ_k when the underlying system is a flow rather than map.

2.2 Extended Dynamic Mode Decomposition

In this subsection, we outline extended dynamic mode decomposition (EDMD), which is a method that approximates the Koopman operator and therefore the Koopman eigenvalue, eigenfunction, and mode tuples defined in Sect. 2.1. The EDMD procedure requires two prerequisites: (a) a data set of snapshot pairs, i.e., $\{(\boldsymbol{x}_m, \boldsymbol{y}_m)\}_{m=1}^{M}$ that we will organize as *a pair of data sets*,

$$\boldsymbol{X} = \left[\boldsymbol{x}_1 \; \boldsymbol{x}_2 \; , \ldots , \; \boldsymbol{x}_M\right], \qquad \boldsymbol{Y} = \left[\boldsymbol{y}_1 \; \boldsymbol{y}_2 \; , \ldots , \; \boldsymbol{y}_M\right], \tag{7}$$

where $\boldsymbol{x}_i \in \mathcal{M}$ and $\boldsymbol{y}_i \in \mathcal{M}$ are *snapshots of the system state* with $\boldsymbol{y}_i = \boldsymbol{F}(\boldsymbol{x}_i)$, and (b) a dictionary of observables, $\mathcal{D} = \{\psi_1, \psi_2, \ldots, \psi_{N_K}\}$ where $\psi_i \in \mathcal{F}$, whose span we denote as $\mathcal{F}_{\mathcal{D}} \subset \mathcal{F}$; for brevity, we also define the vector-valued function $\boldsymbol{\Psi} : \mathcal{M} \to \mathbb{C}^{1\times N_K}$ where

$$\boldsymbol{\Psi}(\boldsymbol{x}) = \left[\psi_1(\boldsymbol{x}) \; \psi_2(\boldsymbol{x}) \; , \ldots , \; \psi_{N_K}(\boldsymbol{x})\right]. \tag{8}$$

The data set needed is typically constructed from multiple short bursts of simulation or from experimental data. For example, if the data were given as a single time series, then for a given snapshot $\boldsymbol{x}_i$, $\boldsymbol{y}_i = \boldsymbol{F}(\boldsymbol{x}_i)$ is the next snapshot in the time series. In the original definition of DMD (Schmid 2010; Rowley et al. 2009), the data provided to the algorithm was assumed to come in the form of a single time series. This restriction was relaxed in Tu et al. (2014), where it was shown that a generalization of DMD that requires only a data set of snapshot pairs would produce equivalent results. Because EDMD is based on this definition of DMD, we will assume only that $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ are related by the dynamics even if the entire data set is in the form of a time series. However, if the data were the form of a time series, then EDMD could also be understood using the Krylov-based arguments used to justify “standard” DMD (Rowley et al. 2009). Furthermore, the optimal choice of dictionary elements remains an open question, but a short discussion including some pragmatic choices will be given in Sect. 3. For now, we assume that $\mathcal{D}$ is “rich enough” to accurately approximate a few of the leading Koopman eigenfunctions.

2.2.1 Approximating the Koopman Operator and its Eigenfunctions

Now we seek to generate $\boldsymbol{K} \in \mathbb{C}^{N_K\times N_K}$, a finite-dimensional approximation of $\mathcal{K}$. By definition, a function $\psi \in \mathcal{F}_{\mathcal{D}}$ can be written as

$$\psi(\boldsymbol{x}) = \sum_{k=1}^{N_K} a_k \psi_k(\boldsymbol{x}) = \boldsymbol{\Psi}(\boldsymbol{x})\boldsymbol{a}, \tag{9}$$

the linear superposition of the N_K elements in the dictionary with the weights $\boldsymbol{a}$. Note that unlike in (2), where N_k could be infinite, N_k is finite in both (9) and all that follows. Because $\mathcal{F}_{\mathcal{D}}$ is typically not an invariant subspace of $\mathcal{K}$,

$$(\mathcal{K}\psi)(\boldsymbol{x}) = (\boldsymbol{\Psi} \circ \boldsymbol{F})(\boldsymbol{x})\boldsymbol{a} = \boldsymbol{\Psi}(\boldsymbol{x})(\boldsymbol{K}\boldsymbol{a}) + r(\boldsymbol{x}) \tag{10}$$

which includes the residual term $r \in \mathcal{F}$. To determine $\boldsymbol{K}$, we will minimize

$$\begin{aligned} J &= \frac{1}{2}\sum_{m=1}^{M} |r(\boldsymbol{x}_m)|^2 \\ &= \frac{1}{2}\sum_{m=1}^{M} |((\boldsymbol{\Psi} \circ \boldsymbol{F})(\boldsymbol{x}_m) - \boldsymbol{\Psi}(\boldsymbol{x}_m)\boldsymbol{K})\boldsymbol{a}|^2 \\ &= \frac{1}{2}\sum_{m=1}^{M} \left|(\boldsymbol{\Psi}(\boldsymbol{y}_m) - \boldsymbol{\Psi}(\boldsymbol{x}_m)\boldsymbol{K})\boldsymbol{a}\right|^2 \end{aligned} \tag{11}$$

where $\boldsymbol{x}_m$ is the mth snapshot in $\boldsymbol{X}$, and $\boldsymbol{y}_m = \boldsymbol{F}(\boldsymbol{x}_m)$ is the mth snapshot in $\boldsymbol{Y}$. Equation 11 is a least-squares problem and therefore cannot have multiple isolated local minima; it must have either a unique global minimizer or a continuous family (or families) of minimizers. As a result, regularization (here via the truncated singular value decomposition) may be required to ensure the solution is unique, and the $\boldsymbol{K}$ that minimizes (11) is:

$$\boldsymbol{K} \triangleq \boldsymbol{G}^{+}\boldsymbol{A}, \tag{12}$$

where $+$ denotes the pseudoinverse and

$$\boldsymbol{G} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\Psi}(\boldsymbol{x}_m)^{*}\boldsymbol{\Psi}(\boldsymbol{x}_m), \tag{13a}$$

$$\boldsymbol{A} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\Psi}(\boldsymbol{x}_m)^{*}\boldsymbol{\Psi}(\boldsymbol{y}_m), \tag{13b}$$

with $\boldsymbol{K}, \boldsymbol{G}, \boldsymbol{A} \in \mathbb{C}^{N_K \times N_K}$. As a result, $\boldsymbol{K}$ is a finite-dimensional approximation of $\mathcal{K}$ that maps $\psi \in \mathcal{F}_{\mathcal{D}}$ to some other $\hat{\psi} \in \mathcal{F}_{\mathcal{D}}$ by minimizing the residuals at the data points. As a consequence, if $\boldsymbol{\xi}_j$ is the jth eigenvector of $\boldsymbol{K}$ with the eigenvalue μ_j, then the EDMD approximation of an eigenfunction of $\mathcal{K}$ is

$$\varphi_j(\boldsymbol{x}) = \boldsymbol{\Psi}(\boldsymbol{x})\boldsymbol{\xi}_j. \tag{14}$$

Finally, in many applications the discrete-time data in $\boldsymbol{X}$ and $\boldsymbol{Y}$ are generated by a continuous-time process with a sampling interval of Δt. If this is the case, we define $\lambda_j = \frac{\ln(\mu_j)}{\Delta t}$ to approximate the eigenvalues of the *continuous-time system*. In the remainder of the manuscript, we denote the eigenvalues of $\boldsymbol{K}$ with the μ_j and (when applicable) the approximation of the corresponding continuous-time eigenvalues as λ_j. Although both embody the same information, one choice is often more natural for a specific problem.

2.2.2 Computing the Koopman Modes

Next, we will compute approximations of the Koopman modes for the full-state observable using EDMD. Recall that the Koopman modes are the weights needed to express the full state in the *Koopman eigenfunction* basis. As such, we will proceed in two steps: First, we will express the full-state observable using the elements of $\mathcal{D}$; then, we will find a mapping from the elements of $\mathcal{D}$ to the numerically computed eigenfunctions. Applying these two steps in sequence will yield the observables expressed as a linear combination of Koopman eigenfunctions, which are, by definition, the Koopman modes for the full-state observable.

Recall that the full-state observable, $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$, is a vector-valued observable (e.g., $\boldsymbol{g} : \mathcal{M} \to \mathbb{R}^N$) that can be generated by "stacking" N scalar-valued observables, $g_i : \mathcal{M} \to \mathbb{R}$, as follows:

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} g_1(\boldsymbol{x}) \\ g_2(\boldsymbol{x}) \\ \vdots \\ g_N(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_1^* \boldsymbol{x} \\ \boldsymbol{e}_2^* \boldsymbol{x} \\ \vdots \\ \boldsymbol{e}_N^* \boldsymbol{x} \end{bmatrix}, \tag{15}$$

where $\boldsymbol{e}_i$ is the ith unit vector in $\mathbb{R}^N$. At this time, we conveniently assume that all $g_i(\boldsymbol{x}) \in \mathcal{F}_{\mathcal{D}}$ so that $g_i(\boldsymbol{x}) = \sum_{k=1}^{N_K} \psi_k(\boldsymbol{x}) b_{k,i} = \boldsymbol{\Psi}(\boldsymbol{x}) \boldsymbol{b}_i$, where $\boldsymbol{b}_i$ is some appropriate vector of weights. If this is not the case, *approximate* Koopman modes can be computed by projecting g_i onto $\mathcal{F}_{\mathcal{D}}$, though the accuracy and usefulness of this fit clearly depend on the choice of $\mathcal{D}$. To avoid this issue, we take $g_i \in \mathcal{F}_{\mathcal{D}}$ for $i = 1, \ldots, N$ in all examples that follow. In either case, the entire vector-valued observable can be expressed (or approximated) in this manner as

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{B}^T \boldsymbol{\Psi}(\boldsymbol{x})^T = (\boldsymbol{\Psi}(\boldsymbol{x}) \boldsymbol{B})^T, \qquad \boldsymbol{B} = \left[\boldsymbol{b}_1 \ \boldsymbol{b}_2 \ , \ldots, \ \boldsymbol{b}_N\right], \tag{16}$$

where $\boldsymbol{B} \in \mathbb{C}^{K \times N_K}$.

Next, we will express the ψ_i in terms of all the φ_i, which are our numerical approximations of the Koopman eigenfunctions. For notational convenience, we define the vector-valued function $\boldsymbol{\Phi} : \mathcal{M} \to \mathbb{C}^{1 \times N_K}$, where

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \left[\varphi_1(\boldsymbol{x}) \ \varphi_2(\boldsymbol{x}) \ , \ldots, \ \varphi_{N_K}(\boldsymbol{x})\right]. \tag{17}$$

Using (14) and (16), this function can also be written as

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{\Psi}(\boldsymbol{x}) \boldsymbol{\Xi}, \qquad \boldsymbol{\Xi} = \left[\boldsymbol{\xi}_1 \ \boldsymbol{\xi}_2 \ , \ldots, \ \boldsymbol{\xi}_{N_K}\right], \tag{18}$$

where $\boldsymbol{\xi}_i \in \mathbb{C}^{N_K}$ is the ith eigenvector of $\boldsymbol{K}$ associated with μ_i. Therefore, we can determine the ψ_i as a function of φ_i by inverting $\boldsymbol{\Xi}^T$. Because $\boldsymbol{\Xi}$ is a matrix of eigenvectors, its inverse is

$$\boldsymbol{\Xi}^{-1} = \boldsymbol{W}^* = \left[\boldsymbol{w}_1 \ \boldsymbol{w}_2 \ , \ldots, \ \boldsymbol{w}_{N_K}\right]^*, \tag{19}$$

where $\boldsymbol{w}_i$ is the ith left eigenvector of $\boldsymbol{K}$ also associated with μ_i (i.e., $\boldsymbol{w}_i^* \boldsymbol{K} = \boldsymbol{w}_i^* \mu_i$) appropriately scaled so $\boldsymbol{w}_i^* \boldsymbol{\xi}_i = 1$. We combine (16) and (19), and after some slight algebraic manipulation found that

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{V}\boldsymbol{\Phi}(\boldsymbol{x})^T = \sum_{k=1}^{N_K} \boldsymbol{v}_k \varphi_k, \qquad \boldsymbol{V} = \left[\boldsymbol{v}_1\ \boldsymbol{v}_2\ \dots\ \boldsymbol{v}_{N_K}\right] = \left(\boldsymbol{W}^* \boldsymbol{B}\right)^T, \tag{20}$$

where $\boldsymbol{v}_i = (\boldsymbol{w}_i^* \boldsymbol{B})^T$ is the ith Koopman mode. This is the formula for the Koopman modes that we desired.

2.2.3 Algorithm Summary

This subsection has been added to address the fourth referee's suggestion about an algorithm summary section. In summary, the EDMD procedure requires the user to supply the following:

(1) A data set of snapshot pairs $\{(\boldsymbol{x}_m, \boldsymbol{y}_m)\}_{m=1}^{M}$.
(2) A set of functions, $\mathcal{D} = \{\psi_1, \psi_2, \dots, \psi_{N_K}\}$, that will be used to approximate the Koopman eigenfunctions; for brevity, we define the vector-valued function, $\boldsymbol{\Psi}$, in (8) that contains these dictionary elements.
(3) A matrix $\boldsymbol{B}$, defined in (16), which contains the weights required to reconstruct the full-state observable using the elements of $\mathcal{D}$.

With these prerequisites, the procedure to obtain the Koopman eigenvalues, modes, and eigenfunctions is as follows:

(1) Loop through the data set of snapshot pairs to form the matrices $\boldsymbol{G}$ and $\boldsymbol{A}$ using (13); these computations can be performed in parallel and are compatible with the MapReduce framework (Dean and Ghemawat 2008).
(2) Form $\boldsymbol{K} \triangleq \boldsymbol{G}^+ \boldsymbol{A}$. Then, compute the set of eigenvalues, μ_i, eigenvectors, $\boldsymbol{\xi}_i$, and left eigenvectors, $\boldsymbol{w}_i$.
(3) Compute the set of Koopman modes by setting $\boldsymbol{v}_i \triangleq (\boldsymbol{w}_i^* \boldsymbol{B})^T$, where $\boldsymbol{v}_i$ is the ith Koopman mode.

As a result of this procedure, we have a set of μ_i, which is our approximation of the ith Koopman eigenvalue and a set of $\boldsymbol{v}_i$, which is our approximation of the ith Koopman mode. We also have the vector $\boldsymbol{\xi}_i$, which allows the ith Koopman eigenfunction to be approximated using (14).

