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

MATTHEW O. WILLIAMS
United Technologies Research Center
411 Silver Lane
East Hartford, CT 06118, 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 Michael Dellnitz)

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 uses a set of scalar observables (functions that map a state to a scalar value) that are defined implicitly by the feature map associated with a user-defined kernel function. This circumvents the computational issues that arise due to the number of functions required to span a “sufficiently rich” subspace of all possible scalar observables in such applications. We illustrate this method on two examples: the first is the FitzHugh-Nagumo PDE, a prototypical one-dimensional reaction-diffusion system, and the second is a set of vorticity data computed from experimentally obtained velocity data from flow past a cylinder at Reynolds number 413. In both examples, we use the output of Dynamic Mode Decomposition, which has a similar computational cost, as the benchmark for our approach.

1. Introduction. In many applications, the evolution of complex spatiotemporal 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 [15, 8, 29, 28, 26, 6, 35, 36, 17, 9, 34, 32, 10, 20]. Perhaps the best known method is the Proper Orthogonal Decomposition (POD) [15, 8, 4], which is related to Principal Component Analysis (PCA) [4, 20]. However, other algorithms that generate different sets of modes exist. In recent years, approximations of the modes of the Koopman operator [19, 18] have become popular in fluid flow applications [26, 6, 24, 1, 9, 36, 14]. The Koopman modes of the full state observable,

2010 Mathematics Subject Classification. Primary: 37M10, 65P99, 47B33.
Key words and phrases. Koopman operator, dynamic mode decomposition, kernel methods, time series analysis, machine learning.
which are vectors for systems of ODEs or spatial profiles for PDEs, are intrinsic to a particular evolution law, and have simple (i.e., exponential) temporal dynamics that are determined by their corresponding Koopman eigenvalues [26]. 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, the Koopman operator acts on scalar observables, which are functions defined on state space, and is therefore an infinite-dimensional operator even if the underlying system is finite-dimensional. As a result, methods that approximate the Koopman operator (often implicitly) select a particular 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) [27, 29, 28, 26, 6] (along with its related modifications [9, 36, 14]), which implicitly uses linear functions of the state as a basis for this subspace [35]. Although this is a useful choice of subspace in many circumstances, one can find systems where this subspace is simply not “rich” enough to effectively approximate the “dynamically relevant” eigenfunctions of the Koopman operator [35, 34].

Other methods, such as Extended DMD [35], 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, this “sufficiently rich” subspace can be captured by explicitly choosing a few hundred or thousand functions and allowing the algorithm to determine the subset needed to approximate the eigenfunctions. For example, one may choose to use all polynomials up to a certain order based on the knowledge that the eigenfunctions of interest are analytic where data are available. Provided the number of these functions is not too large, this is computationally tractable even on a laptop. However, applying this “brute force” procedure in systems with higher dimensional state spaces becomes difficult because the size of an effective basis set grows rapidly which makes the associated computations intractable in practice. This explosion in the computational cost is common in machine learning applications, and one facet of the “curse of dimensionality” [4].

In this manuscript, we introduce a data-driven, kernel-based method to approximate the Koopman operator in systems with large state dimension. The objective of this approach is to combine the “rich” set of observables associated with Extended DMD with the computational efficiency associated with DMD. To do this, we will circumvent the dimensionality issues encountered by Extended DMD by defining a kernel function that implicitly computes inner products in the high-dimensional space of observables. As we will show, choosing a kernel function is synonymous with choosing a set of observables, so the kernel-based approach can reproduce the same Koopman eigenvalues and eigenfunctions that Extended DMD would given the same data and the corresponding set of observables. The benefit of using a kernel rather than an explicit set of observables is that the computational cost of the approach is determined by the number of snapshots and the dimension of state space rather than the number of functions used to represent the scalar observables. As a result, the computational cost of this method is now comparable to that of DMD, which has successfully been applied to problems with high dimensional state spaces in the past.
We apply the kernel-based 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 at Reynolds number 413. Using the FitzHugh-Nagumo PDE example, we will demonstrate that using a higher-dimensional subspace of observables than DMD can enable Koopman-based algorithms to more accurately and consistently reproduce the leading Koopman modes, eigenvalues, and eigenfunctions. The cylinder example represents a typical use case for methods like DMD, and our objective here is to demonstrate that the kernel-based approach has practical advantages in more realistic settings where the data are noisy and the “true” Koopman eigenvalues and modes 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. Since this example is a more typical application area for DMD and similar methods, we compare the computational cost of our method, as measured in terms of wall-clock time, against a DMD benchmark. 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 [30]. In this section we will: (i) briefly review the Koopman operator, (ii) give a short derivation of “standard” Extended DMD, which enables the “tuples” of Koopman eigenvalues, eigenfunctions, and modes to be approximated from data, and (iii) present the kernel-based reformulation of the approach.

