

A KERNEL-BASED METHOD FOR DATA-DRIVEN KOOPMAN SPECTRAL ANALYSIS

MATTHEW O. WILLIAMS

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

CLARENCE W. ROWLEY

Department of Mechanical and Aerospace Engineering
Princeton University
Princeton, NJ 08544, USA

IOANNIS G. KEVREKIDIS

Department of Chemical and Biological Engineering & PACM
Princeton University
Princeton, NJ 08544, USA

(Communicated by the associate editor name)

ABSTRACT. A data-driven, kernel-based method for approximating the leading Koopman eigenvalues, eigenfunctions, and modes in problems with high-dimensional state spaces is presented. This approach approximates the Koopman operator using a set of scalar observables, which are functions that map states to scalars, that is determined *implicitly* by the choice of a kernel function. This circumvents the computational issues that arise due to the number of basis functions required to span a “sufficiently rich” subspace of the space of scalar observables in such applications. We illustrate this method on two examples: the FitzHugh-Nagumo PDE, a prototypical one-dimensional reaction-diffusion system, and vorticity data obtained from experimentally obtained velocity data for flow over a cylinder at Reynolds number 413. In both examples, we compare our results with related methods, such as Dynamic Mode Decomposition (DMD) that have the same cost as our approach.

1. Introduction. In many applications, the evolution of complex spatio-temporal phenomena can be characterized using models based on the interactions between a relatively small number of modes. Due to the availability of data and computational power, algorithmic techniques for identifying such modes have become increasingly common [1–14]. Perhaps the best known method is the Proper Orthogonal Decomposition (POD) [1, 2, 15], which is also known as Principal Component Analysis (PCA) [14, 15]. However, other algorithms that generate different sets of modes exist. In recent years, approximations of the modes of the Koopman operator [16, 17] have become popular in fluid applications [5, 6, 8, 10, 18–20]. The Koopman modes of the full state observable, which are vectors for systems of ODEs or spatial profiles

2010 *Mathematics Subject Classification.* Primary: 37M10, 65P99, 47B33.

Key words and phrases. Koopman operator, Dynamic Mode Decomposition, kernel methods, time series analysis, machine learning.

for PDEs, are intrinsic to a particular evolution law, and have temporal dynamics that are determined by their corresponding Koopman eigenvalues [5]; in effect, the Koopman modes look and act like the eigenvectors of a linear system even when the underlying evolution law is nonlinear.

This representation is possible because the Koopman operator, which defines these quantities, is a *linear* operator and therefore can have eigenfunctions and eigenvalues, which are later used to define the modes. However, it acts on *scalar observables*, which are functions defined on state space, and is therefore an infinite-dimensional operator even for finite-dimensional dynamical systems. As a result, methods that approximate the Koopman operator (often implicitly) select a finite-dimensional subspace of the space of scalar observables to use during the computation. Currently, the most widely used method is Dynamic Mode Decomposition (DMD) [3–6, 21] (along with its related modifications [8, 10, 20]), which implicitly select *linear* functions of the state as the basis functions [7]. This restrictive choice of basis allows DMD to be applied to large systems of ODEs or PDEs, but in many applications, this subspace is simply not “rich” enough to effectively approximate the Koopman operator [7, 11].

Other methods, such as Extended DMD [7], use an expanded set of observables, which can produce a more accurate approximation of the set of Koopman eigenvalues, eigenfunctions, and modes for nonlinear systems. For small systems of ODEs, a “sufficiently rich” subspace can often be spanned using a few hundred or thousand basis functions, which is computationally tractable even on a laptop. However, the size of the needed basis grows rapidly as the dimension of the state space increases, so the necessary computations quickly become infeasible. This explosion in the computational cost is common in machine learning applications, and one facet of the “curse of dimensionality” [15].

In this manuscript, we introduce a data-driven, kernel-based method to approximate the Koopman operator in systems with large state dimension. This approach circumvents the dimensionality issues encountered by Extended DMD by defining a kernel function that *implicitly* computes inner products in the high-dimensional space of observables. Because we do not form this space explicitly, the computational cost of the method is determined by the number of snapshots and the dimension of state space rather than the number of basis functions used to represent the scalar observables; therefore, the cost of this approach is comparable to that of DMD. We apply this approach to a pair of sample problems: the one-dimensional FitzHugh-Nagumo PDE, a prototypical reaction-diffusion system; and experimentally obtained vorticity data from flow over a cylinder obtained at Reynolds number 413. Using the FitzHugh-Nagumo PDE example, we will demonstrate that using a higher-dimensional subspace of observables can result in more accurate and reproducible approximations of the leading Koopman modes, eigenvalues, and eigenfunctions. Finally, the cylinder example demonstrates that this approach has practical advantages in more realistic settings where the data are noisy and the “true” Koopman modes and eigenvalues are unknown.

The remainder of the manuscript is outlined as follows: in Sec. 2 we present the kernel reformulation of Extended Dynamic Mode Decomposition. In Sec. 3, we apply it to a numerical discretization of the FitzHugh-Nagumo PDE in one spatial dimension, where a subset of the Koopman eigenvalues and modes are known. In Sec. 4, we apply our method to experimentally obtained data of flow over a cylinder,

which is a more realistic example with measurement noise and other experimental realities. Finally, in Sec. 5, we present some concluding remarks and future outlook.

2. A Data-Driven Approximation of the Koopman Operator. In this section, we present a data-driven approach for approximating the Koopman operator that can be applied to systems with high-dimensional state spaces. The method presented here is a reformulation of the Extended DMD procedure that makes use of the so-called *kernel trick* [22]. In the following subsections, we will (i) briefly review the Koopman operator (ii) give a brief derivation of “standard” Extended DMD, which enables the “tuples” of Koopman eigenvalues, eigenfunctions, and modes to be approximated from data; (iii) present the kernel approach; and (iv) give a practical algorithm for computing the leading tuples of eigenvalues, eigenfunctions, and modes.

2.1. The Koopman Operator. The Koopman operator [16, 17], along with its eigenvalues, eigenfunctions, and modes, is defined by a dynamical system and not a set of data. Given the discrete time dynamical system $(n, \mathcal{M}, \mathbf{F})$, where $n \in \mathbb{Z}$ is time, $\mathcal{M} \subseteq \mathbb{R}^N$ is the state space, and $\mathbf{x} \mapsto \mathbf{F}(\mathbf{x})$ defines the dynamics, the action of the Koopman operator, \mathcal{K} , on a *scalar observable*, $\phi : \mathcal{M} \rightarrow \mathbb{C}$, is

$$(\mathcal{K}\phi)(\mathbf{x}) = (\phi \circ \mathbf{F})(\mathbf{x}) = \phi(\mathbf{F}(\mathbf{x})). \quad (1)$$

Intuitively, $\mathcal{K}\phi$ is a new function that gives the value of ϕ “one step in the future”. Note that *the Koopman operator acts on functions of the state, and not the states themselves*. Since $\phi \in \mathcal{F}$, where \mathcal{F} is an appropriate space of scalar observables, the Koopman operator is infinite dimensional. However, it is also linear, and thus it can have eigenvalues and eigenfunctions, which we refer to as Koopman eigenvalues and eigenfunctions. Accompanying the eigenvalues and eigenfunctions are the Koopman modes for a given *vector valued observable*, $\mathbf{g} : \mathcal{M} \rightarrow \mathbb{R}^{N_o}$, where $N_o \in \mathbb{N}$. These modes are vectors in a system of ODEs (or spatial profiles in a PDE) that contain the coefficients required to construct \mathbf{g} using a Koopman eigenfunction basis [5, 7]. One particularly useful set of modes is that of the identity operator or, equivalently, the *full state observable*, $\mathbf{g}(\mathbf{x}) = \mathbf{x}$, which we refer to as simply the *Koopman modes* in all that follows.