Ultimately, the EDMD procedure is a regression procedure and can be applied to any set of snapshot pairs without modification. However, the contribution of this manuscript is to show how the regression problem above relates to the Koopman operator, and making this connection requires some knowledge about the process that created the data. If the underlying system is (effectively) a discrete-time, autonomous dynamical system (or, equivalently, an autonomous flow sampled at a fixed interval of Δt), then the results of this section hold. If the underlying system is stochastic, then a connection with the Koopman operator still exists, but will be discussed later in Sect. 5. Furthermore, even some non-autonomous systems could, in principle, be analyzed using EDMD by augmenting the state vector to include time; this, however, will be left for future work.

2.3 Convergence of the EDMD Algorithm to a Galerkin Method

In this subsection, we relate EDMD to the Galerkin methods one would use to approximate the Koopman operator with complete information about the underlying dynamical system. In this context, a Galerkin method is a weighted-residual method where the residual, as defined in (10), is orthogonal to the span of $\mathcal{D}$. In particular, we show that the EDMD approximation of the Koopman operator converges to the approximation that would be obtained from a Galerkin method in the large-data limit. In this manuscript, the large-data limit is when $M \to \infty$ and the elements of $\boldsymbol{X}$ are drawn independently from a distribution on $\mathcal{M}$ with the probability density ρ. Note that one method of generating a more complex ρ (e.g., not a Gaussian or uniform distribution) is to randomly sample points from a single, infinitely long trajectory. In this case, ρ is one of the natural measures associated with the underlying dynamical system. We also assume that $\mathcal{F} = L^2(\mathcal{M}, \rho)$. The first assumption defines a process for adding new data points to our set and could be replaced with other sampling schemes. The second assumption is required so that the inner products in the Galerkin method converge, which is relevant for problems where $\mathcal{M} = \mathbb{R}^N$.

If EDMD were a Galerkin method, then the entries of $\boldsymbol{G}$ and $\boldsymbol{A}$ in (12) would be defined as

$$\begin{aligned} \hat{\boldsymbol{G}}_{ij} &= \int_{\mathcal{M}} \psi_i^*(\boldsymbol{x})\psi_j(\boldsymbol{x})\rho(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x} = \langle \psi_i, \psi_j \rangle_\rho , \\ \hat{\boldsymbol{A}}_{ij} &= \int_{\mathcal{M}} \psi_i^*(\boldsymbol{x})\psi_j(\boldsymbol{F}(\boldsymbol{x}))\rho(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x} = \langle \psi_i, \mathcal{K}\psi_j \rangle_\rho , \end{aligned} \tag{21}$$

where $\langle p, q \rangle_\rho = \int_{\mathcal{M}} p^*(\boldsymbol{x})q(\boldsymbol{x})\rho(\boldsymbol{x})\ \ \mathrm{d}\boldsymbol{x}$ is the inner product and the finite-dimensional Galerkin approximation of the Koopman operator would be $\hat{\boldsymbol{K}} = \hat{\boldsymbol{G}}^{-1}\hat{\boldsymbol{A}}$. The performance of this method certainly depends upon the choice of ψ_j and ρ, but it is nevertheless a Galerkin method as the residual would be orthogonal to $\mathcal{F}_{\mathcal{D}}$ (Boyd 2013; Trefethen 2000). There are non-trivial questions about what sets of ψ and what measures, ρ, are required if the Galerkin method is to generate a *useful* approximation of the Koopman operator (e.g., when can we "trust" our eigenfunctions if ρ is compactly supported but $\mathcal{M} = \mathbb{R}^N$?), but they are beyond the scope of this manuscript and will be the focus of future work.

For a finite M, the ijth element of $\boldsymbol{G}$ is

$$\boldsymbol{G}_{ij} \triangleq \frac{1}{M}\sum_{m=1}^{M} \psi_i^*(\boldsymbol{x}_m)\psi_j(\boldsymbol{x}_m), \tag{22a}$$

So the ijth element of $\boldsymbol{G}$ contains the sample mean of $\psi_i^*(\boldsymbol{x})\psi_j(\boldsymbol{x})$. Similarly,

$$\boldsymbol{A}_{ij} \triangleq \frac{1}{M}\sum_{m=1}^{M} \psi_i^*(\boldsymbol{x}_m)\psi_j(\boldsymbol{y}_m). \tag{22b}$$

When M is finite, (21) is approximated by (22). However, by the law of large numbers, the sample means almost surely converge to the expected values when the number of samples, M, becomes sufficiently large. For this system, the expectations can be written as

$$\begin{aligned} \lim_{M\to\infty} \boldsymbol{G}_{ij} &= \int_{\mathcal{M}} \psi_i^*(\boldsymbol{x})\psi_j(\boldsymbol{x})\rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \left\langle \psi_i, \psi_j \right\rangle_\rho = \hat{\boldsymbol{G}}_{ij}, \\ \lim_{M\to\infty} \boldsymbol{A}_{ij} &= \int_{\mathcal{M}} \psi_i^*(\boldsymbol{x})\psi_j(\boldsymbol{F}(\boldsymbol{x}))\rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \left\langle \psi_i, \mathcal{K}\psi_j \right\rangle_\rho = \hat{\boldsymbol{A}}_{ij}, \end{aligned} \tag{23}$$

which reintroduces the integrals in (21). As a result, the entries of $\boldsymbol{A}$ and $\boldsymbol{G}$ converge to their analytically determined values, and therefore, the output of the EDMD procedure will converge to the output of a Galerkin method. With randomly distributed initial data, the needed integrals are computed using Monte Carlo integration, and the rate of convergence will be $\mathcal{O}(M^{-1/2})$. Other sampling choices, such as placing points on a uniform grid, effectively use different quadrature rules and could therefore obtain a better rate of convergence.

Implicit in this argument is the fact that $\boldsymbol{x}_m$ and $\boldsymbol{y}_m$ are snapshots of the system state, which implies that a snapshot, say $\boldsymbol{x}_m$, cannot map to "two different places." However, there are many cases where $\boldsymbol{x}_m$ and $\boldsymbol{y}_m$ are only partial measurements of the system state, in which case this could happen. For example, $\boldsymbol{x}_m$ and $\boldsymbol{y}_m$ could consist of N elements of a vector in $\mathbb{R}^{N_0}$ where $N_0 > N$, and so some state measurements have simply been neglected. This is quite common and occurs when the data have been compressed using POD or other related techniques. Another common example is where $\boldsymbol{x}_m$ and $\boldsymbol{y}_m$ are snapshots of the full state, but the underlying system is periodically forced; in this case, the missing information is the "phase" of the forcing.

Because the Koopman operator acts on scalar observables, one could still consider the EDMD procedure as an approximation of a Koopman operator using a set of functions, ψ_k, that are constant in the "missing" components. As a result, if the true eigenfunctions are nearly constant in these directions [an assumption that is justifiable in some cases, such as fast-slow systems (Froyland et al. 2014)], then the missing information may have only a small impact on the resulting eigenfunctions. However, this assumption clearly does not hold in general, so care must be taken in order to have as "complete" a set of measurements as possible.

2.4 Relationship with DMD

When M is not large, EDMD will not be an accurate Galerkin method because the quadrature errors generated by the Monte Carlo integrator will be significant, and so the residual will probably not be orthogonal to $\mathcal{F}_{\mathcal{D}}$. However, it is still formally an extension of DMD, which has empirically been shown to yield meaningful results even without exhaustive data sets. In this section, we show that EDMD is equivalent to DMD *for a very specific—and restrictive—choice of $\mathcal{D}$* because EDMD and DMD will produce the same set of eigenvalues and modes for any set of snapshot pairs.

Because there are many conceptually equivalent but mathematically different definitions of DMD, the one we adopt here is taken from Tu et al. (2014), which defines the *DMD modes* as the eigenvectors of the matrix

$$\boldsymbol{K}_{\text{DMD}} \triangleq \boldsymbol{Y}\boldsymbol{X}^{+}, \tag{24}$$

where the jth mode is associated with the jth eigenvalue of $\boldsymbol{K}_{\text{DMD}}$, μ_j. $\boldsymbol{K}_{\text{DMD}}$ is constructed using the data matrices in (7), where $+$ again denotes the pseudoinverse. This definition is a generalization of preexisting DMD algorithms (Schmid 2010; Schmid et al. 2011) and does not require the data to be in the form of a single time series. All that is required are states and their images (for a map) or their updated positions after a fixed interval in time of Δt (for a flow). However, in the event that they *are* in the form of a single time series, then as discussed in Tu et al. (2014), this generalization is related to the Krylov method presented in Rowley et al. (2009).

Now, we will prove that the Koopman modes computed using EDMD are equivalent to DMD modes, if the dictionary used for EDMD consists of scalar observables of the form $\psi_i(\boldsymbol{x}) = \boldsymbol{e}_i^* \boldsymbol{x}$ for $i = 1, \ldots, N$. *This is the special (if relatively restrictive) choice of dictionary alluded to earlier*. In particular, we show that the ith Koopman mode, $\boldsymbol{v}_i$, is also an eigenvector of $\boldsymbol{K}_{\text{DMD}}$, hence a DMD mode. Because the elements of the full-state observable are the dictionary elements, $\boldsymbol{B} = \boldsymbol{I}$ in (16). Then, the Koopman modes are the complex conjugates of the left eigenvectors of $\boldsymbol{K}$, so $\boldsymbol{v}_i^T = \boldsymbol{w}_i^*$. Furthermore, $\boldsymbol{G}^T = \frac{1}{M}\boldsymbol{X}\boldsymbol{X}^*$ and $\boldsymbol{A}^T = \frac{1}{M}\boldsymbol{Y}\boldsymbol{X}^*$. Then,

$$\boldsymbol{K}^T = \boldsymbol{A}^T \boldsymbol{G}^{T+} = \boldsymbol{Y}\boldsymbol{X}^* \left(\boldsymbol{X}\boldsymbol{X}^*\right)^{+} = \boldsymbol{Y}\boldsymbol{X}^{+} = \boldsymbol{K}_{\text{DMD}}. \tag{25}$$

Therefore, $\boldsymbol{K}_{\text{DMD}}\boldsymbol{v}_i = (\boldsymbol{v}_i^T \boldsymbol{K}_{\text{DMD}}^T)^T = (\boldsymbol{w}_i^* \boldsymbol{K})^T = (\mu_i \boldsymbol{w}_i^*)^T = \mu_i \boldsymbol{v}_i$, and all the Koopman modes computed by EDMD are eigenvectors of $\boldsymbol{K}_{\text{DMD}}$ and are thus the DMD modes. Once again, the choice of dictionary is critical; *EDMD and DMD are equivalent only for this very specific $\mathcal{D}$*, and other choices of $\mathcal{D}$ will enable EDMD to generate different (and potentially more useful) results.

Conceptually, DMD can be thought of as *producing an approximation of the Koopman eigenfunctions* using the set of linear monomials as basis functions for $\mathcal{F}_{\mathcal{D}}$, which is analogous to a one-term Taylor expansion. For problems where the eigenfunctions can be approximated accurately using linear monomials (e.g., in some small neighborhood of a stable fixed point), then DMD will produce an *accurate* local approximation of the Koopman eigenfunctions. However, this is certainly not the case for all systems (particularly beyond the region of validity for local linearization). EDMD can be thought of as an extension of DMD that retains additional terms in the expansion, where these additional terms are determined by the elements of $\mathcal{D}$. The quality of the resulting approximation is governed by $\mathcal{F}_{\mathcal{D}}$ and, therefore, depends upon the choice of $\mathcal{D}$. In all the examples that follow, the dictionaries we use will be strict supersets of the dictionary chosen implicitly by DMD. Assuming this "richer" dictionary and using the argument presented in Tu et al. (2014), it is easy to show that if DMD is

Table 1 Some commonly used dictionaries of functions and the application where they are, in our experience, best suited

Name	Suggested context
Hermite polynomials	Problems defined on $\mathbb{R}^N$ with normally distributed data
Radial basis functions	Problems defined on irregular domains
Discontinuous spectral elements	Large problems where a block-diagonal $\boldsymbol{G}$ is beneficial/computationally important

able to exactly recover a Koopman mode and eigenvalue from a given set of data, then EDMD will also exactly recover the same Koopman modes and eigenvalues. However, in many applications the tuples of interest cannot be exactly recovered. In these cases, the idea is that the more extensive dictionary used by EDMD will produce better approximations of the leading Koopman tuples because $\mathcal{F}_\mathcal{D}$ is larger and therefore better able to represent the eigenfunctions of interest.

3 The Choice of the Dictionary

As with all spectral methods, the accuracy and rate of convergence of EDMD depend on $\mathcal{D}$, whose elements span the subspace of observables, $\mathcal{F}_\mathcal{D} \subset \mathcal{F}$. Possible choices for the elements of $\mathcal{D}$ include: polynomials (Boyd 2013), Fourier modes (Trefethen 2000), radial basis functions (Wendland 1999), and spectral elements (Karniadakis and Sherwin 2013), but the optimal choice of basis functions likely depends on both the underlying dynamical system and the sampling strategy used to obtain the data. Any of these sets are, in principle, a useful choice for $\mathcal{D}$, though some care must be taken on infinite domains to ensure that any needed inner products will converge.

Choosing $\mathcal{D}$ for EDMD is, in some cases, more difficult than selecting a set of basis functions for use in a standard spectral method because the domain on which the underlying dynamical system is defined, $\mathcal{M}$, and is not necessarily known. Typically, we can define $\Omega \supset \mathcal{M}$ so that it contains all the data in $\boldsymbol{X}$ and $\boldsymbol{Y}$; e.g., pick Ω to be a "box" in $\mathbb{R}^N$ that contains every snapshot in $\boldsymbol{X}$ and $\boldsymbol{Y}$. Next, we choose the elements of $\mathcal{D}$ to be a basis for $\tilde{\mathcal{F}}_\mathcal{D} \subset \tilde{\mathcal{F}}$, where $\tilde{\mathcal{F}}$ is the space of functions that map $\Omega \to \mathbb{C}$. Because $\mathcal{F} \subset \tilde{\mathcal{F}}$, this choice of $\mathcal{D}$ can be used in the EDMD procedure, but there is no guarantee that the elements of $\mathcal{D}$ form a basis for $\mathcal{F}_\mathcal{D}$ as there may be redundancies. The potential for these redundancies and the numerical issues they generate is why regularization, and hence, the pseudoinverse (Hansen 1990) is required in (12). An example of these redundancies and their effects is given in Appendix.