2.1. The Koopman operator. Although we will focus on data-driven approximations throughout this manuscript, the Koopman operator [19, 18], along with its eigenvalues, eigenfunctions, and modes, is defined by a dynamical system and not a set of data.

Definition 1 (Koopman operator). Given the discrete time dynamical system \((n, M, F)\), where \(n \in \mathbb{Z}\) is time, \(M \subseteq \mathbb{R}^N\) is the state space, and \(x \mapsto F(x)\) defines the dynamics, the action of the Koopman operator, \(K\), on a scalar observable, \(\phi : M \to \mathbb{C}\), is

\[
(K\phi)(x) = (\phi \circ F)(x) = \phi(F(x)).
\]

Intuitively, \(K\phi\) is a new scalar valued function that gives the value of \(\phi\) “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 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, \(g : M \to \mathbb{R}^{N_o}\), 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 \(g\) using a Koopman eigenfunction.
basis [26, 35]. One particularly useful set of modes is that of the full state observable, \( g(x) = x \), which we refer to as simply the Koopman modes in all that follows.

In many systems [26, 23, 24], triples consisting of an eigenvalue, an eigenfunction, and a mode enable a simple yet powerful means of representing the system state and predicting future state values. In particular,

\[
x = \sum_{k=1}^{N_k} \xi_k \varphi_k(x), \quad F(x) = \sum_{k=1}^{N_k} \mu_k \xi_k \varphi_k(x),
\]

where \( \xi_k \) is the Koopman mode corresponding to the eigenfunction \( \varphi_k \), \( \mu_k \) is the corresponding eigenvalue, and \( N_k \) is the number of tuples required for a practically useful reconstruction, which could be infinite. The eigenfunction \( \varphi_k : \mathcal{M} \to \mathbb{C} \) is a function, but the mode \( \xi_k \in \mathbb{C}^N \) is a vector. From (2), the coefficient associated with the \( k \)-th mode, \( \xi_k \), is obtained by evaluating the \( k \)-th eigenfunction, \( \varphi_k \), and the temporal evolution is dictated by \( \mu_k \). Ultimately, one benefit of this approach is that the Koopman eigenvalues, eigenfunctions, and modes are intrinsic to the dynamical system, so unlike the principal components or the outputs of most other data-driven dimensionality reduction methods, they are defined absent any data.

### 2.2. Extended dynamic mode decomposition

Extended Dynamic Mode Decomposition (Extended DMD) [35] is a regression procedure whose solution produces a finite-dimensional approximation of the Koopman operator. To do this, Extended DMD must be provided with a set of \( N_k \) scalar observables, which we denote as \( \psi_k : \mathcal{M} \to \mathbb{R} \) for \( k = 1, \ldots, N_k \), that span \( \mathcal{F}_{N_k} \subset \mathcal{F} \). In addition to defining a subspace of the scalar observables, the \( \psi_k \) also define a mapping from state space to feature space:

**Definition 2 (Feature space).** Suppose the space of scalar observables is approximated using \( N_k \) functions, then feature space is \( \mathbb{R}^{N_k} \). The feature map from state space to a subset of feature space is determined by

\[
\psi(x) = \begin{bmatrix}
\psi_1(x) \\
\psi_2(x) \\
\vdots \\
\psi_{N_k}(x)
\end{bmatrix},
\]

which is simply the set of scalar observables “stacked” into a column vector.