In many systems [5, 18, 23], tuples consisting of an eigenvalue, an eigenfunction, and a mode enable a simple yet powerful means of representing the system state and making predictions of future values. In particular,

$$\mathbf{x} = \sum_{k=1}^K \boldsymbol{\xi}_k \varphi_k(\mathbf{x}), \quad \mathbf{F}(\mathbf{x}) = \sum_{k=1}^K \mu_k \boldsymbol{\xi}_k \varphi_k(\mathbf{x}) \quad (2)$$

where $\boldsymbol{\xi}_k$ is the Koopman mode corresponding to the eigenfunction φ_k , μ_k is the corresponding eigenvalue, and K is the number of tuples required for the reconstruction, which could be infinite. The eigenfunction $\varphi_k : \mathcal{M} \rightarrow \mathbb{C}$ is a *function*, but the mode $\boldsymbol{\xi}_k \in \mathbb{C}^N$ is a *vector*. From (2), the coefficient associated with the k -th mode, $\boldsymbol{\xi}_k$, is obtained by evaluating the k -th eigenfunction, φ_k , and the temporal evolution is dictated by μ_k . Ultimately, one benefit of this Koopman-based approach is that the eigenvalues, eigenfunctions, and modes are *intrinsic to the dynamical system*; in the subsequent section they will be approximated using data, but unlike the Proper Orthogonal Decomposition (POD) and related methods, they still exist absent any data.

2.2. Extended Dynamic Mode Decomposition. Extended Dynamic Mode Decomposition (Extended DMD) [7] is a regression procedure whose solution produces a finite-dimensional approximation of the Koopman operator. To obtain this approximation, we define a basis set that consists of K scalar observables, which we denote as ψ_k for $k = 1, \dots, K$, that span $\mathcal{F}_K \subset \mathcal{F}$. We also define the vector valued observable, $\boldsymbol{\psi} : \mathcal{M} \rightarrow \mathbb{R}^K$, where

$$\boldsymbol{\psi}(\mathbf{x}) = \begin{bmatrix} \psi_1(\mathbf{x}) \\ \psi_2(\mathbf{x}) \\ \vdots \\ \psi_K(\mathbf{x}) \end{bmatrix}. \quad (3)$$

In this application, $\boldsymbol{\psi}$ is the mapping from physical space to *feature space*. Any $\phi, \hat{\phi} \in \mathcal{F}_K$ can be written as

$$\phi = \sum_{k=1}^K a_k \psi_k = \boldsymbol{\psi}^T \mathbf{a}, \quad \hat{\phi} = \sum_{k=1}^K \hat{a}_k \psi_k = \boldsymbol{\psi}^T \hat{\mathbf{a}}, \quad (4)$$

for some set of coefficients $\mathbf{a}, \hat{\mathbf{a}} \in \mathbb{C}^K$. Although \mathbf{F} is unknown, we assume access to a data set of snapshot *pairs*:

$$\{(\mathbf{x}_m, \mathbf{y}_m)\}_{m=1}^M, \quad \mathbf{y}_m = \mathbf{F}(\mathbf{x}_m), \quad (5)$$

where $\mathbf{x}_m, \mathbf{y}_m \in \mathcal{M}$. One important special case of such a data set is a single time-series of data, which can be written in the above form by “grouping” sequential pairs of snapshots that were obtained with a fixed sampling interval, say Δt .

Let $\hat{\phi} = \mathcal{K}\phi + r$, where $r \in \mathcal{F}$ is a residual function that appears because \mathcal{F}_K is not necessarily invariant to the action of the Koopman operator. Using the notation in (4), the objective of the Extended DMD procedure [7] is to define a mapping from some given \mathbf{a} to a new vector $\hat{\mathbf{a}}$ that minimizes this residual. Because the Koopman operator is linear, this mapping can be represented by a matrix $\mathbf{K} \in \mathbb{R}^{K \times K}$. To determine the entries of \mathbf{K} , the Extended DMD approach takes ideas from collocation methods typically used to solve PDEs, but uses the \mathbf{x}_m as the collocation points rather than a pre-determined grid [24, 25]. As a result, the finite dimensional approximation is

$$\mathbf{K} \triangleq \boldsymbol{\Psi}_x^+ \boldsymbol{\Psi}_y, \quad (6a)$$

where

$$\boldsymbol{\Psi}_x \triangleq \begin{bmatrix} \boldsymbol{\psi}(\mathbf{x}_1)^T \\ \boldsymbol{\psi}(\mathbf{x}_2)^T \\ \vdots \\ \boldsymbol{\psi}(\mathbf{x}_M)^T \end{bmatrix}, \quad \boldsymbol{\Psi}_y \triangleq \begin{bmatrix} \boldsymbol{\psi}(\mathbf{y}_1)^T \\ \boldsymbol{\psi}(\mathbf{y}_2)^T \\ \vdots \\ \boldsymbol{\psi}(\mathbf{y}_M)^T \end{bmatrix}, \quad (6b)$$

are in $\mathbb{R}^{M \times K}$, and $+$ denotes the pseudoinverse.

In Williams et al. [7], the relationship $\boldsymbol{\Psi}_x^+ = (\boldsymbol{\Psi}_x^T \boldsymbol{\Psi}_x)^+ \boldsymbol{\Psi}_x^T$ was used to rewrite (6) as

$$\mathbf{K} \triangleq \mathbf{G}^+ \mathbf{A}, \quad (7a)$$

$$\mathbf{G} = \boldsymbol{\Psi}_x^T \boldsymbol{\Psi}_x = \sum_{m=1}^M \boldsymbol{\psi}(\mathbf{x}_m) \boldsymbol{\psi}(\mathbf{x}_m)^T, \quad \mathbf{A} = \boldsymbol{\Psi}_x^T \boldsymbol{\Psi}_y = \sum_{m=1}^M \boldsymbol{\psi}(\mathbf{x}_m) \boldsymbol{\psi}(\mathbf{y}_m)^T, \quad (7b)$$

which is advantageous when the number of snapshots is much larger than the number of basis functions, i.e., $M \gg K$, because each term in the sum can be evaluated

individually. As a result, one need only store the matrices $\mathbf{G}, \mathbf{A} \in \mathbb{R}^{K \times K}$ rather than $\Psi_x, \Psi_y \in \mathbb{R}^{M \times K}$, which would be much larger in this regime.