Although the optimal choice of $\mathcal{D}$ is unknown, there are three choices that are broadly applicable in our experience. They are: Hermite polynomials, radial basis functions (RBFs), and discontinuous spectral elements. The *Hermite polynomials* are the simplest of the three sets and are best suited to problems defined on $\mathbb{R}^N$ if the data in $\boldsymbol{X}$ are normally distributed. The observables that comprise $\mathcal{D}$ are products of the Hermite polynomials in a single dimension (e.g., $H_1(x)H_2(y)H_0(z)$, where H_i is the ith Hermite polynomial and $\boldsymbol{x} = (x, y, z)$). This set of basis functions is simple to implement and conceptually related to approximating the Koopman eigenfunctions with a Taylor expansion. Furthermore, because they are orthogonal with respect to Gaussian weights, $\boldsymbol{G}$ will be diagonal if the $\boldsymbol{x}_m$ are drawn from a normal distribution, which can be beneficial numerically.

An alternative to the Hermite polynomials is *discontinuous spectral elements*. To use this set, we define a set of B_N boxes, $\{\mathcal{B}_i\}_{i=1}^{B_N}$, such that $\cup_{i=1}^{B_N}\mathcal{B}_i \supset \mathcal{M}$. Then, on each of the $\mathcal{B}_i$, we define K_i (suitably transformed) Legendre polynomials. For example, in one dimension, each basis function is of the form

$$\psi_{ij}(x) = \begin{cases} L_j(\xi) & x \in \mathcal{B}_i, \\ 0 & \text{otherwise}, \end{cases} \tag{26}$$

where L_j is the jth Legendre polynomial and ξ is x transformed such that the "edges" of the box are at $\xi = \pm 1$. The advantage of this basis is that $\boldsymbol{G}$ will be block-diagonal and therefore easy to invert even if a very large number of basis functions are employed.

With a fixed amount of data, an equally difficult task is choosing the $\mathcal{B}_i$; the number and arrangement of the $\mathcal{B}_i$ are a balance between span of the basis functions (i.e., h-type convergence), which increases as the number of boxes is increased, and the accuracy of the quadrature rule, which decreases because smaller boxes contain fewer data points. To generate a covering of $\mathcal{M}$, we use a method similar to the one used by GAIO (Dellnitz et al. 2001). Initially, all the data (i.e., $\boldsymbol{X}$ and $\boldsymbol{Y}$) are contained within a single user-selected box, $\mathcal{B}_1^{(0)}$. If this box contains more than a pre-specified number of data points, it is subdivided into 2^N domains of equal Lebesgue measure (e.g., in one dimension, $\mathcal{B}_1^{(0)} = \mathcal{B}_1^{(1)} \cup \mathcal{B}_2^{(1)}$). We then proceed recursively: if any of $\mathcal{B}_i^{(1)}$ contain more than a pre-specified number of points, then they too are subdivided; this proceeds until no box has an "overly large" number of data points. Any $\mathcal{B}_i^{(j)}$ that do not contain any data points are pruned, which after j iterates leaves the set of subdomains, $\{\mathcal{B}_i^{(j)}\}$, on which we define the Legendre polynomials. The resulting set of functions are compactly supported and can be evaluated efficiently using 2^N trees, where N is the dimension of a snapshot. Finally, the higher-order polynomials used here allow for more rapid p-type convergence if the eigenfunctions happen to be smooth.

The final choice is a set of *radial basis functions* (RBFs), which appeal to previous work on "mesh-free" methods (Liu 2010). Because these methods do not require a computational grid or mesh, they are particularly effective for problems where $\mathcal{M}$ has what might be called a complex geometry. Many different RBFs could be effective, but one particularly useful set of RBFs are the thin plate splines (Wendland 1999; Belytschko et al. 1996) because they do not require the scaling parameter that other RBFs (e.g., Gaussians) do. However, we still must choose the "centers" about which the RBFs are defined, which we do with k-means clustering (Bishop 2006) with a pre-specified value of k on the combined data set. Although we make no claims of optimality, in our examples, the density of the RBF centers appears to be directly related to the density of data points, which is, intuitively, a reasonable method for distributing the RBF centers as regions with more samples will also have more spatial resolution.

There are, of course, other dictionaries that may prove more effective in other circumstances. For example, basis functions defined in polar coordinates are useful when limit cycles or other periodic orbits are present as they mimic the form of the Koopman eigenfunctions for simple limit cycles (Bagheri 2013). How to choose the best set of functions is an important, yet open, question; fortunately, the EDMD method often produces useful results even with the relatively naïve choices presented in this section and summarized in Table 1.

4 Deterministic Data and the Koopman Eigenfunctions

Most applications of DMD assume that the data sets were generated by a deterministic dynamical system. In Sect. 2, we showed that, as long as the dictionary can accurately approximate the leading Koopman eigenfunctions, EDMD produces an approximation of the Koopman eigenfunctions, eigenvalues, and modes with large amounts of data (the regime in which EDMD is equivalent to a Galerkin method). In this section, we demonstrate that EDMD can produce accurate approximations of the Koopman eigenfunctions, eigenvalues, and modes with relatively limited amounts of data by applying the method to two illustrative examples. The first is a discrete-time linear system, for which the eigenfunctions, eigenvalues, and modes are known analytically, and serves as a test case for the method. The second is the unforced Duffing equation. Our goal there is to demonstrate that the approximate Koopman eigenfunctions obtained via EDMD have the potential to serve as a data-driven parameterization of a system with multiple basins of attraction.

4.1 A Linear Example

4.1.1 The Governing Equation, Data, and Analytically Obtained Eigenfunctions

One system where the Koopman eigenfunctions are known analytically is a simple LTI system of the form

$$\boldsymbol{x}(n+1) = \boldsymbol{J}\boldsymbol{x}(n), \tag{27}$$

with $\boldsymbol{x}(n) \in \mathbb{R}^N$ and $\boldsymbol{J} \in \mathbb{R}^{N\times N}$. It is clear that an eigendecomposition yields complete information about the underlying system provided $\boldsymbol{J}$ has a complete set of eigenvectors. Because the underlying dynamics are linear, it should not be surprising that the Koopman approach contains *the eigendecomposition* of $\boldsymbol{J}$.

To show this, note that the action of the Koopman operator for this problem is

$$\mathcal{K}\psi(\boldsymbol{x}) = \psi(\boldsymbol{J}\boldsymbol{x}), \tag{28}$$

where $\psi \in \mathcal{F}$. Assuming $\boldsymbol{J}$ has a complete set of eigenvectors, it will have N left eigenvectors, $\boldsymbol{w}_i$, that satisfy $\boldsymbol{w}_i^*\boldsymbol{J} = \mu_j \boldsymbol{w}_i^*$, and where the ith eigenvector is associated with the eigenvalue μ_i. Then, the function

$$\varphi_{n_1,n_2,\ldots,n_N}(\boldsymbol{x}) = \prod_{i=1}^{N} (\boldsymbol{w}_i^*\boldsymbol{x})^{n_i} \tag{29}$$

is an eigenfunction of the Koopman operator with the eigenvalue $\prod_{i=1}^{N} \mu_i^{n_i}$ for $n_i \in \mathbb{N}$. This is a well-known result (see, e.g., Rowley et al. 2009), so we show the proof of this only for the sake of completeness. We proceed directly:

$$\begin{aligned}\mathcal{K}\varphi_{n_1,n_2,\ldots,n_N}(\boldsymbol{x}) &= \prod_{i=1}^{N} (\boldsymbol{w}_i^* \boldsymbol{J}\boldsymbol{x})^{n_i} = \prod_{i=1}^{N} \mu_i^{n_i} (\boldsymbol{w}_i^* \boldsymbol{x})^{n_i} \\ &= \left(\prod_{i=1}^{N} \mu_i^{n_i} \right) \varphi_{n_1,n_2,\ldots,n_N}(\boldsymbol{x}),\end{aligned}$$

by making use of (28) and the definition of $\boldsymbol{w}_i$ as a left eigenvector. Then, the representation of the full-state observable in terms of the Koopman modes and eigenfunctions (i.e., (2)) is

$$\boldsymbol{x} = \sum_{i=1}^{N} \boldsymbol{v}_i \left(\boldsymbol{w}_i^* \boldsymbol{x} \right), \tag{30}$$

where the ith Koopman mode, $\boldsymbol{v}_i$, is the ith eigenvector of $\boldsymbol{J}$ suitably scaled so that $\boldsymbol{w}_i^* \boldsymbol{v}_i = 1$. This is identical to writing $\boldsymbol{x}$ in terms of the eigenvectors of $\boldsymbol{J}$; inner products with the left eigenvectors determine the component in each direction, and the (right) eigenvectors allow the full state to be reconstructed. As a concrete example, consider

$$\boldsymbol{x}(n+1) = \begin{bmatrix} 0.9 & -0.1 \\ 0.0 & 0.8 \end{bmatrix} \boldsymbol{x}(n) = \boldsymbol{J}\boldsymbol{x}(n), \tag{31}$$

where $\boldsymbol{x}_n = [x_n, y_n]$. From (29), the Koopman eigenfunctions and eigenvalues are

$$\varphi_{ij}(x, y) = \left(\frac{x - y}{\sqrt{2}} \right)^i y^j, \qquad \lambda_{ij} = (0.9)^i (0.8)^j, \tag{32}$$

for $i, j \in \mathbb{Z}$, where the factor of $1/\sqrt{2}$ was added to normalize the left eigenvectors of $\boldsymbol{J}$. Figure 2 shows the first 8 nontrivial eigenfunctions sorted by their associated eigenvalue. The zeroth eigenfunction, $\varphi_{00}(\boldsymbol{x}) = 1$ with $\mu_{00} = 1$, was omitted because it is always an eigenfunction and will be recovered by EDMD if $\psi = 1$ is included as a dictionary element.

To apply the EDMD procedure, one needs both data and a dictionary of observables. The data in $\boldsymbol{X}$ consist of 100 normally distributed initial conditions, $\boldsymbol{x}_i$, and their images, $\boldsymbol{y}_i = \boldsymbol{J}\boldsymbol{x}_i$, which we aggregate in the matrix $\boldsymbol{Y}$, i.e., $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{2\times 100}$. The dictionary, $\mathcal{D}$, is chosen to contain the direct product of Hermite polynomials in x and in y that include up to the fourth-order terms in x and y, i.e.,

$$\begin{aligned}\mathcal{D} &= \{\psi_0, \psi_1, \psi_2, \ldots\} \\ &= \begin{matrix} \{H_0(x)H_0(y), & H_1(x)H_0(y), & H_2(x)H_0(y), & H_3(x)H_0(y), & H_4(x)H_0(y), \\ H_0(x)H_1(y), & H_1(x)H_1(y), & H_2(x)H_1(y), & H_3(x)H_1(y), & H_4(x)H_1(y), \\ H_0(x)H_2(y), & H_1(x)H_2(y), & H_2(x)H_2(y), & H_3(x)H_2(y), & H_4(x)H_2(y), \\ H_0(x)H_3(y), & H_1(x)H_3(y), & H_2(x)H_3(y), & H_3(x)H_3(y), & H_4(x)H_3(y), \\ H_0(x)H_4(y), & H_1(x)H_4(y), & H_2(x)H_4(y), & H_3(x)H_4(y), & H_4(x)H_4(y)\}, \end{matrix}\end{aligned} \tag{33}$$

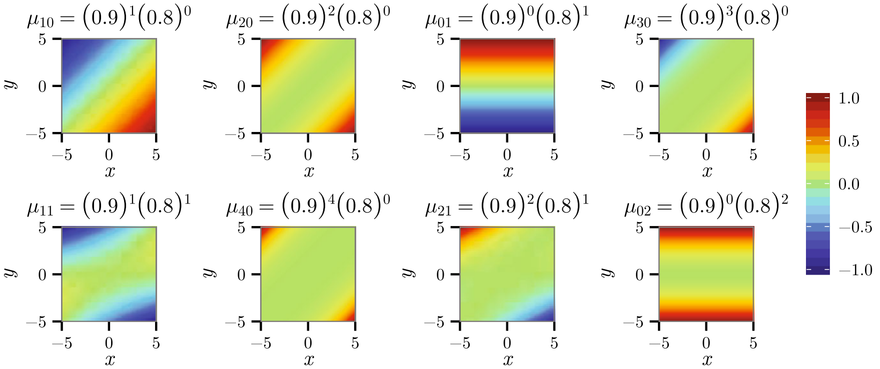


Fig. 2 Pseudocolor plots of the first eight Koopman eigenfunctions, as ordered by their corresponding eigenvalue, plotted using their analytical expressions, (32), along with the associated eigenvalue. Note that the eigenfunctions were scaled so that $\|\varphi_i\|_\infty = 1$ on the domain shown

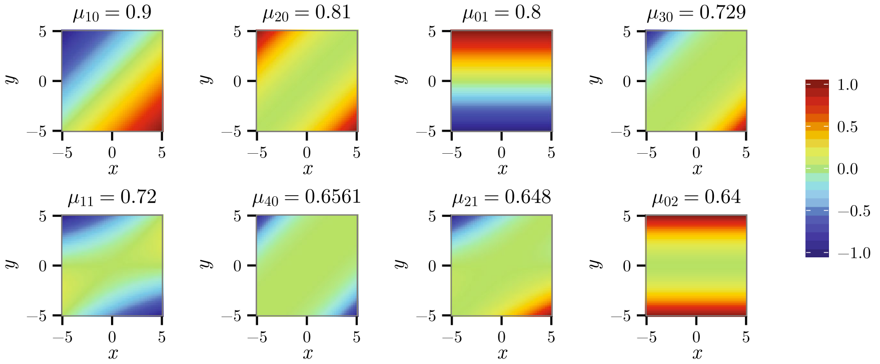


Fig. 3 Pseudocolor plots of the first eight non-trivial eigenfunctions of the Koopman operator, ordered by their corresponding eigenvalue, computed using the EDMD procedure. The eigenfunctions obtained using EDMD were scaled such that $\|\varphi_i\|_\infty = 1$ on the domain shown. With this scaling, there is excellent agreement between these results and those presented in Fig. 2

so the function ψ_i is the product of a Hermite polynomial in x of degree $(i \mod 5)$ and a Hermite polynomial in y of degree $\left\lfloor \frac{i}{5} \right\rfloor$. The Hermite polynomials were chosen because they are orthogonal with respect to the weight function $\rho(\boldsymbol{x}) = e^{-\|\boldsymbol{x}\|^2}$ that is implicit in the normally distributed sampling strategy used here and can also represent the leading Koopman eigenfunctions in this problem.