**Remark 1.** Due to the connection between state space and the definition of feature space, the value of any function \( \phi, \hat{\phi} \in \mathcal{F}_{N_k} \) at \( x \in \mathcal{M} \) can be written as the inner product of a coefficient vector \( a, \hat{a} \in \mathbb{C}^{N_k} \) and an element of feature space:

\[
\phi(x) = \sum_{k=1}^{N_k} a_k \psi_k(x) = \psi(x)^T a, \quad \hat{\phi}(x) = \sum_{k=1}^{N_k} \hat{a}_k \psi_k(x) = \psi(x)^T \hat{a}.
\]

Our goal is to capture the evolution of a scalar observable in the subspace \( \mathcal{F}_{N_k} \). In practice, \( F \) is not known explicitly, and all we have access to is a data set of snapshot pairs:

\[
\{(x_m, y_m)\}_{m=1}^M, \quad y_m = F(x_m),
\]

where \( x_m, y_m \in \mathcal{M} \). One important special case of such a data set is a single time-series of data, which can be expressed in the above form by “grouping” sequential pairs of snapshots that were obtained with a fixed sampling interval \( \Delta t \).
Then, let \( \hat{\phi} = K \phi + r \), where \( r \in \mathcal{F} \) is a residual function that appears because \( \mathcal{F}_{N_k} \) is not necessarily invariant to the action of the Koopman operator. Using the notation in (4), the objective of the Extended DMD procedure [35] is to use the data in (5) to identify a mapping \( K \in \mathbb{R}^{N_k \times N_k} \) from some given coefficient vector, \( \mathbf{a} \), to a new vector \( \hat{\mathbf{a}} \) such that this residual is minimized. One procedure that does this is:

**Algorithm 1** (Conceptual procedure for Extended DMD).

1. Using the data set of snapshot pairs in (5) and the set of scalar observables aggregated into \( \psi \), compute the observation matrices

\[
\Psi_x \triangleq \begin{bmatrix}
\psi(x_1)^T \\
\psi(x_2)^T \\
\vdots \\
\psi(x_M)^T
\end{bmatrix}, \quad \Psi_y \triangleq \begin{bmatrix}
\psi(y_1)^T \\
\psi(y_2)^T \\
\vdots \\
\psi(y_M)^T
\end{bmatrix},
\]

where \( \Psi_x, \Psi_y \in \mathbb{R}^{M \times N_k} \).

2. Then compute

\[
K \triangleq \Psi_x^+ \Psi_y,
\]

where \( + \) denotes the pseudoinverse obtained using the SVD.

**Remark 2.** While this procedure is extremely simple, it is only applicable to problems with moderate numbers of snapshots and scalar observables as both \( \Psi_x \) and \( \Psi_y \), which are typically dense matrices, must be stored in their entirety. As we will show, identical results can be obtained without explicitly forming \( \Psi_x \) and \( \Psi_y \).

In Ref. [35], the authors considered the case where \( M \gg N_k \), and proposed the following computational procedure:

**Algorithm 2** (Extended DMD).

1. Form the matrices of outer products

\[
G = \Psi_x^T \Psi_x = \sum_{m=1}^{M} \psi(x_m)\psi(x_m)^T,
\]

\[
A = \Psi_x^T \Psi_y = \sum_{m=1}^{M} \psi(x_m)\psi(y_m)^T,
\]

by evaluating \( \psi(x_m) \) and \( \psi(y_m) \), computing the outer products of the appropriate pair of vectors, and updating the entries of \( G \) and \( A \). By constructing \( G \) and \( A \) term by term, we avoid the need to explicitly form the \( \Psi_x \) and \( \Psi_y \) matrices.

2. Compute the pseudoinverse of \( G \), and let

\[
K \triangleq G^+ A.
\]

**Remark 3.** Because EDMD is a least-squares problem, the matrix \( K \) in (7) is equivalent to the definition in (8). Equation 7 identifies \( K \) by solving the least squares problem using the pseudoinverse while (8) uses the normal equations.

Regardless of the method, the properties of \( K \) of interest here are:
1. The $k$-th eigenvalue of $K$, $\mu_k$, is an approximation of an eigenvalue of $K$.
   When the data are generated by sampling a continuous time dynamical system with a fixed sampling interval $\Delta t$, 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, $v_k$, approximates an eigenfunction of the Koopman operator via
   \[ \varphi_k \triangleq \psi^T v_k. \] (9)

3. The left eigenvector, $w_k$, can be used to approximate the Koopman mode, $\xi_k$. For a derivation of these relationships, see Ref. [35].

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 [28, 29, 27, 26, 6, 24, 1, 9, 36, 14] where the number of snapshots is small compared to the dimension of our subspace of scalar observables (i.e., $M \ll N_k$).

Remark 4. The difficulty that appears when $N_k \gg M$ is that the Extended DMD procedure, as formulated in Sec. 2.2, requires $O(N_k^2 M)$ time to assemble $K$ and $O(N_k^3)$ time to compute the modes and eigenvalues. However, the value of $N_k$ for a “rich” set of basis functions grows rapidly as the dimension of state space increases, which makes this approach computationally infeasible.