Expressions (6) and (7) are mathematically equivalent, and produce the same matrix \mathbf{K} . Therefore, regardless of how it was computed, the properties of \mathbf{K} of interest here are unchanged:

1. The k -th eigenvalue of \mathbf{K} , μ_k , is an approximation of an eigenvalue of \mathcal{K} . When the data are generated by sampling a continuous time dynamical system with a fixed sampling interval Δt , we also define the approximation of the *continuous time eigenvalue* as $\lambda_k \triangleq \log(\mu_k)/\Delta t$.
2. Using (4), the corresponding eigenvector, \mathbf{v}_k , approximates an *eigenfunction* of the Koopman operator via

$$\varphi_k \triangleq \psi^T \mathbf{v}_k. \quad (8)$$

3. The left eigenvector, \mathbf{w}_k , can be used to approximate the Koopman mode, ξ_k . For a derivation of these relationships, see Williams et al. [7].

2.3. The Kernel Method. In Sec. 2.2, we considered the case where the number of snapshots was large compared to the dimension of the desired subspace of functions. Now we will consider the opposite and more commonly encountered regime [3–6, 8, 10, 18–21] where the number of snapshots is small compared to the dimension of our subspace of scalar observables (i.e., $M \ll K$).

The difficulty here is that the Extended DMD procedure, as formulated in Sec. 2.2, requires a $K \times K$ matrix to be formed and decomposed, which requires $\mathcal{O}(K^2 M)$ and $\mathcal{O}(K^3)$ time respectively, and the value of K for a “rich” set of basis functions grows rapidly as the dimension of state space increases. For instance, consider the case where \mathcal{F}_K is the space of all (multivariate) polynomials on \mathbb{R}^{256} with degree up to 20, as it will be in our first example discussed in Section 3. In this case, $K \sim 10^{30}$, which is far too large for practical computations [15]. This explosion in the size of the problem is common, and is one facet of the *curse of dimensionality*.

Because the matrix \mathbf{K} is the solution to a regression problem, the non-zero eigenvalues and their associated left and right eigenvectors can also be obtained by solving the *dual form* of this problem [15]. To show this, note that $\mathcal{R}(\mathbf{K}) \subseteq \mathcal{R}(\Psi_x^T)$, i.e., the range of Ψ_x^T contains the range of \mathbf{K} . If we could compute the SVD of Ψ_x ,

$$\Psi_x \triangleq \mathbf{Q}\Sigma\mathbf{Z}^T, \quad (9)$$

where $\mathbf{Q}, \Sigma \in \mathbb{R}^{M \times M}$ and $\mathbf{Z} \in \mathbb{R}^{K \times M}$, then an eigenvector of \mathbf{K} with $\mu_k \neq 0$ could be written as $\mathbf{v} = \mathbf{Z}\hat{\mathbf{v}}$ for some $\hat{\mathbf{v}} \in \mathbb{C}^M$. With some simple algebraic manipulations, the eigenvalue problem can be written as:

$$\begin{aligned} \mu \mathbf{v} &= \mathbf{K} \mathbf{v} \\ \iff \mu \mathbf{Z} \hat{\mathbf{v}} &= \mathbf{Z} \Sigma^+ \mathbf{Q}^T \Psi_y \mathbf{Z} \hat{\mathbf{v}} \\ &= \mathbf{Z} (\Sigma^+ \mathbf{Q}^T) (\Psi_y \mathbf{Z}) \hat{\mathbf{v}} \\ &= \mathbf{Z} [(\Sigma^+ \mathbf{Q}^T) (\Psi_y \Psi_x^T) (\mathbf{Q} \Sigma^+)] \hat{\mathbf{v}}. \end{aligned}$$

Therefore, an alternative method for computing an eigenvector of \mathbf{K} is to form the matrix

$$\hat{\mathbf{K}} \triangleq (\Sigma^+ \mathbf{Q}^T) \hat{\mathbf{A}} (\mathbf{Q} \Sigma^+), \quad (10)$$

where $\hat{\mathbf{A}} \triangleq \Psi_y \Psi_x^T$, compute an eigenvector of $\hat{\mathbf{K}}$, say $\hat{\mathbf{v}}$, and set $\mathbf{v} = \mathbf{Z}\hat{\mathbf{v}}$. Here $\hat{\mathbf{K}} \in \mathbb{R}^{M \times M}$, so the computational cost of the decomposition is determined by the

number of snapshots rather than the dimension of the system state or “feature” space.

The benefit of the expression in $\hat{\mathbf{K}}$ is that all the required matrices can be obtained by computing inner products in feature space. In addition to $\hat{\mathbf{A}}$, we define the matrix $\hat{\mathbf{G}} \triangleq \Psi_x \Psi_x^T$. Despite appearing “flipped,” the ij -th elements of $\hat{\mathbf{G}}$ and $\hat{\mathbf{A}}$ are

$$\hat{\mathbf{G}}^{(ij)} \triangleq \psi(\mathbf{x}_j)^T \psi(\mathbf{x}_i), \quad \hat{\mathbf{A}}^{(ij)} \triangleq \psi(\mathbf{x}_j)^T \psi(\mathbf{y}_i). \quad (11)$$

However $\hat{\mathbf{G}} = \mathbf{Q}\Sigma^2\mathbf{Q}^T$, using the definition of \mathbf{Q} and Σ in (9). Therefore, given $\hat{\mathbf{G}}$ we can obtain \mathbf{Q} and Σ via its eigen-decomposition, which is simply the *method of snapshots* approach for computing the POD modes from snapshot data [26]. As a result, we *could* compute $\hat{\mathbf{K}}$ by forming $\hat{\mathbf{G}}$ and $\hat{\mathbf{A}}$ using (11) in $\mathcal{O}(M^2K)$ time. This is a large improvement over “standard” Extended DMD, but still impractical as K can be extremely large.