4.1.2 Results

Figure 3 shows the same eight eigenfunctions computed using the EDMD method. Overall, there is (as expected) excellent quantitative agreement, both in the eigenvalues and in the eigenfunctions, with the analytical results presented in Fig. 2. On the domain shown, the eigenvalues are accurate to ten digits, and the maximum pointwise difference between the true and computed eigenfunction is 10^{-6}. In this problem,

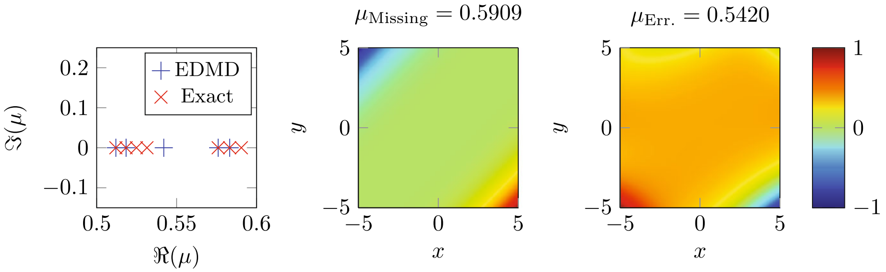


Fig. 4 A subset of the spectrum of the Koopman operator and the EDMD computed approximation. As shown here, there are clearly errors in the spectrum further away from the unit circle (which is not contained in the plotting region). The *center plot* shows an example of a “missing” eigenfunction that is not captured by EDMD; this eigenfunction is proportional to $(x - y)^5 \notin \mathcal{F}_{\mathcal{D}}$ and cannot be represented with our basis functions. The *right plot* shows an example of an “erroneous” eigenfunction that appears because $\mathcal{F}_{\mathcal{D}}$ is not an invariant subspace of the Koopman operator

standard DMD also generates highly accurate approximations of φ_{10} and φ_{01} along with their associated eigenvalues, but *will not produce any of the other eigenfunctions*; the standard choice of the dictionary contains only linear terms and, therefore, cannot reproduce eigenfunctions with constant terms or any nonlinear terms. As a result, expanding the basis allows EDMD to capture more of the Koopman eigenfunctions than standard DMD could. These additional eigenfunctions are not necessary to reconstruct the full-state observable of an LTI system, but are in principle needed in nonlinear settings.

This level of accuracy is in large part because the first nine eigenfunctions are in $\mathcal{F}_{\mathcal{D}}$, the subspace of observables spanned by $\mathcal{D}$. When this is not the case, the result is either a *missing or erroneous* eigenfunction like the examples shown in Fig. 4. The eigenfunction $\varphi_{50} = \left(\frac{x-y}{\sqrt{2}}\right)^5$ with $\mu = 0.9^5 = 0.59049$ is not captured by EDMD *with the dictionary chosen here* because it lacks the needed fifth-order monomials in x and y, which is similar to how DMD skips the second Koopman eigenfunction due to a lack of quadratic terms.

The erroneous eigenfunction appears because $\mathcal{F}_{\mathcal{D}}$ is not invariant with respect to the action of the Koopman operator. In particular, φ_{50} contains the term yx^4 whose image $\mathcal{K}(yx^4) \notin \mathcal{F}_{\mathcal{D}}$ because $x^5, y^5 \notin \mathcal{F}_{\mathcal{D}}$. In most applications, there are small components of the eigenfunction that cannot be represented in the dictionary chosen, which results in errors in the eigenfunction such as the one seen here. Even in the limit of infinite data, we would compute the eigenfunctions of $\mathcal{P}_{\mathcal{F}_{\mathcal{D}}}\mathcal{K}$, where $\mathcal{P}_{\mathcal{F}_{\mathcal{D}}}$ is the projection onto $\mathcal{F}_{\mathcal{D}}$, rather than the eigenfunction of $\mathcal{K}$. To see that this not a legitimate eigenfunction, we added $H_5(x)$ and $H_5(y)$ to $\mathcal{D}$, which removes this erroneous eigenfunction.

In our experience, erroneous eigenfunctions tend to appear in one of the two places. Sometimes they will be associated with one of the slowly decaying eigenvalues and in that case are often associated with lower mode energies (i.e., the value of the corresponding Koopman eigenfunction is small when the corresponding Koopman mode is normalized). Otherwise, they also typically form a “cloud” of rapidly decaying eigenvalues, which are ignored as we are primarily only interested in the leading

Koopman eigenvalues. One pragmatic method for determining whether or not an eigenvalue is spurious or not is to apply the EDMD procedure to subsets of the data and compare the resulting eigenvalues; while all of the eigenvalues will be perturbed, the erroneous ones tend to have larger fluctuations. Unfortunately, missing eigenvalues are more difficult to detect without prior knowledge, but tend to live in the same part of the complex plane that the erroneous eigenvalues do as both are caused by a lack of dictionary elements. In practice, we take the cloud of erroneous eigenvalues as the cutoff point below which the EDMD method cannot be trusted to produce accurate results.

Finally, we compute the Koopman modes for the full-state observable. Using the ordering of the dictionary elements given in (33), the weight matrix, $\boldsymbol{B}$ in (16), needed to compute the Koopman modes for the full-state observable, $\boldsymbol{g}(\boldsymbol{x})^T = [x, y]$ is

$$\boldsymbol{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ \vdots & \vdots \end{bmatrix}. \tag{34}$$

This $\boldsymbol{B}$ in conjunction with (20) yields a numerical approximation of a Koopman mode for each of the Koopman eigenvalues computed previously. The Koopman modes associated with $\mu_{10} = 0.9$ is $\boldsymbol{v}_{10} = [0, -\sqrt{2}]^T$, while the Koopman mode associated with $\mu_{01} = 0.8$ is $\boldsymbol{v}_{01} = [-1, -1]^T$; again, these are the eigenvectors of $\boldsymbol{J}$. The contribution of the other *numerically computed* eigenfunctions in reconstructing the full-state observable is negligible (i.e., $\|\boldsymbol{v}_k\| \approx 10^{-11}$ for $k \neq 1, 3$), so the Koopman/EDMD analysis is an eigenvalue/eigenvector decomposition once numerical errors are taken into consideration.

Although EDMD reveals a richer set of Koopman eigenfunctions that are analytically known to exist, their associated Koopman modes are zero, and hence, they can be neglected. Our goal in presenting this example is not to demonstrate any new phenomenon, but rather to demonstrate that there is good quantitative agreement between the analytically obtained Koopman modes, eigenvalues, and eigenfunctions and the approximations produced by EDMD. Furthermore, it allowed us to highlight the types of errors that appear when $\mathcal{F}_{\mathcal{D}}$ is not an invariant subspace of $\mathcal{K}$, which results in erroneous eigenfunctions, or when the dictionary is missing elements, which results in missing eigenfunctions.

4.2 The Duffing Equation

In this section, we will compute the Koopman eigenfunctions for the unforced Duffing equation, which for the parameter regime of interest here has two stable spirals and a saddle point whose stable manifold defines the boundary between the basins of attraction. Following Mauroy et al. (2013) and the references contained therein, the eigenvalues of the linearizations about the fixed points in the system are known to be a subset of the Koopman eigenvalues, and for each stable spiral, the magnitude and phase of the associated Koopman eigenfunction parameterizes the relevant basin of attraction. Additionally, because basins of attraction are forward invariant sets, there will be two eigenfunctions with $\mu = 0$, each of which is supported on one of the two basins of attraction in this system (or, equivalently, there will be a trivial eigenfunction and another eigenfunction with $\mu = 0$ whose level sets denote the basins of attraction). Ultimately, we are not interested in recovering highly accurate eigenfunctions in this example. Instead, we will demonstrate that the eigenfunctions computed by EDMD are *accurate enough* that they can be used to identify and parameterize the basins of attraction that are present in this problem for the region of interest.

The governing equations for the unforced Duffing equation are

$$\ddot{x} = -\delta\dot{x} - x(\beta + \alpha x^2), \tag{35}$$

which we will study using the parameters $\delta = 0.5$, $\beta = -1$, and $\alpha = 1$. In this regime, there are two stable spirals at $x = \pm 1$ with $\dot{x} = 0$, and a saddle at $x, \dot{x} = 0$, so almost every initial condition, except for those on the stable manifold of the saddle, is drawn to one of the spirals. Despite the existence of an unstable equilibrium, the continuous-time Koopman eigenvalues for this system that are required to reconstruct the full-state observable are all non-positive, which can intuitively be understood by considering the value of $\lim_{t\to\infty} \boldsymbol{e}_i^*(\boldsymbol{x}(t))$ for the ith unit vector $\boldsymbol{e}_i$. For any initial condition $\boldsymbol{x}(0)$, the values of these observables will relax to either 0 or ± 1 depending on i and the initial condition. This relaxation implies that the eigenvalues used in the expansion in (6) should not have positive real parts, otherwise exponential growth would be observed.

This is shown in Gaspard and Tasaki (2001) and Gaspard et al. (1995) for the pitchfork and Hopf bifurcations, where the eigenfunctions and *eigendistributions* can be obtained analytically. Unlike those examples, we do not have analytical expressions for the eigenfunctions (or eigendistributions) here, but we will appeal to the same intuition. In particular, we expect EDMD to approximate the Koopman eigenfunctions associated with the attractors at $(\pm 1, 0)$ because those quantities, at least away from the saddle, can lie in or near the subspace spanned by our dictionary. There will also be eigenvalues associated with the unstable equilibrium at the origin, but they are paired with eigendistributions [e.g., Dirac delta distributions and their derivatives (Gaspard and Tasaki 2001; Gaspard et al. 1995)], which are not in the span of any dictionary we will choose and therefore could not be recovered even if we had a large quantity of data near that equilibrium.

Therefore, we use a set of data that consists of 10^3 trajectories with 11 samples each with a sampling interval of $\Delta t = 0.25$ (i.e., $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{2\times 10^4}$), and initial conditions uniformly distributed over $x, \dot{x} \in [-2, 2]$. With this sampling rate and initialization scheme, many trajectories will *approach* the stable spirals, but few will have (to numerical precision) reached the fixed points. As a result, the basins of attraction cannot be determined by observing the last snapshot in a given trajectory. Instead, EDMD will

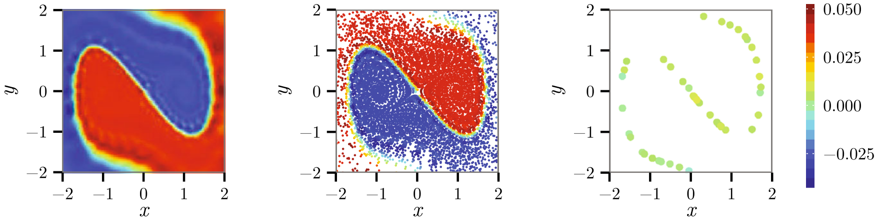


Fig. 5 (*left*) A plot of the first non-trivial eigenfunction generated with an eigenvalue of $\lambda = -10^{-3}$ obtained from 10^3 randomly initialized trajectories each consisting of ten steps taken with $\Delta t = 0.25$ This eigenfunction should be constant in each basin of attraction and have $\lambda = 0$, so EDMD is generating an *approximation of the eigenfunction* rather than a fully converged solution. (*center*) The numerically computed eigenfunction on the left evaluated at the points where we had data. (*right*) A plot of the misclassified points (less than 0.5 % of the data); each of these points lies (to the eye) on the boundary between the invariant sets

be used to “stitch” together this ensemble of trajectories to form a single coherent picture.

However, because there are multiple basins of attraction, the leading eigenfunctions appear to be discontinuous (Mauroy et al. 2013) and supported only on the appropriate basin of attraction. In principle, our computation could be done “all at once” using a single $\mathcal{D}$ and applying EDMD to the complete data set. To enforce the compactly supported nature of the eigenfunctions regardless of which dictionary we use, we will proceed in a two-tiered fashion. First, the basins of attraction will be identified using all of the data and a dictionary with support everywhere we have data. Once we have identified these basins, both state space and the data will be partitioned into subdomains based on the numerically identified basins. The EDMD procedure will then be run on each subdomain and the corresponding partitioned data set individually.

4.2.1 Locating Basins of Attraction

Figure 5 highlights the first step: partitioning of state space into basins of attraction. We used a dictionary consisting of the constant function and 1,000 radial basis functions (RBFs) (the thin plate splines described in Sect. 3), where k-means clustering (Bishop 2006) on the full data set was used to choose the RBF centers. RBFs were chosen here because of the geometry of the computational domain; indeed, RBFs are often a fundamental component of “mesh-free” methods that avoid the non-trivial task of generating a computational mesh (Liu 2010).

The leading (continuous-time) eigenvalue is $\lambda_0 = -10^{-14}$ which corresponds to the constant function. The second eigenfunction, shown in the leftmost image of Fig. 5, has $\lambda_1 = -10^{-3}$, which should be considered an approximation of zero. The discrepancy between the numerically computed eigenfunction and the theoretical one is due to the choice of the dictionary. The analytical eigenfunction possesses a discontinuity on the edge of the basin of attraction (i.e., the stable manifold of the saddle point at the origin), but discontinuous functions are not in the space spanned by RBFs. Therefore, the numerically computed approximation “blurs” this edge as shown in Fig. 5.

The scatter plot in the center of Fig. 5 shows the data points colored by the first nontrivial eigenfunction. There is good qualitative agreement between the numerically computed basin of attraction and the actual basin. By computing the mean value of φ_1 on the data and using that value as the threshold that determines which basin of attraction a point belongs to, the EDMD approach misclassifies only 46 of the 10^4 data points, resulting in an error of only 0.5 % as shown by the rightmost plot. As a result, the leading eigenfunctions computed by EDMD are sufficiently accurate to produce a meaningful partition of the data.