Example 1. Consider the case where $\mathcal{F}_{N_k}$ is the space of all (multivariate) polynomials on $\mathbb{R}^{256}$ with degree up to 20. Then, $N_k \sim 10^{30}$, which is far too large for practical computations [4]. This explosion in the size of the problem is common in machine learning applications, and is one facet of the curse of dimensionality.

Similar to the approach taken by Schmid [27], we make the problem tractable by computing $\hat{K}$, a matrix similar to $K$ that allows its non-zero eigenvalues and their corresponding eigenvectors to be computed. Next, we define $\hat{K}$.

Proposition 1. Let the SVD of $\Psi_x$ be
   \[ \Psi_x \triangleq Q \Sigma Z^T, \] (10)
   where $Q, \Sigma \in \mathbb{R}^{M \times M}$ and $Z \in \mathbb{R}^{N_k \times M}$. The pair $\mu \neq 0$ and $\hat{\vartheta}$ are an eigenvalue and eigenvector of
   \[ \hat{K} \triangleq (\Sigma^+ Q^T)(\Psi_y \Psi_x^T) (Q \Sigma^+) = (\Sigma^+ Q^T) \hat{A} (Q \Sigma^+) \] (11)
   if and only if $\mu$ and $v = Z \hat{\vartheta}$ are an eigenvalue/eigenvector pair of $K$.

Proof. First, note that $Z = \Psi_x^T Q \Sigma^+$. Then, starting with
   \[ Kv = (\Psi_y^T \Psi_y) (Z \hat{\vartheta}) = Z [\Sigma^+ Q^T \Psi_y Z] \hat{\vartheta} \]
   \[ = Z ([\Sigma^+ Q^T](\Psi_y \Psi_x^T)(Q \Sigma^+)) \hat{\vartheta} \]
   \[ = Z \hat{K} \hat{\vartheta} = \mu Z \hat{\vartheta} = \mu v, \]
   so if $\hat{\vartheta}$ is an eigenvector of $\hat{K}$ then $v = Z \hat{\vartheta}$ is an eigenvector of $K$. Starting with
   \[ \hat{K} \hat{\vartheta} = ([\Sigma^+ Q^T](\Psi_y \Psi_x^T)(Q \Sigma^+)) (Z^T v) \]
   \[ = ([\Sigma^+ Q^T](\Psi_y))(\Psi_x^T)(\Sigma^+)^+ v \]
   \[ = Z^T [Z(\Sigma^+ Q^T) \Psi_y] v = Z^T [\Psi_x^T \Psi_y] v \]
   \[ = Z^T K v = \mu Z^T v = \mu \hat{\vartheta}. \]
Note that the identity \((\Psi_x^T \Psi_x)^+ v = v\) holds because \(\mu \neq 0\), so \(v\) lies in the image of \(\Psi_x^+\) and therefore the image of \(Z\). As a result, if \(v\) is an eigenvector of \(\hat{K}\), then \(\hat{v}\) is an eigenvector of \(\hat{K}\).

If we could form \(\hat{K}\), then we could compute the non-zero eigenvalues and the corresponding left and right eigenvectors of \(\hat{K}\), which would yield the desired approximations of the Koopman eigenvalues, eigenfunctions and modes. However, this has simply moved the computational bottleneck from the eigendecomposition of \(K\) to computing the SVD of \(\Psi_x\), which is an \(O(N_k M^2)\) task. While this is an improvement, due to the (possibly extremely large) size of \(N_k\), even linear dependence on the number of observables will make the resulting problem computationally intractable.

In order to develop a more efficient algorithm, we must first demonstrate that the scalar observables do not need to be defined explicitly. More concretely:

**Remark 5.** The matrix \(\hat{K}\) in (11) can be obtained from matrices whose elements are inner products in feature space evaluated at pairs of data points. To demonstrate this, we first define the matrix \(\hat{G} \equiv \Psi_y \Psi_x^T\), and reuse the matrix \(\hat{A} \equiv \Psi_y \Psi_y^T\) from (11). Despite appearing “flipped,” the \(i,j\)-th elements of \(\hat{G}\) and \(\hat{A}\) are

\[
\hat{G}^{(ij)} \equiv \psi(x_j)^T \psi(x_i), \quad \hat{A}^{(ij)} \equiv \psi(x_j)^T \psi(y_i).
\]

(12)

However \(\hat{G} = Q \Sigma^2 Q^T\), using the definition of \(Q\) and \(\Sigma\) in (10). Therefore, given \(\hat{G}\) we can obtain \(Q\) and \(\Sigma\) via its eigendecomposition. With \(\hat{A}, Q\), and \(\Sigma\) in hand, \(\hat{K}\) can be constructed using (11). Note that this is simply the method of snapshots approach for computing the POD modes performed in feature space [31].