Rather than *explicitly defining* the function ψ and computing the entries of $\hat{\mathbf{K}}$ directly, the *kernel trick* is a common technique for *implicitly* computing inner products [22, 27–29], which can be used to assemble $\hat{\mathbf{K}}$ in $\mathcal{O}(M^2N)$ time. Instead of defining ψ , we define a *kernel function* $f: \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$ that computes inner products in feature space given pairs of data points; that is, $f(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \psi(\mathbf{x}_j)^T \psi(\mathbf{x}_i)$ [27]. In effect, the choice of f defines ψ , which is equivalent to choosing the basis in Extended DMD. It is, however, crucial to note that f *does not compute these inner products directly*. The simplest example is the polynomial kernel

$$f(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{z}^T \mathbf{x})^2 \quad (12)$$

with $\mathbf{x}, \mathbf{z} \in \mathbb{R}^2$, which, when expanded, is

$$\begin{aligned} f(\mathbf{x}, \mathbf{z}) &= (1 + x_1 z_1 + x_2 z_2)^2 \\ &= (1 + 2x_1 z_1 + 2x_2 z_2 + 2x_1 x_2 z_1 z_2 + x_1^2 z_1^2 + x_2^2 z_2^2) \\ &= \psi(\mathbf{z})^T \psi(\mathbf{x}) \end{aligned}$$

if $\psi(\mathbf{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2]$. In general, a polynomial kernel of the form

$$f(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{z}^T \mathbf{x})^\alpha \quad (13)$$

is equivalent to a basis that can represent all polynomials up to and including terms of degree α , and takes only $\mathcal{O}(N)$ time to evaluate.

In the example that follows, we use a polynomial kernel with $\alpha = 20$. We selected a polynomial kernel because the Koopman eigenfunctions are often analytic in a disk about a fixed point [30], and polynomial kernels mimic an α order power-series expansion. The specific choice of $\alpha = 20$ is more *ad hoc*. In general, large values of α use a “richer” set of basis functions, but also deleteriously impact the condition number of $\hat{\mathbf{G}}$. Other choices of kernels, such as Gaussian kernels (i.e., $f(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/\sigma^2)$), are also used in machine learning applications [15, 22, 31], and may result in better performance (both in terms of numerical conditioning and in approximating the Koopman operator) [32]. The optimal choice of kernel, which is equivalent to the ideal choice of the basis set, remains an open question.

Ultimately, the procedure for obtaining an approximation of the Koopman operator is as follows:

1. Using the data set of snapshot pairs and the kernel function, f , compute the elements of $\hat{\mathbf{G}}$ and $\hat{\mathbf{A}}$ using:

$$\hat{\mathbf{G}}^{(ij)} \triangleq f(\mathbf{x}_i, \mathbf{x}_j), \quad \hat{\mathbf{A}}^{(ij)} \triangleq f(\mathbf{y}_i, \mathbf{x}_j). \quad (14)$$

2. Compute the eigendecomposition of the Gramian $\hat{\mathbf{G}}$ to obtain \mathbf{Q} and Σ .
3. Construct $\hat{\mathbf{K}}$ using (10).

The difference between (11) and (14) is that the former defines ψ explicitly, and as a result, computes the set of the needed inner products in $\mathcal{O}(M^2K)$ time, while the latter defines ψ implicitly, which allows the inner products to be computed in $\mathcal{O}(M^2N)$ time. If a linear kernel, $f(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y}$ is chosen, the kernel approach outlined here is identical to DMD [11], where the subspace of observables used to approximate the Koopman eigenfunctions is chosen using the Proper Orthogonal Decomposition. The overarching idea here is that the kernel approach chooses this subspace differently by using what is in effect *Kernel Principal Component Analysis* [15], which exploits the kernel trick to make the computation efficient when $M \ll K$.

2.4. Computing the Koopman Eigenvalues, Modes, and Eigenfunctions.

In this subsection, we show how to approximate the Koopman eigenvalues, modes, and eigenfunctions given $\hat{\mathbf{K}}$. Let $\hat{\mathbf{V}}$ be the matrix whose columns are the eigenvectors of $\hat{\mathbf{K}}$. Then using (8), we define the matrix of eigenfunctions values:

$$\begin{aligned} \Phi_x &\triangleq \Psi_x \mathbf{Z} \hat{\mathbf{V}} = (\Psi_x \Psi_x^T) (\mathbf{Q} \Sigma^+) \hat{\mathbf{V}} = \hat{\mathbf{G}} (\mathbf{Q} \Sigma^+) \hat{\mathbf{V}} = (\mathbf{Q} \Sigma^2 \mathbf{Q}^T) (\mathbf{Q} \Sigma^+) \hat{\mathbf{V}} \\ &= \mathbf{Q} \Sigma_r \hat{\mathbf{V}} \end{aligned} \quad (15)$$

where Σ_r is the diagonal matrix of singular values with any entry neglected in the pseudo-inverse set to zero. The i -th row of Φ_x contains the numerically computed eigenfunctions evaluated at \mathbf{x}_i . The k -th numerically approximated Koopman eigenfunction can also be evaluated at a new data point via:

$$\begin{aligned} \varphi_k(\mathbf{x}) &= (\psi(\mathbf{x})^T \Psi_x^T) (\mathbf{Q} \Sigma^+ \hat{\mathbf{v}}_k) \\ &= [f(\mathbf{x}, \mathbf{x}_1) \quad f(\mathbf{x}, \mathbf{x}_2) \quad \cdots \quad f(\mathbf{x}, \mathbf{x}_M)] (\mathbf{Q} \Sigma^+ \hat{\mathbf{v}}_k), \end{aligned} \quad (16)$$

using the same arguments as in (15) though without the simplifications and cancellations that occur in that case.

To compute the Koopman modes, we use (2), which when evaluated at each of the data points, results in the matrix equation

$$\mathbf{X} = \Phi_x \Xi, \quad (17)$$

where

$$\mathbf{X} \triangleq \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_M^T \end{bmatrix}, \quad \Xi \triangleq \begin{bmatrix} \xi_1^T \\ \xi_2^T \\ \vdots \\ \xi_M^T \end{bmatrix} = \Phi_x^+ \mathbf{X} = \hat{\mathbf{V}}^{-1} \Sigma^+ \mathbf{Q}^T \mathbf{X}. \quad (18)$$

provided that $\hat{\mathbf{V}}$ is full rank. In this case,

$$\hat{\mathbf{V}}^{-1} = \begin{bmatrix} \hat{\mathbf{w}}_1^* \\ \hat{\mathbf{w}}_2^* \\ \vdots \\ \hat{\mathbf{w}}_M^* \end{bmatrix}, \quad (19)$$

where $\hat{\boldsymbol{w}}^*$ is a left eigenvector of $\hat{\boldsymbol{K}}$ scaled so that $\hat{\boldsymbol{w}}_i^* \hat{\boldsymbol{v}}_j = \delta_{ij}$. This implies that the k -th Koopman mode, $\boldsymbol{\xi}_k$, is

$$\boldsymbol{\xi}_k = (\hat{\boldsymbol{w}}_k^* \boldsymbol{\Sigma}^+ \boldsymbol{Q}^T \boldsymbol{X})^T, \quad (20)$$

and therefore approximate Koopman tuples can be obtained via the left and right eigenvectors of $\hat{\boldsymbol{K}}$, and do not require $\hat{\boldsymbol{K}}$ to be computed in its entirety. For the problems considered here, the complete decomposition of $\hat{\boldsymbol{K}}$ is computed, but for problems with larger numbers of snapshots, Krylov methods could be used to compute a leading subset of its eigenvalues and vectors [33].