4.2.2 Parameterizing a Basin of Attraction

Now that the basins of attraction have been identified, the next task is to develop a coordinate system or parameterization of the individual basins. To do so, we will use the eigenfunctions associated with the eigenvalues of the system linearization about the corresponding fixed point. Because these fixed points are spirals, this parameterization can be realized using the amplitude and phase of one member of the complex-conjugate pair of eigenfunctions. To approximate these eigenfunctions, we first partition our data into two sets as mentioned above using the leading Koopman eigenfunctions (including the misclassified data points). On each subset of the data, the k-means procedure was run again to select a new set of 1000 RBF centers, and this "adjusted" basis along with the constant function comprised the $\mathcal{D}$ used by EDMD. Figure 6 shows the amplitude and phase of the eigenfunction with eigenvalue closest to $-0.25 + 1.387\imath$ computed using the data in each basin of attraction. The computed eigenvalues agree favorably with the analytically obtained eigenvalue; the basin of the spiral at $(1, 0)$ has the eigenvalue $-0.237 + 1.387\imath$, and the basin of the spiral at $(-1, 0)$ has the eigenvalue $-0.24 + 1.35\imath$.

Figure 6 demonstrates that the amplitude and phase of a Koopman eigenfunction forms something analogous to an "action–angle" parameterization of the basin of attraction. Due to the nonlinearity in the Duffing oscillator, this parameterization is more complicated than an appropriately shifted polar coordinate system and is, therefore, not the parameterization that would be generated by linearization about either $(\pm 1, 0)$. The level sets of the amplitude of this eigenfunction are the so-called isostables (Mauroy et al. 2013). One feature predicted in that manuscript is that the 0-level set of the isostable is the fixed point in the basin of attraction; this feature is reflected in Fig. 6 by the blue region, which corresponds to small values of the eigenfunction that are near the fixed points at $(\pm 1, 0)$. Additionally, a singularity in the phase can be observed there. The EDMD approach produces noticeable numerical errors near the edges of the basin. These errors can be due to a lack of data or due to the singularities in the eigenfunctions that can occur at unstable fixed points (Mauroy and Mezic 2013).

In this section, we applied the EDMD procedure to deterministic systems and showed that it produces an approximation of the Koopman operator. With a sensible choice of data and $\mathcal{D}$, we showed that EDMD generates a quantitatively accurate approximation of the Koopman eigenvalues, eigenfunctions, and modes for the linear example. In the second example, we used the Koopman eigenfunctions to identify and parameterize the basins of attraction of the Duffing equation. Although the EDMD approximation of the eigenfunctions could be made more accurate with more data,

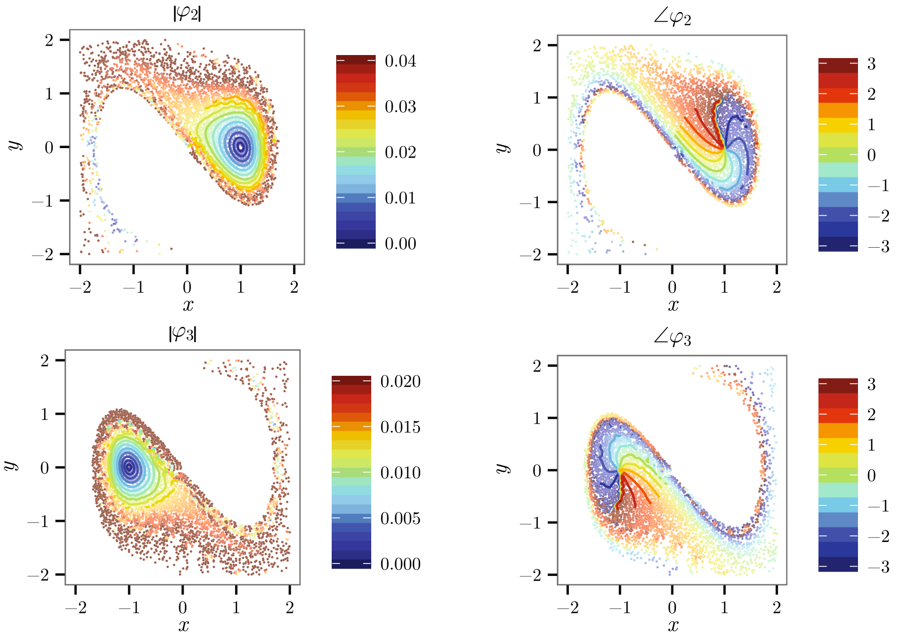


Fig. 6 (*top*) Amplitude and phase of the Koopman eigenfunction with $\lambda = -0.237 + 1.387i$ (analytically, $-0.250 + 1.392i$) for the stable spiral at $(1, 0)$. (*bottom*) The same pair of plots for the spiral at $(-1, 0)$. In all four images, the data points are colored by value of the eigenfunction. Near the equilibria, the level sets of the eigenfunctions are included as a guide for the eye. Although there are errors near the "edge" of the basin of attraction, the amplitude and phase of this eigenfunction can serve as a polar coordinate system for the corresponding basin of attraction

it is still accurate enough to serve as an effective parameterization. As a result, the EDMD method can be useful with limited quantities of data and should be considered an enabling technology for data-driven approximations of the Koopman eigenvalues, eigenfunction, and modes.

5 Stochastic Data and the Kolmogorov Backward Equation

The EDMD approach is entirely data driven and will produce an output regardless of the nature of the data given to it. However, if the results of EDMD are to be meaningful, then certain assumptions must be made about the dynamical system that produced the data used. In the previous section, it was assumed that the data were generated by a deterministic dynamical system; as a result, EDMD produced approximations of the tuples of Koopman eigenfunctions, eigenvalues, and modes.

Another interesting case to consider is when the underlying dynamical system is a Markov process, such as a stochastic differential equation (SDE). For such systems, the evolution of an observable is governed by the Kolmogorov backward (KB) equation [33], whose "right-hand side" has been called the "stochastic Koopman operator" (SKO) (Mezić 2005). In this section, we will show that EDMD produces approximations of the eigenfunctions, eigenvalues, and modes of the SKO if the underlying dynamical system happens to be a Markov process.

To accomplish this, we will prove that the EDMD method converges to a Galerkin method in the large-data limit. After that, we will demonstrate its accuracy with finite amounts of data by applying it to the model problem of a one-dimensional SDE with a double-well potential, where the SKO eigenfunctions can be computed using standard numerical methods.

Another proposed application of the Koopman operator is for the purposes of *model reduction*, which was first explored in Mezić (2005) and later work such as Froyland et al. (2014). Model reduction based on the Koopman eigenfunctions is equally applicable in both deterministic and stochastic settings, but we choose to present it for stochastic systems to highlight the similarities between EDMD and *manifold learning techniques* such as diffusion maps (Nadler et al. 2005; Coifman and Lafon 2006). In particular, we apply EDMD to an SDE defined on a "Swiss roll," which is a nonlinear manifold often used to test manifold learning methods (Lee and Verleysen 2007). The purpose of this example is twofold: First, we show that a data-driven parameterization of the Swiss roll can be obtained using EDMD, and second, we show that this parameterization will preferentially capture "slow" dynamics on that manifold before the "fast" dynamics when the noise is made anisotropic.

5.1 EDMD with Stochastic Data

For a *discrete-time Markov process*,

$$\boldsymbol{x} \mapsto \boldsymbol{F}(\boldsymbol{x}; \boldsymbol{\omega}),$$

the SKO (Mezić 2005) is defined as

$$(\mathcal{K}\psi)(\boldsymbol{x}) = \mathbb{E}[\psi(\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{\omega}))], \tag{36}$$

where $\boldsymbol{\omega} \in \Omega_s$ is an element in the probability space associated with the stochastic dynamics (Ω_s) and probability measure P, $\mathbb{E}$ denotes the expected value over that space, and $\psi \in \mathcal{F}$ is a scalar observable. The SKO (Mezić 2005) takes an observable of the full system state and returns the conditional expectation of the observable "one timestep in the future." Note that this definition is compatible with the deterministic Koopman operator because $\mathbb{E}[\psi(\boldsymbol{F}(\boldsymbol{x}))] = \psi(\boldsymbol{F}(\boldsymbol{x}))$ if $\boldsymbol{F}$ is deterministic.

As with the deterministic case, we assume the snapshots in $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ were generated by randomly placing initial conditions on $\mathcal{M}$ with the density of $\rho(\boldsymbol{x})$ and that M is sufficiently large. Once again, ρ does not need to be an invariant measure of the underlying dynamical system; it is simply the sampling density of the data. Due to the stochastic nature of the system, there are two probability spaces involved: one related to the samples in $\boldsymbol{X}$ and another for the stochastic dynamics. Because our system has "process" rather than "measurement" noise, the $\boldsymbol{x}_i$ are known exactly, and the interpretation of the Gram matrix, $\boldsymbol{G}$, remains unchanged. Therefore,

$$\lim_{M\to\infty} \boldsymbol{G}_{i,j} = \int_{\mathcal{M}} \psi_i^*(\boldsymbol{x})\psi_j(\boldsymbol{x})\rho(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \left\langle \psi_i, \psi_j \right\rangle_\rho ,$$

by the law of large numbers when M is large enough. This is identical to the deterministic case. However, the definition of $\boldsymbol{A}$ will change. Assuming that the choice of $\boldsymbol{\omega}$ and $\boldsymbol{x}$ are independent,

$$\begin{aligned}\lim_{M\to\infty} \boldsymbol{A}_{i,j} &= \mathbb{E}[\psi_i^*(\mathcal{K}\psi_j)] = \int_{\mathcal{M}\times\Omega_s} \psi_i^*(\boldsymbol{x})\psi_j(\boldsymbol{F}(\boldsymbol{x},\boldsymbol{\omega}))\rho(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}\mathrm{d}P(\boldsymbol{\omega}) \\ &= \int_{\mathcal{M}} \psi_i^*(\boldsymbol{x})\mathbb{E}[\psi_j(\boldsymbol{F}(\boldsymbol{x},\boldsymbol{\omega}))]\rho(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \left\langle \psi_i, \mathcal{K}\psi_j \right\rangle_\rho .\end{aligned} .$$

The elements of $\boldsymbol{A}$ now contain a *second* integral over the probability space that pertains to the stochastic dynamics, which produces the expectation of the observable in the expression above.

The accuracy of the resulting method will depend on the dictionary, $\mathcal{D}$, the manifold on which the dynamical system is defined, the data, and the dynamics used to generate it. One interesting special case is if the basis functions are indicator functions supported on "boxes." When this is the case, EDMD is equivalent to the widely used Ulam Galerkin method (Bollt and Santitissadeekorn 2013; Dellnitz et al. 2001). This equivalence is lost for other choices of $\mathcal{D}$ and ρ, but as we will demonstrate in the subsequent sections, EDMD can produce accurate approximations of the eigenfunctions for many other choices of these quantities.

The "stochastic Koopman modes" can then be computed using (20), but they too must be reinterpreted as the weights needed to reconstruct the *expected value* of the full-state observable using the eigenfunctions of the SKO. Due to the stochastic nature of the dynamics, the Koopman modes can no longer exactly specify the state of the system. However, they can be used as approximations of the Koopman modes that would be obtained in the "noise-free" limit when some appropriate restrictions are placed on the nature of the noise and the underlying dynamical system. Indeed, these are the modes we are truly computing when we apply DMD or EDMD to experimental data, which by its very nature contains some noise.

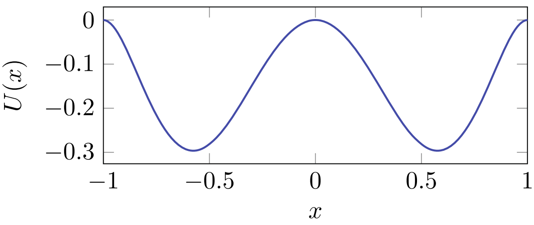
5.2 A Stochastic Differential Equation with a Double-Well Potential

In this section, we will show that the EDMD procedure is capable of accurately approximating the eigenfunctions of the stochastic Koopman operator by applying it to an SDE with a double-well potential. Although we do not have analytical solutions for the eigenfunctions, the problem is simple enough that we can accurately compute them using standard numerical methods.

5.2.1 The Double-Well Problem and Data

First, consider an SDE with a double-well potential. Let the governing equations for this system be

Fig. 7 Double-well potential $U(x) = -2(x^2 - 1)^2 x^2$



$$\mathrm{d}x = -\nabla U(x)dt + \sigma \mathrm{d}W_t, \tag{37}$$

where x is the state, $-\nabla U(x)$ the drift, σ is the (constant) the diffusion coefficient, and W_t is a Wiener process. Furthermore, no-flux boundary conditions are imposed at $x = \pm 1$. For this problem, we let $U(x) = -2(x^2 - 1)^2 x^2$ as shown in Fig. 7.

The Fokker–Planck equation associated with this SDE is

$$\frac{\partial \rho(x,t)}{\partial t} = -\frac{\partial}{\partial x}\left(-\frac{\partial U}{\partial x}\rho(x,t)\right) + \frac{\sigma^2}{2}\frac{\partial^2 \rho(x,t)}{\partial x^2} = \mathcal{P}\rho, \tag{38}$$

where ρ is a probability density with $\partial_x \rho(x,t)\big|_{x=\pm 1} = 0$ due to the no-flux boundary conditions we impose, and $\hat{\mathcal{P}}$ is the infinitesimal generator of the semigroup of Perron–Frobenius operators. The adjoint of the Perron–Frobenius operator determines the Kolmogorov backward equation and thus defines the generator of the semigroup of stochastic Koopman operators, $\hat{\mathcal{K}} = \hat{\mathcal{P}}^\dagger$, which we refer to as the continuous-time stochastic Koopman operator. In the remainder of this section, we focus on the continuous-time SKO rather than on its discrete-time analog like we did in Sect. 2. For this example,

$$\hat{\mathcal{K}}\psi = -\frac{\partial U}{\partial x}\frac{\partial \psi}{\partial x} + \frac{\sigma^2}{2}\frac{\partial^2 \psi}{\partial x^2} \tag{39}$$

with Neumann boundary conditions, $\partial_x \psi\big|_{x=\pm 1} = 0$. To directly approximate the Koopman eigenfunctions, (39) is discretized in space using a second-order finite-difference scheme with 1024 interior points. The eigenvalues and eigenvectors of the resulting finite-dimensional approximation of the Koopman operator will be used to validate the EDMD computations.

The data are 10^6 initial points on $x \in [-1, 1]$ drawn from a uniform distribution, which constitute $\boldsymbol{X}$, and their positions after $\Delta t = 0.1$, which constitute $\boldsymbol{Y}$. The evolution of each initial condition was accomplished through 10^2 steps of the Euler–Maruyama method (Higham 2001; Kloeden and Platen 1992) with a timestep of 10^{-3} using the double-well potential in Fig. 7. Note that only the initial and final points in this trajectory were retained, so we have access to 10^6 and not 10^8 snapshot pairs. The dictionary chosen is a discontinuous spectral element basis that splits $x \in [-1, 1]$ into four equally sized subdomains with up to tenth-order Legendre polynomials on each subdomain (see Sect. 3) for a total of forty degrees of freedom.