With this property established, we can now use the kernel trick:

**Definition 3** (The kernel trick). The kernel trick is a common technique for efficiently computing inner products in an implicitly defined reproducing kernel Hilbert space [7, 3, 30, 25]. A kernel function has the form \(f : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}\), and is associated with a feature map \(\psi\), which maps from state space to feature space. The relationship between \(f\) and \(\psi\) is that \(f(x_i, x_j) = \psi(x_j)^T \psi(x_i)\) for all \(x_i, x_j \in \mathcal{M}\).

**Example 2** (A quadratic kernel). The simplest example is the polynomial kernel

\[
f(x, z) = (1 + z^T x)^2
\]

(13)

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

\[
f(x, z) = (1 + x_1 z_1 + x_2 z_2)^2
\]

\[
= (1 + 2x_1 z_1 + 2x_2 z_2 + 2x_1 x_2 z_1 z_2 + x_1^2 z_1^2 + x_2^2 z_2^2)
\]

\[
= \psi(z)^T \psi(x)
\]

if \(\psi(x) = [1, \sqrt{2} x_1, \sqrt{2} x_2, \sqrt{2} x_1 x_2, x_1^2, x_2^2]\). As a result, one could either compute inner products in feature space using the kernel function or the mapping, \(\psi\), identified above. The results are analogous, but the kernel approach requires an inner product of vectors in \(\mathbb{R}^2\) while the direct approach requires an inner product in \(\mathbb{R}^6\).

There are libraries of kernel functions, each of which is associated with a different implicitly defined feature map. In this paper, we use the following kernels:
Definition 4 (Polynomial kernel). Polynomial kernels have the form:

\[
f(x, z) = \left(1 + \frac{z^T x}{d^2}\right)^{\alpha},
\]

which are equivalent to a basis set that can represent all polynomials up to and including terms of degree \(\alpha\), but takes only \(O(N)\) time to evaluate. The addition of the \(d\) parameter scales the data by a constant factor, and affects the relative importance of higher order terms in the polynomial expansion.

In the example that follows in Sec. 3, 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 \([22]\), and polynomial kernels mimic an \(\alpha\) order power-series expansion. The specific choice of \(\alpha = 20\) is more ad hoc. In general, large values of \(\alpha\) use a “richer” set of basis functions, but also deleteriously impact the condition number of \(\hat{G}\). The choice of \(\alpha = 20\) appeared to strike a balance between having the “right” basis functions and avoiding numerical issues.

Definition 5 (Gaussian kernel). Gaussian kernels have the form:

\[
f(x, z) = \exp\left(-\frac{\|x - z\|^2}{\sigma^2}\right),
\]

which is commonly used in machine learning applications \([4, 11, 30]\), in order to highlight the performance of a different possible choice of the kernel. Similar to \(d, \sigma\) is a scaling parameter that is dependent on the characteristic length scales of the underlying problem.

This choice will result in different Koopman eigenvalues, eigenfunctions, and modes, and may yield superior (or inferior) results compared to the polynomial kernel. Ultimately, the choice of kernel is equivalent to the choice of dictionary elements in EDMD. The goal is to choose a kernel whose implicit reproducing kernel Hilbert space contains the Koopman eigenfunctions of interest, but the optimal choice is problem dependent and remains an open research question.

Once a kernel has been chosen, the procedure for obtaining an approximation of the Koopman operator is as follows:

Algorithm 3 (Kernel variant of Extended DMD).

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

\[
\hat{G}^{(ij)} \triangleq f(x_i, x_j), \quad \hat{A}^{(ij)} \triangleq f(y_i, x_j).
\]

2. Compute the eigendecomposition of the Gramian \(\hat{G}\) to obtain \(Q\) and \(\Sigma\).

3. Construct \(\hat{K}\) using (11).

Remark 6. The practical advantage of this formulation is that \(\hat{K}\) can be assembled in \(O(M^2N)\) time rather than \(O(M^2N_k)\) time. However, because \(\psi\) has not been defined explicitly, we must adjust the approach for Koopman mode and eigenfunction computation because the method in [35] assumes \(\psi\) is known.