3. Example: The FitzHugh-Nagumo PDE. In order to highlight the effectiveness of the kernel method, we first apply it and, as a benchmark, DMD to the FitzHugh-Nagumo PDE [34] in one spatial dimension. This example is particularly useful, because a subset of the true Koopman eigenvalues and modes can be deduced from the system linearization. The governing equations are:

$$\partial_t v = \partial_{xx} v + v - w - v^3, \quad (21a)$$

$$\partial_t w = \delta \partial_{xx} w + \epsilon(v - c_1 w - c_0), \quad (21b)$$

where v is the activator field, w is the inhibitor field, $c_0 = -0.03$, $c_1 = 2.0$, $\delta = 4.0$, $\epsilon = 0.02$, for $x \in [0, 20]$ with Neumann boundary conditions. Both v and w are approximated using a discrete cosine transform-based spectral method with 128 basis functions each.

With these parameter values, (21) has a stable equilibrium point that resembles a standing wave-front in both the activator and inhibitor fields. In order to explore a subset of state space and to mimic the presence of impulsive actuation, every 25 time units we perturb the system state by setting

$$v(x, t) \leftarrow v(x, t) + \sum_{i=1}^3 u_i \exp(-(x - x_i)^2), \quad (22)$$

where $x_1 = 7.5$, $x_2 = 10$, $x_3 = 12.5$ and $u_1, u_2, u_3 \sim \mathcal{N}(0, 0.1^2)$. Note that $w(x, t)$ remains unchanged. Starting from the equilibrium point, we generate five separate trajectories consisting of 2500 snapshots (or 2499 snapshot pairs) with a sampling interval of $\Delta t = 1$ starting with five different random seeds. Figure 1 shows the v and w data for a prototypical trajectory: the perturbations applied even 25 units in time excite “fast” directions that decay almost immediately (and thus are not visible in the figure) as well as “slow” directions that take the form of “shifts” in the position of the wave front.

To compute the Koopman eigenvalues, eigenfunctions, and modes, we use a polynomial kernel with $\alpha = 20$. Furthermore, we scale each data set so that the mean value of $\|\boldsymbol{x}_m\| = 1$, which is equivalent to introducing a scaling parameter in the kernel f . For these data sets, both \boldsymbol{X} (for DMD) and $\boldsymbol{\Psi}_x$ (for the kernel method) have small singular values, so when we compute the pseudoinverse of \boldsymbol{X} (for DMD) or $\boldsymbol{\Sigma}$ (for the kernel method), we retain the 150 largest singular values and set the others to zero. This is equivalent to projecting \boldsymbol{X} and \boldsymbol{Y} onto the 150 leading POD modes of \boldsymbol{X} or $\boldsymbol{\Psi}_x$ and $\boldsymbol{\Psi}_y$ onto the leading 150 kernel principal components of $\boldsymbol{\Psi}_x$ before carrying out the computation. In our experience, this often helps to avoid the spurious unstable eigenvalues that both DMD and Extended DMD can produce.

Figure 2 shows the approximation of the continuous-time eigenvalues obtained for the five different data sets using this methodology. Because we used truncated SVDs

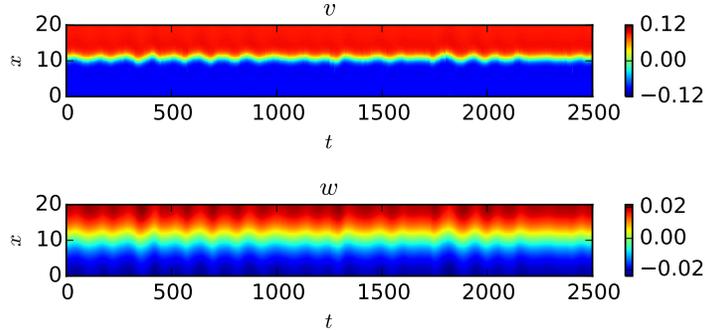


FIGURE 1. One of the five sets of data that were used to approximate the leading Koopman modes, eigenfunctions, and eigenvalues. (left) The numerically computed approximation of v sampled uniformly in time and space with $\Delta t = 1$. (right) The w data obtained at the same points in time. Although the initial condition for all five data sets is the equilibrium point, the random perturbations applied every 25 units of time result in ten different trajectories.

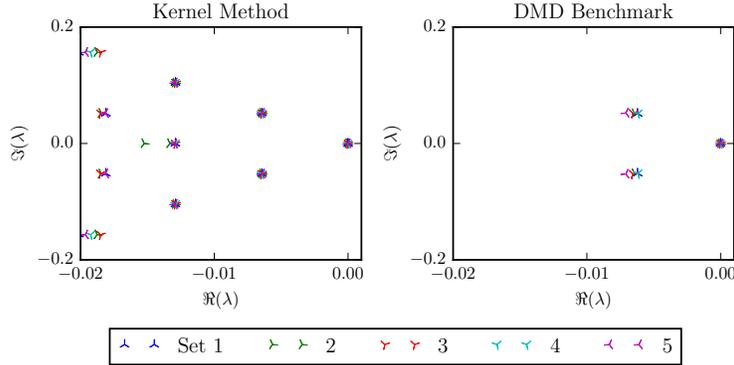


FIGURE 2. The leading eigenvalues computed using the kernel method and Dynamic Mode Decomposition are shown on the left and right respectively. These computations were performed on data sets like the one shown in Fig. 1 using only the 150 largest singular values to compute pseudo-inverses. Both DMD and the kernel approach consistently identify the $\lambda_1 = 0$ eigenvalue and the pair associated with the system linearization ($\lambda_{2,3} \approx -0.006 \pm 0.053i$). However, the kernel-based method appears to do so with less variance, and is also able to reconstruct two additional “layers” of eigenvalues, such as $\lambda_4 \approx -0.013$ and $\lambda_7 \approx -0.019 + 0.053i$, that are known to exist analytically [35].