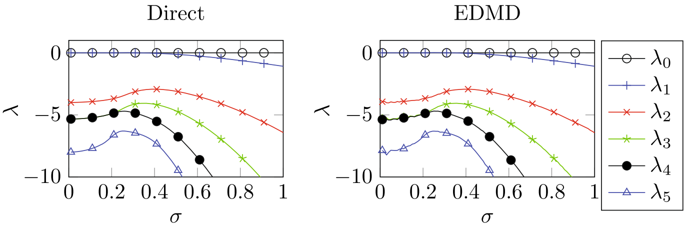


Fig. 8 (*left*) First six eigenvalues of the stochastic Koopman operator obtained with a finite-difference discretization of $\mathcal{K}$. (*right*) The first six eigenvalues obtained with the EDMD approach. In both plots, a marker is placed every tenth data point. While there is good quantitative agreement between the true eigenvalues and those obtained with EDMD, some small "noise" due to quadrature errors does appear in the right plot as $\sigma \to 0$

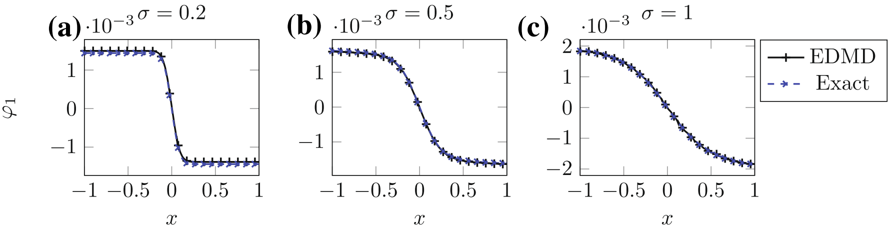


Fig. 9 A comparison of the leading non-trivial eigenfunction computed with the finite-difference method and EDMD for: **a** $\sigma = 0.2$, **b** $\sigma = 0.5$, and **c** $\sigma = 1.0$. As with the eigenvalues, there is excellent agreement between the "true" and data-driven approximations of this eigenfunction, though there are small quantitative differences in the approximation

5.2.2 Recovering the Koopman Eigenfunctions and Eigenvalues

Because the Koopman operator is infinite dimensional, we will clearly be unable to approximate all of the tuples. Instead, we focus on the leading (i.e., most slowly decaying) tuples, which govern the long-term dynamics of the underlying system. In this example, we seek to demonstrate that our approximation is: (a) quantitatively accurate and (b) valid over a range of coefficients, σ, and not solely in the small (or large) noise limits.

Figure 8 shows the first six eigenvalues obtained using a finite-difference discretization of the Koopman operator and the EDMD approximation as a function of σ. In this problem, the constant function $\varphi_0 = 1$ is always an eigenfunction of the Koopman operator with $\lambda_0 = 0$ for all values of σ. Because $\mathcal{D}$ contains the constant function, it should be no surprise that the EDMD method is able to identify it as an eigenfunction. Although trivial, the existence of this eigenfunction is a "sanity check" for the method.

More interesting is the first *non-trivial* eigenfunction which has the eigenvalue λ_1. The change in φ_1 as a function of σ is shown in Fig. 9. As with the eigenvalue, there is good agreement between EDMD and the directly computed eigenfunctions at different values of σ. For all values of σ, φ_1 is an odd function; what changes is how rapidly φ_1 transitions from its maximum to its minimum. When σ is small, this transition is rapid, and φ_1 will approach a step function as $\sigma \to 0$. When σ grows, this eigenfunction is "smoothed out" and the transition becomes slower. In the limit as $\sigma \to \infty$, the dynamics of the problem are dominated by the diffusion term, and φ_1 will be proportional to $\cos(\pi x/L)$ as is implied by the rightmost plot in the figure.

In many system-identification algorithms [e.g., Juang (1994)], one often constructs deterministic governing equations from inherently stochastic data (either due to measurement or due to process noise). Similarly, methods like DMD have been applied to noisy sets of data to produce an approximation of the Koopman modes and eigenvalues with the assumption that the underlying system is deterministic. In this example, this is equivalent to using the output of EDMD with data taken with $0 < \sigma \ll 1$ as an approximation of the Koopman tuples that would be obtained with $\sigma = 0$.

For certain tuples, this is a reasonable approach. Taking $\sigma \to 0$, λ_3 and λ_4 and φ_3 and φ_4 are good approximations of their deterministic counterparts. In particular, φ_3 and φ_4 are one-to-one with their associated basin of attraction and appear to possess a zero at the stable fixed point. However, these approximate eigenfunctions lack some important features such as a singularity at $x = 0$ that occurs due to the unstable fixed point there. Therefore, both eigenfunctions are good approximations of their $\sigma = 0$ counterparts, but cannot be "trusted" in the vicinity of an unstable fixed point.

For other tuples, even a small amounts of noise can be important. Consider the "slowest" nonzero eigenvalue, λ_2, which appears to approach -4 as $\sigma \to 0$, but is *not* obtained by the EDMD method when $\sigma = 0$. Formally, the existence of an eigenvalue of -4 is not surprising. The fixed point at $x = 0$ is unstable with $\lambda = 4$, and in continuous time, if (λ_n, φ_n) is an eigenvalue/eigenfunction pair then $(k\lambda_n, \varphi_n^k)$ is, at least formally, an eigenvalue/eigenfunction pair for any scalar k. Using an argument similar to Matkowsky and Schuss (1981), it can be shown that $\varphi_2(x) = C_0 \exp(-4x^2/\sigma^2) + \mathcal{O}(\sigma^2)$ as $\sigma \to 0$ where C_0 is chosen to normalize φ_2. However, this approaches a delta function as $\sigma \to 0$ and therefore leaves the subspace of observables spanned by our dictionary. When this occurs, this tuple appears to "vanish," which is why it does not appear in the $\sigma = 0$ limit. As a result, when applying methods like EDMD or DMD to noisy data, the spectrum of the finite-dimensional approximation is not necessarily a good approximation of the spectrum that would be obtained with noise-free data. Some of the tuples, such as those containing φ_1, φ_3, and φ_4, have eigenvalues that closely approximate the ones found in the deterministic problem. However, others such as the tuple containing φ_2 do not. Furthermore, the only method to determine that λ_2 can be neglected is by directly examining the eigenfunction. As a result, when we apply methods like DMD/EDMD to noisy data with the purpose of using the spectrum to determine the timescales and behaviors of the underlying system, we must keep in mind that not all of the eigenvalues obtained with noisy data will be present if "clean" data are used instead.

5.2.3 Rate of Convergence

Among other things, the performance of the EDMD method is dependent upon the number of snapshots provided to it, the distribution of the data, the underlying dynamical system, and the dictionary. In this section, we examine the convergence of EDMD to a Galerkin method as the number of snapshots increases in order to provide some intuition about the "usefulness" of the eigenfunctions obtained without an exhaustive amount of data. To do so, we generated a larger set of data consisting of 10^7 ini-

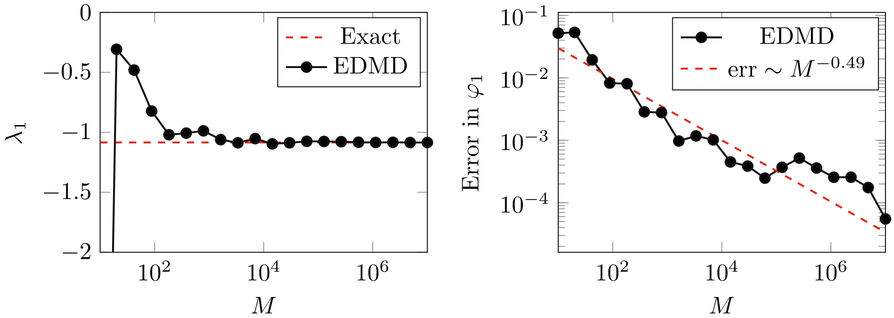


Fig. 10 (*left*) Plot of the first non-trivial eigenvalue as a function of the number of data points with $\sigma = 1$. The *red dashed line* denotes the "exact" value computed using direct numerical methods. Although the EDMD approximation is poor with $M < 100$, it quickly becomes more accurate as M is increased. (*right*) Plot of the error, i.e., $\|\varphi_{1,\text{EDMD}} - \varphi_{1,\text{True}}\|$, as a function of M. Because the scalar products in (12) are evaluated using Monte Carlo integration, the method converges like $\mathcal{O}(M^{-1/2})$ as shown by the fit on the *right* (Color figure online)

tial conditions chosen from a spatially uniform distribution for the case with $\sigma = 1$. Each initial condition was propagated using the Euler–Maruyama method described in the previous section, but only the initial and terminal states were retained. Then, we applied EDMD using the same dictionary to subsets of the data, computed the leading nontrivial eigenvalue and eigenfunction, and compared the results to the "true" leading eigenfunction and eigenvalue computed using a finite-difference approximation of the stochastic Koopman operator.

Figure 10 shows the convergence of the leading non-trivial eigenvalue and eigenfunction as a function of the number of snapshots, M. In the rightmost plot, we define the error as $\|\varphi_{1,\text{EDMD}} - \varphi_{1,\text{True}}\|$ after both eigenfunctions have been normalized so that $\|\varphi_{1,\text{EDMD}}\|_2 = \|\varphi_{1,\text{True}}\|_2$. As expected, EDMD is inaccurate when M is small (here, $M < 100$); there is not enough data to accurately approximate the scalar products. For $M > 10^3$, the eigenfunction produced by EDMD have the right shape, and the eigenvalue is approaching its true value. For $M > 10^4$, there is no "visible" difference in the leading eigenvalue, and the error in the leading eigenfunction is less than 10^{-3}.

To quantify the rate of convergence, we fit a line to the plot of error versus M in the right panel of Fig. 10. As expected, EDMD converges like $M^{-0.49}$, which is very close to the predicted value of $\mathcal{O}(M^{-0.5})$ associated with Monte Carlo integration. Because this problem is stochastic, we cannot increase the rate of convergence by uniform sampling (the integral over the probability space associated with the stochastic dynamics will still converge like $\mathcal{O}(M^{-1/2})$), even though that is a simple method for enhancing the rate of convergence for deterministic problems.

To provide some intuition about what the eigenfunctions with $\sigma = 1$ look like, Fig. 11 plots the leading nontrivial eigenfunction for $M = 1623$, 14384, and 1128837. Ultimately, the purpose of EDMD is to produce effective approximations of the leading Koopman eigenfunctions. As shown in this subsection, EDMD is qualitatively accurate even at the smallest values of M, but there are clearly some numerical issues at the edges of the domain and near $x = 0$ where the discontinuities in the numerically computed eigenfunctions can occur with our choice of dictionary. To obtain

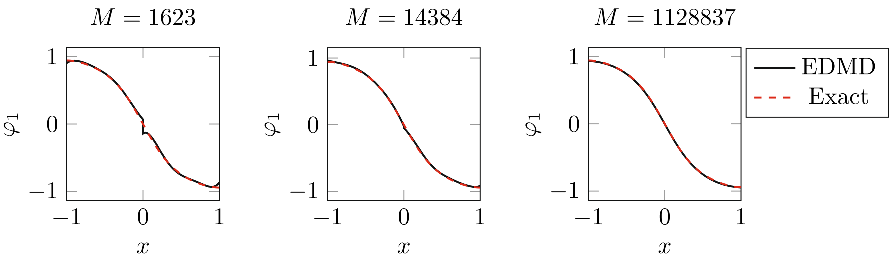


Fig. 11 Comparison of the EDMD approximation of the leading non-trivial Koopman eigenfunction with $\sigma = 1$ for $M = 1623$, 14384, and 1128837 with the "exact" solution obtained using direct numerical methods. The EDMD solution is similar to the exact solution even when $M \sim 10^3$, but we would not characterize it as quantitatively accurate until $M > 10^4$. Beyond $M > 10^5$, EDMD and the exact solution are visually identical

a more quantitatively accurate solution, additional data points are required. When $M = 14384$, the numerical issues at the boundaries and the discontinuity at $x = 0$ have diminished. As shown in the plot with $M = 1128837$, this process continues until the EDMD eigenfunction is visually identical to the true eigenfunction. In principle, highly accurate approximations of the leading Koopman eigenfunctions using EDMD are possible, but because EDMD for stochastic systems is a Monte Carlo method, the relatively slow rate of convergence may make the amount of data required to obtain this level of accuracy infeasible in practice.

5.3 Parameterizing Nonlinear Manifolds and Reducing Stochastic Dynamics

In this section, we will briefly demonstrate how the EDMD method can be used to parameterize nonlinear manifolds and reduce stochastic differential equations defined on those manifolds. Everything done here could also be done for a deterministic system; we chose to use an SDE rather than an ODE only to highlight the similarities between EDMD and nonlinear manifold learning techniques such as diffusion maps, and not because of any restriction on the Koopman approach. We proceed in two steps: First, we will show that data from an SDE defined on the Swiss roll, which is a nonlinear manifold often used as a test of nonlinear manifold learning techniques (Nadler et al. 2005, 2006; Coifman and Lafon 2006; Lee and Verleysen 2007); in conjunction with the EDMD procedure can generate a data-driven parameterization of that manifold. For this first example, isotropic diffusion is used, so there is no "fast" or "slow" solution component that can be meaningfully neglected. Instead, we will show that the leading eigenfunctions are one-to-one with the "length" and "width" of the Swiss roll. Then, we alter the SDE and introduce a "fast" component by making the diffusion term anisotropic. In this case, EDMD will "pick" the slower components before the faster ones. We should stress that in this application, EDMD could be used as a replacement for (rather than in conjunction with) other methods such as diffusion maps (DMAPs) and its variants (Nadler et al. 2005, 2006; Coifman and Lafon 2006; Dsilva et al. 2013). Although there are advantages to *combining diffusion maps and Koopman* (Budisic and Mezic 2012), exploration of those approaches is beyond the scope of this manuscript.