Next we will show how to compute the values of the Koopman eigenfunctions and modes with the restriction that \(\psi\) has been defined implicitly through the choice of the kernel function. More concretely, we will demonstrate that the needed operations in feature space can be expressed entirely in terms of inner products, which will then be evaluated using the kernel function.
Remark 7. The values of the Koopman eigenfunctions can be evaluated at the supplied data points using the (right) eigenvectors of $\hat{K}$ and (9). Let $\hat{V}$ be the matrix whose columns are the eigenvectors of $\hat{K}$. Substituting this matrix into (9) results in the matrix of eigenfunctions values:

$$
\Phi_x \triangleq \Psi_x Z \hat{V} = (Q \Sigma Z^T) Z \hat{V} = Q \Sigma \hat{V},
$$

where $i$-th row of $\Phi_x$ contains the numerically computed eigenfunctions evaluated at $x_i$. Because $Q$ and $\Sigma$ are obtained as byproducts of the solving process, the Koopman eigenfunctions can be evaluated at the data points without explicit knowledge of $\psi$.

This formulation once again exploits the property that the eigenfunctions can be written as the inner product of two vectors in feature space, which is computed using the kernel selected. However because $\Phi_x$ evaluates the eigenfunctions at the data points, there is a significant amount of cancellation/pre-computation, which results in the simple expression shown above.

Remark 8. The $k$-th numerically approximated Koopman eigenfunction can also be evaluated at a new data point. Substituting $v_k = Z \hat{v}_k$ into (9) yields:

$$
\varphi_k(x) = \psi(x)^T (Z \hat{v}_k) = (\psi(x)^T \Psi_x^T) (Q \Sigma^+ \hat{v}_k)
= \left[ f(x, x_1) \quad f(x, x_2) \quad \cdots \quad f(x, x_M) \right] (Q \Sigma^+ \hat{v}_k),
$$

which uses the same reasoning as in (17) though without the simplifications and cancellations that occur in that case.

Next, we focus on the computation of the Koopman modes. In [35], it was assumed that the state itself could be written as a linear combination of the $\psi_k$ with known coefficients. Because we have not defined $\psi$ explicitly, this assumption will not hold in the kernel case. Instead, we select the modes to minimize the error associated with reconstructing the state data from the Koopman eigenfunction embedding, i.e.,

$$
\Xi = \arg \min_\Xi \frac{1}{2} \| X - \Phi_x \Xi \|_F^2.
$$

If

$$
X \triangleq \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_M^T \end{bmatrix}, \quad \Xi \triangleq \begin{bmatrix} \xi_1^T \\ \xi_2^T \\ \vdots \\ \xi_M^T \end{bmatrix} = \Phi_x^+ X = \hat{V}^{-1} \Sigma^+ Q^T X,
$$

provided that $\hat{V}$ has full rank. As a result, given the eigenvectors of $\hat{K}$ and the matrices used to construct it, we can also approximate the modes.

Remark 9. Provided $\hat{K}$ has a full set of eigenvectors, the inverse of $\hat{V}$ is

$$
\hat{V}^{-1} = \begin{bmatrix} \hat{w}_1^\top \\ \hat{w}_2^\top \\ \vdots \\ \hat{w}_M^\top \end{bmatrix},
$$

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

$$
\xi_i = (\hat{w}_i^\top \Sigma^+ Q^T X)^T.
$$
Therefore approximate Koopman modes and eigenfunctions can be obtained via
the left and right eigenvectors of \( \hat{K} \), and do not require a complete eigendecom-
position of \( \hat{K} \). For the problems considered here, a complete decomposition of \( \hat{K} \)
is performed, but for problems with larger numbers of snapshots, Krylov methods
could be used to compute a leading subset of its eigenvalues and vectors \([21]\). We
give a short example of the speedup possible using this approach in Sec. 4.

3. Example: The FitzHugh-Nagumo PDE. In order to highlight the effective-
ness of the kernel method, we first apply it to the FitzHugh-Nagumo PDE \([12]\) in
one spatial dimension using DMD as the benchmark. 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:

\[
\begin{align*}
\partial_t v &= \partial_{xx} v + v - w - v^3, \\
\partial_t w &= \delta \partial_{xx} w + \epsilon (v - c_1 w - c_0),
\end{align*}
\]

where \( v \) is the activator field, \( w \) is the inhibitor field, \( c_0 = -0.03, c_1 = 2.0, \delta = 4.0, \epsilon = 0.02, \) and \( x \in [0, 20] \) with Neumann boundary conditions. Both \( v \) and \( w \) are
approximated using a discrete cosine transform-based spectral method with 128
basis functions each \([5]\).