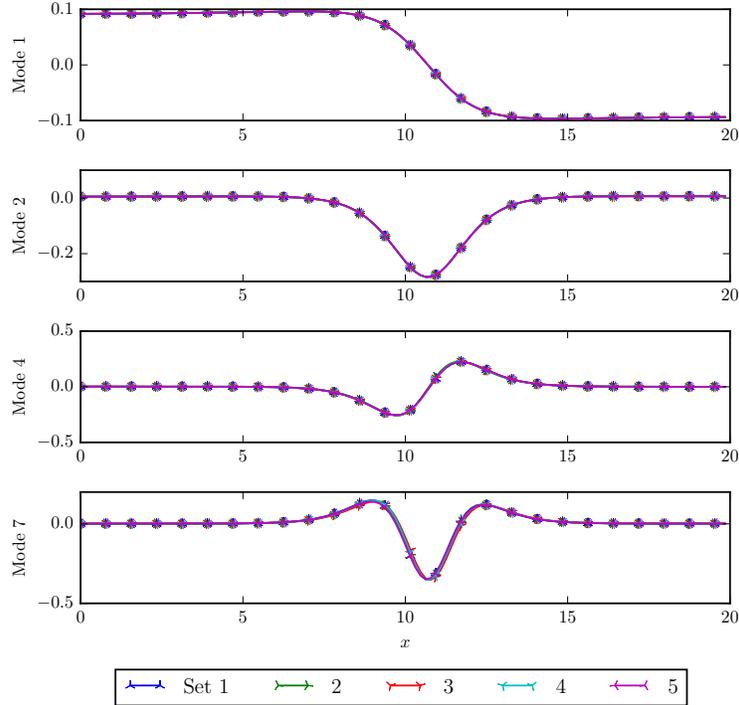


FIGURE 3. The real part of four of the Koopman modes for v computed using the kernel method associated with the eigenvalues near $\lambda_1 = 0$, $\lambda_2 = -0.006 + 0.053i$, $\lambda_4 = -0.013$, and $\lambda_7 = -0.019 + 0.053i$ respectively. Modes 1 and 2 are scaled versions of the equilibrium point and an eigenvector of the linearized system respectively. Mode 3, which is not shown, is the complex conjugate of mode 2, and is also computed accurately. There is a larger amount of variation in the 4th and 7th modes between the five data sets, but as shown above, the shape of these modes remain consistent.

for computing pseudoinverses, both DMD and the kernel method produce eigenvalues that are, up to numerical errors, contained in the left half plane. Furthermore, both DMD and the kernel method consistently identify the eigenvalues with $\lambda_1 \approx 0$ and $\lambda_{2,3} \approx -0.006 \pm 0.053i$ though, as shown in the figure, the variance in this pair is smaller when the kernel method is used. Although DMD identifies other eigenvalues with more negative real parts, no other eigenvalues lie in the subset of the complex plane shown in the figure. The kernel method, which uses a “richer” set of basis functions to approximate the Koopman eigenfunctions, also identifies two additional layers of the “pyramid” of eigenvalues that the Koopman operator should possess [36]. As a result, although the numerically computed eigenvalues are truly data dependent, the kernel approach: (i) appears to approximate the leading eigenvalues of the Koopman operator with less variance than the DMD benchmark,

and (ii) identifies additional Koopman eigenvalues the DMD benchmark appears to neglect.

The v -part of the modes corresponding to four of the leading eigenvalues are shown in Fig. 3. The first mode, which has $\lambda_1 \approx 0$, is a scaled version of the solution at the fixed point where the appropriate scaling factor is determined by the corresponding Koopman eigenfunction. The error, i.e., $\|\xi_1 - \xi_{\text{true}}\|$ with the normalization $\|\xi\| = 1$, was less than 0.01 for the five data sets used in the computation (the mean value of the error was 6.9×10^{-3}). The next pair of modes (modes 2 and 3), which have $\lambda_{2,3} \approx -0.006 \pm 0.053i$, can be shown to be the “slowest” pair of eigenvectors of the system linearization, and provide a useful description of the evolution near the fixed point. The error in these modes was less than 0.02 for the five data sets (with a mean value of 1.9×10^{-2}). As a result, not only does the kernel method consistently identify the leading Koopman modes in this problem, but it is accurate as well.

The first mode associated with *nonlinear effects*, which has the eigenvalue $\lambda_4 \approx -0.013$, is shown in the third image of Fig. 3. As shown in the figure, the kernel method consistently identifies this mode, though there are differences in the mode shape that are visible to the eye if one “zooms in” near $x = 10$ in the figure. In general, both DMD and the kernel method become less accurate the further “down” (i.e., into the left half-plane) one goes, as is demonstrated by the slight yet visible differences in the 7th mode with $\lambda_7 \approx -0.019 + 0.053i$ shown in the bottom panel of Fig. 3. Intuitively, this is because the corresponding eigenfunctions tend to become more complicated functions; in this example, the eigenfunctions corresponding to each subsequent layer of the pyramid are higher powers of the leading two eigenfunctions. As a result, they are less likely to lie in or near the subspace spanned by our implicitly chosen basis set, and therefore less accurately computed.

In this section, we applied the kernel approach to the FitzHugh-Nagumo PDE, and found that it consistently identified a subset of the Koopman modes and eigenvalues including those that are not associated with the system linearization. The first three modes, where the true solutions happen to be known, are identified accurately, with a relative error of less than 2%. Given a sufficiently large amount of data and the proper choice of kernel, the this approach can also identify additional “layers” in the pyramid of Koopman eigenvalues, but here the variance in the numerically obtained eigenvalues and modes is larger, which is likely due to the fact the eigenfunctions of interest to not lie entirely within the span of the basis functions used to represent them. As a result, at a certain point the method transitions from yielding quantitatively accurate modes and eigenvalues to a more qualitative description of the underlying dynamics.

4. Example: Experimental Data of Flow Over a Cylinder. In this example, we apply the kernel approach to experimentally obtained PIV data for flow over a cylinder taken at Reynolds number 413, used in previous work by Tu et al. [37] and Hemati et al. [20]. The experiment was conducted in a water channel designed to minimize three-dimensional effects, and the computed velocity field sampled at 20 Hz at a resolution of 135×80 pixels [37]. From these snapshots of the velocity field, we compute the vorticity via standard finite difference methods. In the original manuscripts, a total of 8000 snapshots of the velocity field were obtained, but here we use 3000 of them in our computation.

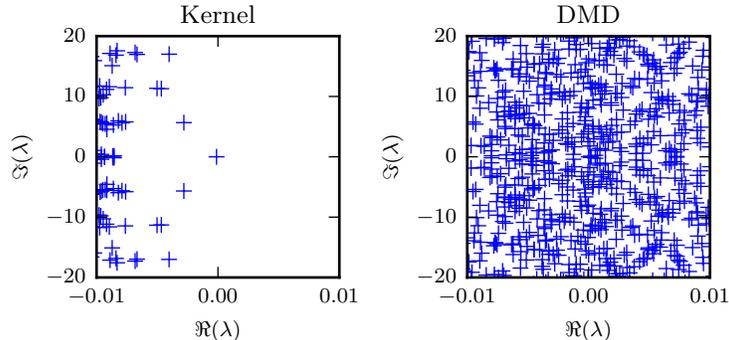


FIGURE 4. (left) A subset of the eigenvalues computed by applying to kernel method to a time-series of vorticity data computed from an experimentally obtained velocity field. (right) A subset of the eigenvalues that result from applying DMD to the same data set. Due to measurement and process noise in the data, the Koopman eigenvalues of this system are not known. However, the practical advantage of the kernel approach is that the *leading* eigenvalues are visually separated from the “cloud” of more rapidly decaying eigenvalues; while DMD also computes the eigenvalues associated with frequencies believed to exist in this system, it is not straightforward to separate these “useful” eigenvalues/modes from the remainder.