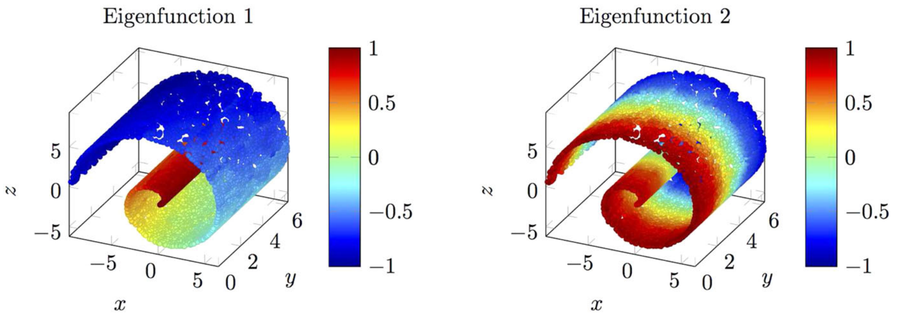


Fig. 12 First two non-trivial Koopman eigenfunctions for the diffusion process on the "Swiss roll" using 3×10^4 data points. The first eigenfunction is one-to-one with s_1 (the "length" of the Swiss roll), and the second eigenfunction is one-to-one with s_2. As a result, they could act as a data-driven parameterization of this nonlinear manifold. The eigenvalues associated with these eigenfunctions are −0.234 and −0.491 compared to the theoretical values of $-\frac{2}{9}$ and $-\frac{1}{2}$ respectively

5.3.1 Parameterizing a Nonlinear Manifold with a Diffusion Process

For this example, the data are generated by a diffusion process on a rectangular domain,

$$\mathrm{d}\boldsymbol{s} = 2\mathrm{d}\boldsymbol{W}_{\mathrm{t}}, \tag{40}$$

where $\boldsymbol{s} = (s_1, s_2)$ is the state and $\boldsymbol{W}_{\mathrm{t}}$ is a two-dimensional Wiener process with $s_1 \in [0, 3\pi]$ and $s_2 \in [0, 2\pi]$. No-flux boundary conditions are imposed at the domain edges. If one had access to the true variables, SKO could be written as

$$\mathcal{K}\psi = 2\partial_{s_1}^2 \psi + 2\partial_{s_2}^2 \psi, \tag{41}$$

also with no-flux boundary conditions; in this particular problem, the SKO is self-adjoint and therefore equivalent to the Perron–Frobenius operator. The eigenfunctions are $\varphi_{ij} = \cos(is_1/3)\cos(js_2/2)$ with the eigenvalues $\lambda_{ij} = -2(i^2/9 + j^2/4)$. Note that the leading eigenfunctions, $\varphi_{1,0}$ and $\varphi_{0,1}$, are $\cos(s_1/3)$ and $\cos(s_2/2)$, which are one-to-one with s_1 and s_2 on $[0, 3\pi]$ and $[0, 2\pi]$, respectively, and could be used to parameterize state space if s_1 and s_2 were not known.

In this example, the true data (i.e., the state expressed in terms of s_1 and s_2) on the rectangle are mapped onto a "Swiss roll" via the transformation

$$\boldsymbol{g}(\boldsymbol{s}) = \begin{bmatrix} (s_1 + 0.1)\cos(s_1) \\ s_2 \\ (s_1 + 0.1)\sin(s_1) \end{bmatrix}, \tag{42}$$

which, among other things, has introduced a new spatial dimension. In all that follows, *the EDMD approach is applied to the three-dimensional, transformed variables and not the two-dimensional, true variables*. Our objective here is to determine a 2-parameter description of what initially appears to be three-dimensional data, directly from the data.

The data given to EDMD were generated by 10^4 initial conditions uniformly distributed in s_1 and s_2 that were evolved for a total time of $\Delta t = 0.1$ using the Euler–Maruyama method with 100 timesteps. Then, both the initial and terminal states of the system were mapped into three dimensions using (42). Next, a dictionary must be defined. However, $\mathcal{M}$ is unknown (indeed, parameterizing $\mathcal{M}$ is the entire point), so the domain Ω is taken to be the "box" in $\mathbb{R}^3$ such that $x \in [-3\pi - 0.1, 3\pi + 0.1]$, $y \in [0, 2\pi]$ and $z \in [-3\pi - 0.1, 3\pi + 0.1]$. In this larger domain, the spectral element basis consisting of 4096 rectangular subdomains (16 each in x, y, and z) with up to linear polynomials in each subdomain is employed. Because $\mathcal{M} \subset \Omega$, extraneous and redundant functions are expected, and $\boldsymbol{G}$ is often ill conditioned.

Figure 12 shows the transformed data colored by the first and second non-trivial eigenfunctions of the Koopman operator. Unlike many of the previous examples, there are clear differences in the analytical and computed eigenvalues and eigenfunctions. However, the first eigenfunction is one-to-one with the "arclength" along the Swiss roll (i.e., the s_1 direction), and the second eigenfunction is one-to-one with the "width" (i.e., the s_2 direction). Furthermore, the first two eigenvalues obtained with EDMD, -0.234 and -0.491, arrange the eigenfunctions in the correct order and compare favorably with the true eigenvalues of $-2/9$ and $-1/2$. Therefore, while our computed Koopman eigenfunctions may not be highly accurate, they are accurate enough to parameterize the nonlinear manifold, which was the goal.

The procedure for incorporating new data points is simple: The embedding for any $\tilde{\boldsymbol{x}} \in \mathcal{M}$ can be obtained simply by evaluating the relevant eigenfunctions at $\tilde{\boldsymbol{x}}$. It should be stressed that although the φ are defined on Ω, their value is only meaningful on (or very near) $\mathcal{M}$ because that is where the dynamical system is defined. Therefore, these new points must be elements of $\mathcal{M}$ if the resulting embedding is to have any meaning.

5.3.2 Reducing Multiscale Dynamics

In the previous example, the noise was isotropic, so the dynamics were equally "fast" in both directions. Because we are interested in the most slowly decaying eigenfunctions, the eigenfunction that is one-to-one with s_1 is more important because it decays more slowly than the eigenfunction that is one-to-one with s_2. In this example, we introduce anisotropic diffusion and therefore create "fast" and "slow" directions on the nonlinear manifold. The purpose of this example is to show that EDMD will "reorder" its ranking of the eigenfunctions and recover the slower component before the faster one if the level of anisotropy is large enough.

Our particular example is

$$\mathrm{d}s_1 = \frac{2}{\epsilon}\mathrm{d}W_1, \tag{43a}$$

$$\mathrm{d}s_2 = 2\mathrm{d}W_2, \tag{43b}$$

with $\epsilon = 0.1$, which is again transformed onto the same Swiss roll. Although the domain in s_1 is larger than it is in the s_2 component, the dynamics of s_1 are now significantly faster than those of s_2 due to a much larger amount of "noise." Because the two

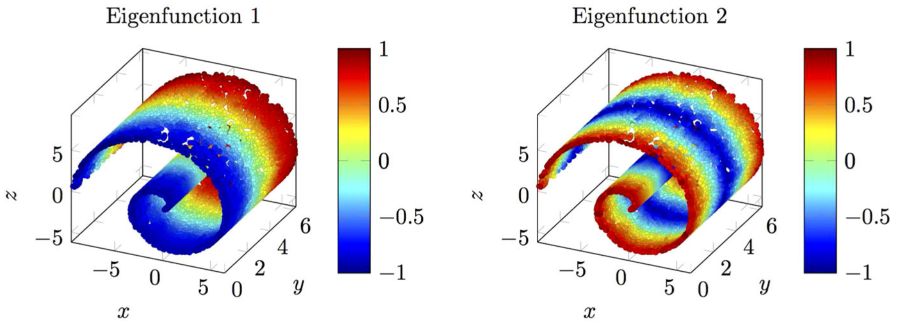


Fig. 13 First two non-trivial Koopman eigenfunctions for the multiscale diffusion process in (43) on the "Swiss roll" computed using 3×10^4 data points. The eigenvalues associated with these eigenfunctions are -0.504 (analytically it has the value $-1/2$) and -2.1 (analytically, -2). In contrast to Fig. 12, these eigenfunctions are a parameterization of the "width" of the Swiss roll (s_2) and a higher harmonic. As a result, the eigenfunctions computed by EDMD take into account both the geometry of the underlying manifold and the dynamics defined on it

random processes in (43) are independent, the eigenfunctions themselves should not change. However, the *eigenvalues* associated with each of the eigenfunctions should.

Figure 13 shows how EDMD captures this difference in the underlying diffusion process. Before, the first two non-trivial eigenfunctions were one-to-one with s_1 and s_2 respectively; now, the first is one-to-one with s_2, and the second is a higher harmonic (but still only a function of s_2). The eigenfunction that is one-to-one with s_1 still exists, but it is no longer associated with a leading eigenvalue. Analytically, its eigenvalue is $-20/9$ though EDMD computes a value of -2.3. Therefore, it is no longer a leading eigenfunction, but still is approximated by the EDMD procedure.

This section explored the application of the EDMD method to data taken from a Markov process. Algorithmically, the method remains the same regardless of how the data were generated, but as demonstrated here, EDMD computes an approximation of the tuples associated with SKO rather than the Koopman operator. To demonstrate the effectiveness of EDMD, we applied the method to a simple SDE with a double-well potential, and an SDE defined on a Swiss roll, which is a nonlinear manifold often used as a benchmark for manifold learning techniques. One advantage of the Koopman approach for applications such as manifold learning or model reduction is that the Koopman tuples take into account both the geometry of the manifold, through the eigenfunction and mode, and the dynamics, through the eigenvalue. As a result, the approach taken here is aware of both geometry and dynamics and does not focus solely on one or the other.

6 Conclusions

In this manuscript, we presented a data-driven method that computes approximations of the Koopman eigenvalues, eigenfunctions, and modes (what we call Koopman tuples) directly from a set of snapshot pairs. We refer to this method as extended dynamic mode decomposition (EDMD). The finite-dimensional approximation generated by

EDMD is the solution to a least-squares problem and converges to a Galerkin method with a large amount of data. While the usefulness of the Galerkin method depends on the sampling density and dictionary selected, several "common sense" choices of both appear to produce useful results.

We demonstrate the effectiveness of the method with four examples: two examples dealt with deterministic data and two with stochastic data. First, we applied EDMD to a linear system where the Koopman eigenfunctions are known analytically. Direct comparison of the EDMD eigenfunctions and the analytic values demonstrated that EDMD can be highly accurate with the proper choice of data and dictionary. Next, we applied EDMD to the unforced Duffing equation, for which the Koopman eigenfunctions are not known explicitly. Although more data will increase the accuracy of the resulting eigenfunctions, they appeared to be accurate enough to effectively partition the domain of interest and parameterize the resulting partitions.

The final two examples used data generated by Markov processes. First, we applied EDMD to data taken from an SDE with a double-well potential and demonstrated the accuracy of the method by comparing those results with a direct numerical approximation of the stochastic Koopman operator over a range of diffusion parameters. Next, we applied EDMD to data from a diffusion process on a "Swiss roll," which is a nonlinear manifold commonly used as an example for nonlinear dimensionality reduction. Similar to those methods (see e.g., Coifman and Lafon (2006) and Lee and Verleysen (2007)), EDMD generated an effective parameterization of the manifold using the leading eigenfunctions. By making the diffusion anisotropic, we then demonstrated that EDMD extracts a parameterization that is dynamically, rather than only geometrically, meaningful. Due to the simplicity of this problem, the eigenfunctions remain unchanged despite the anisotropy; the difference appears in the temporal evolution of the eigenfunctions, which is dictated by the corresponding set of eigenvalues. As a result, the purpose of that example was to show that EDMD "ordered" the eigenvalues of each tuple differently.

The Koopman operator governs the evolution of observables defined on the state space of a dynamical system. By judiciously selecting how we observe our system, we can generate *linear models* that are valid on all of (or, at least, a larger subset of) state space rather than just some small neighborhood of a fixed point; this could allow algorithms designed for linear systems to be applied even in nonlinear settings. However, the tuples of eigenvalues, eigenfunctions, and modes required to do so are decidedly non-trivial to compute. Data-driven methods such as EDMD have the potential to allow accurate approximations of these quantities to be computed without knowledge of the underlying dynamics or geometry. As a result, they could be a practical method for enabling Koopman-based analysis and model reduction in large nonlinear systems.

Acknowledgments The authors would like to thank Igor Mezić, Jonathan Tu, Maziar Hemati, and Scott Dawson for interesting and useful discussions on dynamic mode decomposition and the Koopman operator. M.O.W. gratefully acknowledges support from NSF DMS-1204783. I.G.K acknowledges support from AFOSR FA95550-12-1-0332 and NSF CMMI-1310173. C.W.R acknowledges support from AFOSR FA9550-12-1-0075.

Appendix: EDMD with Redundant Dictionaries

In this appendix, we present a simple example of applying EDMD to a problem where the elements of $\mathcal{D}$ contain redundancies (i.e., the elements of $\mathcal{D}$ are not a basis for $\mathcal{F}_{\mathcal{D}} \subset \mathcal{F}$). Given full knowledge of the underlying dynamical system, one would always choose the elements of $\mathcal{D}$ to be a basis for $\mathcal{F}_{\mathcal{D}}$, but due to our ignorance of $\mathcal{M}$, a redundant set of functions may be chosen. Our objective here is to demonstrate that accurate results can still be obtained even if such a choice is made. To separate quadrature errors from errors resulting from our choice of $\mathcal{D}$, we assume that M is large enough that the EDMD method has already converged to a Galerkin method in that the residual is orthogonal to the space spanned by $\mathcal{D}$.

For the purposes of demonstration, we replace $\mathcal{K}$ with $\mathcal{L} = \partial_s^2$, the Laplace–Beltrami operator defined on the manifold $\mathcal{M}$, where $(x, y) = (s, s)$ for $s \in [0, 2\pi)$ with periodic boundary conditions, which would correspond to, say, the EDMD procedure applied to a diffusion process on a periodic domain. A useful basis for this problem would be $\tilde{\psi}_k(x, y) = \exp(\imath ks) = \exp(\imath k(x + y))$, but without prior knowledge of $\mathcal{M}$, it is difficult to determine this choice should be made. Because the problem appears two-dimensional, one may choose a dictionary whose elements have the form $\psi_{m,n}(x, y) = \exp(\imath mx + \imath ny)$, which contains the $\tilde{\psi}$ but is not linearly independent on $\mathcal{M}$. The indexes we use for the set of functions are $\psi_k(x, y) = \psi_{m,n}(x, y)$ with $m = (k \mod K) - K/2$ and $n = \left\lfloor \frac{k}{K} \right\rfloor - K/2$ with $k = 0, 1, \ldots, K^2$. Here, $K \in \mathbb{N}$ is the total number of basis functions in a single spatial dimension.