With these parameter values, \((23)\) 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

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

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 every 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 \( \alpha = 20 \). The scaling parameter \( d = \frac{1}{M} \sum_{m=1}^{M} ||x_m|| \). For
these data sets, both \( \mathbf{X} \) (for DMD) and \( \mathbf{\Psi}_x \) (for the kernel method) have small
singular values, so when we compute the pseudoinverse of \( \mathbf{X} \) (for DMD) or \( \mathbf{\Sigma} \) (for
the kernel method), we retain the 150 largest singular values and set the others to
zero. This is equivalent to projecting \( \mathbf{X} \) and \( \mathbf{Y} \) onto the 150 leading POD modes of
\( \mathbf{X} \) or \( \mathbf{\Psi}_x \) and \( \mathbf{\Psi}_y \) onto the leading 150 kernel principal components of \( \mathbf{\Psi}_x \) 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
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
Figure 1. One of the five sets of data that were used to approximate the leading Koopman modes, eigenfunctions, and eigenvalues. (top) The numerically computed approximation of $v$ sampled uniformly in time and space with $\Delta t = 1$. (bottom) 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 produce different trajectories.

Figure 2. The leading eigenvalues computed using the kernel method (left) and DMD (right). 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 $\lambda_7 \approx -0.019 + 0.053i$.

Pair is smaller when the kernel method is used. Although DMD identifies other eigenvalues with larger negative real parts, no other eigenvalues lie in the subset of the complex plane shown in the figure. The kernel method, which effectively 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 in this problem [13]. As a result, although the numerically computed eigenvalues are truly data dependent, the kernel approach: (i) appears to
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. There is a larger amount of variation in the 4th and 7th modes between the five data sets, but the shapes of these modes remain consistent.

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 real part of the $v$-component 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, $\|\xi - \xi_{\text{true}}\|$ 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 $6.9 \times 10^{-3}$). The next pair of modes (modes 2 and 3), which have $\lambda_{2,3} \approx -0.006 \pm 0.053i$, are 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 $1.9 \times 10^{-2}$). 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 are 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, this approach can also identify additional “layers” 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 do not lie entirely within the span of the functions used to represent them. As a result, by choosing a larger set of functions, the kernel method has the potential to accurately reproduce more of the leading Koopman eigenvalues, eigenfunctions, and modes than the DMD benchmark.

4. Example: Experimental data of flow over a cylinder. In this example, we apply the kernel approach to experimentally obtained Particle Image Velocimetry (PIV) data for flow over a cylinder taken at Reynolds number 413, which was used in previous works [33, 14]. 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 $135 \times 80$ pixels [33]. From these snapshots of the velocity field, we compute the vorticity via standard finite difference methods. In the original manuscripts, a single time-series consisting of 8000 snapshots of the vorticity were used. In this manuscript, we create multiple data sets by randomly selecting 3000 snapshot pairs from this master data set. As a result, although we have access to the same data, the data set is just a set of snapshot pairs and not a single time-series.

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 several of the properties that the true eigenvalues should have, which implies the kernel method is producing a more accurate approximation than DMD. Because we cannot quantitative judge the performance of the kernel method in this setting, our objective here is simply to demonstrate the practical advantages of the kernel method, rather than the accuracy of the resulting modes and eigenvalues.

We use a polynomial kernel with $\alpha = 5$, set $d = \frac{1}{M} \sum_{m=1}^{M} \| x_m \|$, and use (11) to approximate the Koopman eigenvalues and modes. To demonstrate the efficacy of
other kernels, we also use the same data to approximate the Koopman eigenvalues and modes using a Gaussian kernel with \( \sigma = d \). In this problem, both kernels will produce similar eigenvalues and modes, but in other applications or even simply with other scaling parameters they may give different results.