This example is intended to highlight how we foresee the kernel approach being used in practice. Due to the experimental nature of the data *and the presence of measurement and process noise*, the true Koopman eigenvalues and modes are not known. However as we will show shortly, the numerically computed Koopman modes and eigenvalues have more of the properties that the true eigenvalues should have, which implies the kernel method is producing a more accurate approximation than DMD. As a result, our objective here is to demonstrate the practical advantages of the kernel method, rather than the accuracy of the resulting modes and eigenvalues.

As before, we use a polynomial kernel with $\alpha = 20$, scale the data such that the mean value of $\|\mathbf{x}_i\| = 1$, and use (10) to approximate the Koopman eigenvalues and modes. From previous efforts including Tu et al. [37] and Hemati et al. [20], an effective set of modes and eigenvalues can be obtained from this data using DMD and similar methods. Figure 4 shows the leading eigenvalues obtained using the kernel method and, for reference, those computed using DMD. In both cases, all 2999 singular values were above machine epsilon, and were retained when computing pseudoinverses. The practical advantage of the kernel approach is that the eigenvalues and modes of interest can be identified visually; in particular, they are the “most slowly decaying” eigenvalues, which could be computed efficiently even in large systems using Krylov methods. On the other hand, DMD has a “cloud” of both stable and unstable eigenvalues, and the modes known to be most important are not necessarily those that decay the slowest. We should note that a “useful” set of eigenvalues are contained in this cloud, but energy [37] or sparsity-promoting [38] methods must be used in order to select this subset of the eigenvalues. To do this, however, both of these methods require all of eigenvalues and modes to be computed,

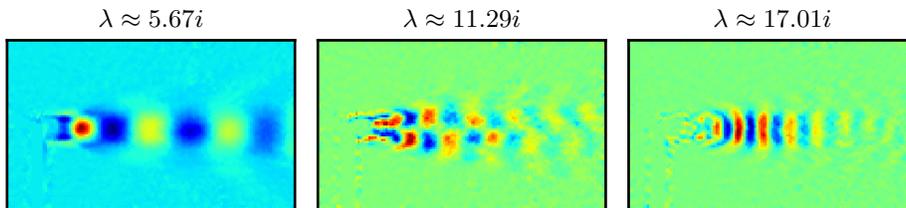


FIGURE 5. The real part of three of the leading Koopman modes and their associated eigenvalues obtained using the kernel method. These modes agree favorably with pre-existing modes computed using DMD [37], but these modes *were identified visually* rather than via energy-based or sparsity-promoting [38] methods for mode selection.

and then truncate this “complete” set, which can become computationally expensive in systems with larger numbers of snapshots and, therefore, modes to choose from.

The other eigenvalues computed by the kernel method decay more rapidly than those pictured in Fig. 4, and no eigenvalues have a positive real part (up to roundoff errors). Conceptually, the underlying system possesses a limit cycle with the addition of both process and measurement noise. In the absence of noise, the Koopman eigenvalues would lie on the imaginary axis and in regular intervals in the left half-plane [36]. The addition of noise causes many of the eigenvalues to decay (or decay more rapidly) [7, 23, 39], as the distribution of trajectories approaches the invariant distribution that is concentrated near the limit cycle. However, this decay rate depends upon the amount and type of noise added to the system, so while some decay is expected, it is unclear whether or not these eigenvalues are accurate.

The real part of three of the leading Koopman modes are shown in Fig. 5. Both the shape and angular frequency (i.e., $\Im(\lambda)$) of these modes are similar to those obtained using the full set of 8000 snapshots in Tu et al. [37] or Hemati et al. [20]. In particular, $\lambda = 5.67i$ corresponds to an oscillation frequency of 0.90 Hz, $\lambda = 11.29i$ to a frequency of 1.79 Hz, and $\lambda = 17.01i$ to 2.71 Hz. In Hemati et al. [20], these frequencies were reported as 0.89 Hz, 1.77 Hz, and 2.73 Hz respectively. The modes shown in Fig. 5 were obtained from the left eigenvectors of the matrix $\hat{\mathbf{K}}$ using (20), which in this example were obtained by completely decomposing $\hat{\mathbf{K}}$, but could also have been obtained by computing the leading left-eigenvectors using the implicitly restarted Arnoldi method [33], which would be more efficient if a larger number of snapshots were available.

In this section, we applied the kernel method to experimental data and computed the leading Koopman modes. Unlike the previous example, where the kernel approach produced modes that DMD could not, in this example both methods produce similar sets of modes. The difference, however, is that the “important” modes identified by Tu et al. [37] and Hemati et al. [20] using an energy-based method, are the clearly-visible leading (i.e., most slowly-decaying) modes of the kernel method. As a result, rather than computing *all* of the Koopman modes and then truncate, one could compute only the *leading* Koopman modes via (20). Furthermore, the eigenvalues identified by the kernel method possess more of the properties that the

Koopman eigenvalues should have; in particular, they lie on the imaginary axis or the left half-plane. Therefore, even though we cannot definitively say that the kernel method has accurately identified the Koopman eigenvalues, there are conceptual and practical advantages to using the kernel method in lieu of standard DMD even in experimental applications like the one shown here.

5. Conclusions. We have presented a data-driven method for approximating the Koopman operator in problems with large state spaces, which commonly occurs in physical applications such as fluid dynamics. The kernel method we have developed defines a subspace of scalar observables implicitly through a kernel function, which allows the method to approximate the Koopman operator with the same asymptotic cost as DMD, but with a far larger set of observables. In this manuscript, we used a polynomial kernel, whose associated set of basis functions can represent α -th order polynomials. However, many other kernels are available, associated with other basis sets; as a result, the choice of kernel, like the choice of a basis set in Extended DMD, is important but user determined.

To highlight the effectiveness of the kernel method, we applied it to a pair of illustrative examples. The first is the FitzHugh-Nagumo PDE in one spatial dimension, where a subset of the Koopman eigenvalues and modes could be deduced by linearizing the system about an equilibrium point. For this subset, the kernel method consistently and accurately identified the leading Koopman eigenvalues and modes. However, it also accurately computed additional Koopman eigenvalues that are not associated with the system linearization and, although we do not have an analytical expression for the corresponding modes, consistently identified a mode shape.

In practical applications, the data often come from systems whose true Koopman eigenvalues (or related information such as the location and linearization about fixed points) are unknown. Our second example, which used experimental data from flow over a cylinder, was a more realistic example of how we envision the kernel approach being applied. In that example, the numerically obtained eigenvalues had more of the properties the Koopman eigenvalues *should* have, but it was unclear whether or not they were quantitatively accurate because the distribution and nature of the noise in this example was unknown. The primary advantage of the kernel method here was that the Koopman modes of interest were associated with the eigenvalues that decay the slowest. As a result, they could be computed using the leading left-eigenvectors of the finite dimensional approximation, rather than requiring the complete set of eigenvectors and eigenvalue be computed and then truncated using energy-based or sparsity promoting methods.