Following (12), the i, jth element of $\boldsymbol{G}$ is

$$\begin{aligned} \boldsymbol{G}_{i,j} &= \int_{\mathcal{M}} \psi_i(\boldsymbol{x})^* \psi_j(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_0^{2\pi} e^{\imath((m_j - m_i)s + (n_j - n_i)s)} \,\mathrm{d}s \\ &= \begin{cases} 2\pi & m_j + n_j - m_i - n_i = 0, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{44}$$

Similarly,

$$\begin{aligned} \boldsymbol{A}_{i,j} &= \int_{\mathcal{M}} \psi_i(\boldsymbol{x})^* \partial_s^2 \psi_j(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} = \int_0^{2\pi} -(m_j + n_j)^2 e^{\imath((m_j - m_i)s + (n_j - n_i)s)} \,\mathrm{d}s \\ &= \begin{cases} -2\pi(m_j + n_j)^2 & m_j + n_j - m_i - n_i = 0, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{45}$$

The diagonal structure we would normally have has been replaced with a more complex sparsity pattern, and it has a large nullspace (when $K = 8$, the nullspace is 50-dimensional). To reiterate, there are no advantages to this choice; the redundancies in $\mathcal{D}$ appear due to ignorance about the nature of $\mathcal{M}$, which is the expected situation. Because $\boldsymbol{G}$ is singular, the use of the pseudoinverse in (12) is *critical* to obtain a unique solution.

However, once this is done, there is excellent agreement between the leading eigenfunctions and eigenvalues of $\mathcal{L}$ and those computed using EDMD; this is shown in

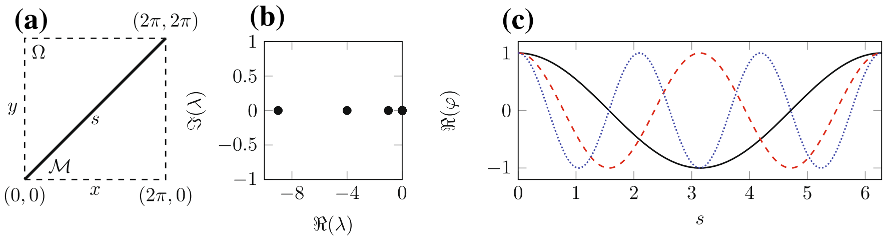


Fig. 14 **a** A sketch of the manifold $s \mapsto (s, s)$ where our dynamical system is defined, and the larger domain, Ω, on which the elements of $\mathcal{D}$ are defined. **b** A plot of the leading 56 eigenvalues of ∂_s^2 computed using EDMD; the redundant functions have increased the dimension of the nullspace from 1 to 50, but accurately capture the pairs of eigenvalues at $-k^2$ for $k = 0, 1, 2, \ldots, 8$. **c** A plot of the real part of the first three non-trivial eigenfunctions shown in *black*, *red*, and *blue* respectively; as expected, they are equivalent to $\cos(ks)$. The imaginary component of the eigenfunctions, which is not shown, captures the $\sin(ks)$ terms (Color figure online)

Fig. 14. The nonzero eigenvalues are quantitatively correct; in particular, pairs of eigenvalues of the form $\lambda = -k^2$ are obtained up until $k = 8$ using $K = 8$. Although the maximum (absolute) value of m or n is only 4, it is clear that the superposition of these functions on $\mathcal{M}$ mimics $k = 8$ modes. The associated eigenfunctions are shown in Fig. 14c; again, there is excellent agreement between the analytic solution (i.e., $\exp(-\imath ks)$) and the EDMD computed solution.

The resulting eigenfunctions can also be evaluated for $(x, y) \notin \mathcal{M}$, but the functions have no dynamical meaning there. Indeed, their value is determined entirely by the regularization used and has no relationship to the underlying dynamical system, which is defined solely on $\mathcal{M}$. This should be contrasted to related works such as Froyland et al. (2014) where the dynamical system is truly defined on Ω, and $\mathcal{M}$ is simply the slow manifold where the eigenfunctions evaluated at $(x, y) \notin \mathcal{M}$ are meaningful as they contain information about the fast dynamics of the system.

Overall, the performance of the EDMD procedure is dependent upon the subspace, $\mathcal{F}_{\mathcal{D}}$ and not the precise choice of $\mathcal{D}$. There are numerical advantages of choosing $\mathcal{D}$ to be a basis for $\mathcal{F}_{\mathcal{D}}$, but in many circumstances this cannot be done without prior knowledge of $\mathcal{M}$. As a result, there are likely benefits to combining EDMD with manifold learning techniques [see, e.g., Coifman and Lafon (2006), Lee and Verleysen (2007), Erban et al. (2007) and Sirisup et al. (2005)]. These methods can numerically approximate $\mathcal{M}$, which could allow a more effective choice of the elements of $\mathcal{D}$ and their associated numerical benefits. As shown here, these methods are not essential to the algorithm, but $\mathcal{M}$ must be identified through some means if EDMD is to be used for more than just data analysis.

References

Bagheri, S.: Effects of weak noise on oscillating flows: linking quality factor, Floquet modes and Koopman spectrum. Phys. Fluids. **26**, 094104 (2014)

Bagheri, S.: Koopman-mode decomposition of the cylinder wake. J. Fluid Mech. **726**, 596–623 (2013)

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., Krysl, P.: Meshless methods: an overview and recent developments. Comput. Methods Appl. Mech. Eng. **139**(1), 3–47 (1996)

Bishop, C.M., et al.: Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, New York (2006)

Bollt, E.M., Santitissadeekorn, N.: Applied and Computational Measurable Dynamics. SIAM, Philadelphia (2013)

Boyd, J.P.: Chebyshev and Fourier Spectral Methods. Courier Dover Publications, New York (2013)

Budišić, M., Mohr, R., Mezić, I.: Applied Koopmanism. Chaos Interdiscip. J. Nonlinear Sci. **22**(4), 047510 (2012)

Budisic, M., Mezic, I.: Geometry of the ergodic quotient reveals coherent structures in flows. Phys. D **241**, 1255–1269 (2012)

Chen, K.K., Tu, J.H., Rowley, C.W.: Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses. J. Nonlinear Sci. **22**(6), 887–915 (2012)

Coifman, R.R., Lafon, S.: Diffusion maps. Appl. Comput. Harmonic Anal. **21**(1), 5–30 (2006)

Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)

Dellnitz, M., Froyland, G., Junge, O.: The algorithms behind GAIO—set oriented numerical methods for dynamical systems. In: Bernold Fiedler (ed.) Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems, pp. 145–174, Springer, Berlin (2001)

Dsilva, C.J., Talmon, R., Rabin, N., Coifman, R.R., Kevrekidis, I.G.: Nonlinear intrinsic variables and state reconstruction in multiscale simulations. J. Chem. Phys. **139**(18), 184109 (2013)

Eisenhower, B., Maile, T., Fischer, M., Mezić, I.: Decomposing building system data for model validation and analysis using the Koopman operator. In: Proceedings of the National IBPSAUSA Conference, New York, USA (2010)

Erban, R., Frewen, T.A., Wang, X., Elston, T.C., Coifman, R., Nadler, B., Kevrekidis, I.G.: Variable-free exploration of stochastic models: a gene regulatory network example. J. Chem. Phys. **126**(15), 155103 (2007)

Froyland, G., Gottwald, G.A., Hammerlindl, A.: A computational method to extract macroscopic variables and their dynamics in multiscale systems. SIAM J. Appl. Dyn. Sys. **13**(4), 1816–1846 (2014)

Froyland, G.: Statistically optimal almost-invariant sets. Phys. D Nonlinear Phenom. **200**(3), 205–219 (2005)

Froyland, G., Padberg, K., England, M.H., Treguier, A.M.: Detection of coherent oceanic structures via transfer operators. Phys. Rev. Lett. **98**(22), 224503 (2007)

Froyland, G., Padberg, K.: Almost-invariant sets and invariant manifolds—connecting probabilistic and geometric descriptions of coherent structures in flows. Phys. D Nonlinear Phenom. **238**(16), 1507–1523 (2009)

Gaspard, P., Nicolis, G., Provata, A., Tasaki, S.: Spectral signature of the pitchfork bifurcation: Liouville equation approach. Phys. Rev. E **51**(1), 74 (1995)

Gaspard, P., Tasaki, S.: Liouvillian dynamics of the Hopf bifurcation. Phys. Rev. E **64**(5), 056232 (2001)

Givon, D., Kupferman, R., Stuart, A.: Extracting macroscopic dynamics: model problems and algorithms. Nonlinearity **17**(6), R55 (2004)

Hansen, P.C.: Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank. SIAM J. Sci. Stat. Comput. **11**(3), 503–518 (1990)

Higham, D.J.: An algorithmic introduction to numerical simulation of stochastic differential equations. SIAM Rev. **43**(3), 525–546 (2001)

Hirsch, C.: Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics vol. 1, Butterworth-Heinemann (2007)

Holmes, P., Lumley, J.L., Berkooz, G.: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambridge University Press, Cambridge (1998)

Jovanović, M.R., Schmid, P.J., Nichols, J.W.: Sparsity-promoting dynamic mode decomposition. Phys. Fluids **26**, 024103 (2014)

Juang, J.-N.: Applied System Identification. Prentice Hall, Englewood Cliffs (1994)

Karniadakis, G., Sherwin, S.: Spectral/Hp Element Methods for Computational Fluid Dynamics. Oxford University Press, Oxford (2013)

Kloeden, P.E., Platen, E.: Numerical solution of stochastic differential equations. Springer, Berlin (1992)

Koopman, B.O.: Hamiltonian systems and transformation in Hilbert space. Proc. Natl. Acad. Sci. U. S. A. **17**(5), 315 (1931)

Koopman, B.O., Neumann, J.V.: Dynamical systems of continuous spectra. Proc. Natl. Acad. Sci. U. S. A. **18**(3), 255 (1932)

Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Springer, Berlin (2007)

Lehoucq, R.B., Sorensen, D.C., Yang, C.: ARPACK users' guide: Solution of large-scale eigenvalue problems. SIAM, Philadelphia (1998)

Liu, G.-R.: Meshfree methods: Moving beyond the finite element method. CRC Press, Boca Raton (2010)

Matkowsky, B., Schuss, Z.: Eigenvalues of the Fokker-Planck operator and the approach to equilibrium for diffusions in potential fields. SIAM J. Appl. Math. **40**(2), 242–254 (1981)

Mauroy, A., Mezic, I.: A spectral operator-theoretic framework for global stability. In: Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on, pp. 5234–5239 (2013)

Mauroy, A., Mezić, I., Moehlis, J.: Isostables, isochrons, and Koopman spectrum for the action-angle representation of stable fixed point dynamics. Phys. D Nonlinear Phenom. **261**, 19–30 (2013)

Mauroy, A., Mezić, I.: On the use of Fourier averages to compute the global isochrons of (quasi) periodic dynamics. Chaos Interdiscip. J. Nonlinear Sci. **22**(3), 033112 (2012)

Mezić, I.: Spectral properties of dynamical systems, model reduction and decompositions. Nonlinear Dyn. **41**(1–3), 309–325 (2005)

Monaghan, J.J.: Smoothed particle hydrodynamics. Annu. Rev. Astron. Astrophys. **30**, 543–574 (1992)

Muld, T.W., Efraimsson, G., Henningson, D.S.: Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition. Comput. Fluids **57**, 87–97 (2012)

Nadler, B., Lafon, S., Kevrekidis, I.G., Coifman, R.R.: Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. Adv Neural Inf Process Syst. **18**, 955–962 (2005)

Nadler, B., Lafon, S., Coifman, R.R., Kevrekidis, I.G.: Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. Appl. Comput. Harmonic Anal. **21**(1), 113–127 (2006)

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

Santitissadeekorn, N., Bollt, E.: The infinitesimal operator for the semigroup of the Frobenius-Perron operator from image sequence data: vector fields and transport barriers from movies. Chaos Interdiscip. J. Nonlinear Sci. **17**(2), 023126 (2007)

Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. **65**(6), 5–28 (2010)

Schmid, P., Li, L., Juniper, M., Pust, O.: Applications of the dynamic mode decomposition. Theor. Comput. Fluid Dyn. **25**(1–4), 249–259 (2011)

Schmid, P.J., Violato, D., Scarano, F.: Decomposition of time-resolved tomographic PIV. Exp. Fluids **52**(6), 1567–1579 (2012)

Seena, A., Sung, H.J.: Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations. Int. J. Heat Fluid Flow **32**(6), 1098–1110 (2011)

Sirisup, S., Karniadakis, G.E., Xiu, D., Kevrekidis, I.G.: Equation-free/Galerkin-free POD-assisted computation of incompressible flows. J. Comput. Phys. **207**(2), 568–587 (2005)

Stengel, R.F.: Optimal Control and Estimation. Courier Dover Publications, New York (2012)

Susuki, Y., Mezic, I.: Nonlinear Koopman modes and power system stability assessment without models. IEEE Trans. Power Syst. **29**(2), 899–907 (2014)

Susuki, Y., Mezić, I.: Nonlinear Koopman modes and coherency identification of coupled swing dynamics. IEEE Trans. Power Syst. **26**(4), 1894–1904 (2011)

Susuki, Y., Mezić, I.: Nonlinear Koopman modes and a precursor to power system swing instabilities. Power Syst. IEEE Trans. **27**(3), 1182–1191 (2012)

Todorov, E.: Optimal control theory. In: Bayesian brain: Probabilistic approaches to neural coding, Kenji Doya (Editor), pp. 269–298. MIT Press, Cambridge (2007)

Trefethen, L.N.: Spectral Methods in MATLAB, vol. 10. SIAM, Philadelphia (2000)

Tu, J.H., Rowley, C.W., Luchtenburg, D.M., Brunton, S.L., Kutz, J.N.: On dynamic mode decomposition: Theory and applications. J Comput Dyn. **1**(2), 391–421 (2014)

Wendland, H.: Meshless Galerkin methods using radial basis functions. Math. Comput. Am. Math. Soc. **68**(228), 1521–1531 (1999)

Wynn, A., Pearson, D., Ganapathisubramani, B., Goulart, P.: Optimal mode decomposition for unsteady flows. J. Fluid Mech. **733**, 473–503 (2013)