In the end, the Koopman operator is an appealing mathematical framework for defining a set of spatial modes because these modes are intrinsic to the dynamics rather than associated with a set of data. However, obtaining effective approximations of this operator is non-trivial, particularly when all that is available are data. One method for obtaining such an approximation is Extended Dynamic Mode Decomposition [7], which is, for certain choices of basis functions, only computationally feasible in problems with small state spaces. The kernel method presented here is conceptually identical to Extended Dynamic Mode Decomposition, but computationally feasible even in large systems; in particular, the asymptotic cost of the kernel method is identical to that of “standard” Dynamic Mode Decomposition, and therefore, it can be used anywhere DMD is currently being used. Like most existing

data-driven methods, there is no guarantee that the kernel approach will produce accurate approximations of even the leading eigenvalues and modes, but it often appears to produce *useful* sets of modes in practice if the kernel and truncation level are chosen properly. As a result, approaches like the kernel method presented here are a first step towards the ultimate goal of enabling the conceptual tools provided by Koopman spectral analysis to be used in practical applications.

Acknowledgments. The authors acknowledge support from NSF (awards DMS-1204783 and CDSE-1310173) and AFOSR.

REFERENCES.

- [1] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, 2nd ed. Cambridge University Press, 2012.
- [2] A. Chatterjee, “An introduction to the proper orthogonal decomposition,” *Current Science*, vol. 78, no. 7, pp. 808–817, 2000.
- [3] P. Schmid, L. Li, M. Juniper, and O. Pust, “Applications of the dynamic mode decomposition,” *Theoretical and Computational Fluid Dynamics*, vol. 25, no. 1-4, pp. 249–259, 2011.
- [4] P. J. Schmid, D. Violato, and F. Scarano, “Decomposition of time-resolved tomographic PIV,” *Experiments in Fluids*, vol. 52, no. 6, pp. 1567–1579, 2012.
- [5] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, “Spectral analysis of nonlinear flows,” *Journal of Fluid Mechanics*, vol. 641, pp. 115–127, 2009.
- [6] M. Budišić, R. Mohr, and I. Mezić, “Applied Koopmanism,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, p. 047510, 2012.
- [7] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, “A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition,” *arXiv:1408.4408*, 2014.
- [8] A. Wynn, D. Pearson, B. Ganapathisubramani, and P. Goulart, “Optimal mode decomposition for unsteady flows,” *Journal of Fluid Mechanics*, vol. 733, pp. 473–503, 2013.
- [9] J.-N. Juang, *Applied system identification*. Prentice Hall, 1994.
- [10] K. K. Chen, J. H. Tu, and C. W. Rowley, “Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses,” *Journal of Nonlinear Science*, vol. 22, no. 6, pp. 887–915, 2012.
- [11] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, “On dynamic mode decomposition: theory and applications,” *arXiv preprint arXiv:1312.0041*, 2013.
- [12] G. Tissot, L. Cordier, N. Benard, and B. R. Noack, “Model reduction using dynamic mode decomposition,” *Comptes Rendus Mécanique*, vol. 342, no. 6, pp. 410–416, 2014.
- [13] R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.
- [14] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. Springer, 2007.
- [15] C. M. Bishop *et al.*, *Pattern Recognition and Machine Learning*. Springer, 2006, vol. 1.

- [16] B. O. Koopman, “Hamiltonian systems and transformation in Hilbert space,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 17, no. 5, p. 315, 1931.
- [17] B. O. Koopman and J. von Neumann, “Dynamical systems of continuous spectra,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 18, no. 3, p. 255, 1932.
- [18] I. Mezic, “Analysis of fluid flows via spectral properties of the Koopman operator,” *Annual Review of Fluid Mechanics*, vol. 45, pp. 357–378, 2013.
- [19] S. Bagheri, “Koopman-mode decomposition of the cylinder wake,” *Journal of Fluid Mechanics*, vol. 726, pp. 596–623, 2013.
- [20] M. S. Hemati, M. O. Williams, and C. W. Rowley, “Dynamic mode decomposition for large and streaming datasets,” *Physics of Fluids*, vol. 26, no. 11, p. 111701, 2014.
- [21] P. J. Schmid, “Dynamic mode decomposition of numerical and experimental data,” *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.
- [22] B. Scholkopf, “The kernel trick for distances,” *Advances in Neural Information Processing Systems*, pp. 301–307, 2001.
- [23] I. Mezić, “Spectral properties of dynamical systems, model reduction and decompositions,” *Nonlinear Dynamics*, vol. 41, no. 1-3, pp. 309–325, 2005.
- [24] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*. Courier Dover Publications, 2013.
- [25] L. N. Trefethen, *Spectral Methods in MATLAB*. SIAM, 2000, vol. 10.
- [26] L. Sirovich, “Turbulence and the dynamics of coherent structures. part i: Coherent structures,” *Quarterly of applied mathematics*, vol. 45, no. 3, pp. 561–571, 1987.
- [27] C. J. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [28] G. Baudat and F. Anouar, “Kernel-based methods and function approximation,” in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2. IEEE, 2001, pp. 1244–1249.
- [29] C. E. Rasmussen, “Gaussian processes for machine learning.” MIT Press, 2006.
- [30] A. Mauroy and I. Mezic, “A spectral operator-theoretic framework for global stability,” in *52nd IEEE Conference on Decision and Control*, Dec 2013, pp. 5234–5239.
- [31] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.
- [32] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, “A kernel view of the dimensionality reduction of manifolds,” in *Proceedings of the 21st International Conference on Machine Learning*. ACM, 2004, p. 47.
- [33] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users’ Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM, 1998, vol. 6.
- [34] C. E. Elmer and E. S. Van Vleck, “Spatially discrete FitzHugh–Nagumo equations,” *SIAM Journal on Applied Mathematics*, vol. 65, no. 4, pp. 1153–1174, 2005.
- [35] P. Gaspard, G. Nicolis, A. Provata, and S. Tasaki, “Spectral signature of the pitchfork bifurcation: Liouville equation approach,” *Physical Review E*, vol. 51, no. 1, p. 74, 1995.

- [36] P. Gaspard and S. Tasaki, “Liouvillian dynamics of the hopf bifurcation,” *Physical Review E*, vol. 64, no. 5, p. 056232, 2001.
- [37] J. H. Tu, C. W. Rowley, J. N. Kutz, and J. K. Shang, “Toward compressed dmd: spectral analysis of fluid flows using sub-nyquist-rate piv data,” *arXiv preprint arXiv:1401.7047*, 2014.
- [38] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, “Sparsity-promoting dynamic mode decomposition,” *arXiv preprint arXiv:1309.4165*, 2013.
- [39] S. Bagheri, “Effects of weak noise on oscillating flows: Linking quality factor, Floquet modes, and Koopman spectrum,” *Physics of Fluids*, vol. 26, no. 9, p. 094104, 2014.

E-mail address: mow2@princeton.edu

E-mail address: cwrowley@princeton.edu

E-mail address: yannis@princeton.edu