From previous efforts [33, 14], a useful set of modes and eigenvalues can be obtained from this data using DMD and similar methods. Figure 4 shows the eigenvalues computed using DMD along with the kernel approach with either a polynomial or Gaussian kernel. In all cases, all 3000 singular values were above machine epsilon, and were retained when computing pseudoinverses. The practical advantage of either the polynomial or Gaussian kernel 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 most slowly. We should note that a useful set of eigenvalues and their concomitant modes are contained in this cloud, but energy [33] or sparsity-promoting [16] methods must be used in order to select this subset of the eigenvalues. To do this, however, both of these methods require all of the eigenvalues and modes to be computed, which could become computationally expensive in systems with larger numbers of snapshots.

In addition to the eigenvalues shown in Fig. 4, both kernel methods generate a cloud of eigenvalues associated with more rapid decay rates (i.e., to the left of the window shown). It should be stressed, however, that none of the neglected eigenvalues have a positive real part. 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 [13]. The addition of process noise causes many of the eigenvalues to decay (or decay more rapidly) [23, 2, 35], as the distribution of trajectories approaches the invariant distribution that is concentrated near the

Figure 4. (left) A subset of the eigenvalues computed by applying DMD to 3000 snapshot pairs of vorticity data. Red points denote eigenvalues in the right half-plane, and blue points denote those in the left half-plane. (center/right) A subset of the eigenvalues computed by the kernel method with a 5th order polynomial kernel and Gaussian kernel respectively.
The real part of three of the leading Koopman modes obtained using the kernel method with a polynomial kernel (top) and a Gaussian kernel (bottom). These modes agree favorably with pre-existing modes computed using DMD [33], but were identified visually rather than via energy-based or sparsity-promoting [16] methods for mode selection. Note: the frequencies listed above are the angular frequencies, and differ from similar results in Refs. [33, 14] by a factor of $2\pi$.

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 Refs. [33, 14]. Using the polynomial kernel results as an example, $\lambda \approx 5.58i$ corresponds to an oscillation frequency of 0.89 Hz, $\lambda \approx 11.15i$ to a frequency of 1.77 Hz, and $\lambda \approx 16.73i$ to 2.66 Hz. In Ref. [14], 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 $\hat{K}$ using (22), which in this example were obtained by completely decomposing $\hat{K}$. However, for both the polynomial and Gaussian kernels, they could also have been obtained by computing the leading left-eigenvectors using the implicitly restarted Arnoldi method [21].

Although more complex methods for scattered-data interpolation exist, one benefit of the kernel-based approach is that it is similar to the “standard” DMD algorithm both in implementation and computational cost. Table 1 shows the “wall-clock” time averaged over 100 realizations associated with DMD, the kernel method with a polynomial kernel, and with a Gaussian kernel. For each iteration, 3000 snapshot pairs (of the 8000 snapshot time-series) were randomly selected without
Matrix Assembly Time (s) | Mode/Eigenvalue Comp. Time (s)
--- | ---
DMD | 13.58 | 60.57 (42.4)
Polynomial | 10.81 | 49.58 (2.68)
Gaussian | 9.84 | 51.28 (12.1)

Table 1. The wall-clock times associated with assembling the matrix $\hat{K}$ and computing the eigenvalues and modes averaged over 100 randomly generated data sets consisting of 3000 snapshot pairs. The values in parentheses are the means of the wall-clock times required to compute the 5 largest eigenvalue/eigenvector pairs using ARPACK [21]. In all three cases, we do not compute the associated Koopman eigenfunction values; those would require the right-eigenvectors and would roughly double the “Mode/Eigenvalue Computation Time” values listed.

Figure 6. A superposition of the eigenvalues that result when DMD (left), a polynomial kernel (center), and a Gaussian kernel (right) are applied to the 100 data sets used to create Table 1. The kernel methods fairly consistently identify the same set of leading eigenvalues, though there is a significant amount of variance in the rapidly decaying eigenvalues. As in Fig. 4, eigenvalues in the left half plane are colored blue, and those in the right half plane are colored red.
there is the potential for significant computational savings if only a few of the leading eigenvalues/modes are required. The values in parentheses in Table 1 are the mean wall-clock times required to compute the five leading eigenvalues and vectors using ARPACK, which can be up to a factor of 10 faster than computing all of the eigenvalues/modes. For the sake of comparison, we also computed the five leading eigenvalues/modes for DMD, but it should be noted that this does not always produce modes of interest, which are the ones shown in Fig. 5, as they are not always associated with the most slowly decaying values of $\lambda$ in this example.

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 in Refs. [33, 14] 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 truncating, one could compute only the leading Koopman modes via (22). 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 in 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 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 $\alpha$-th order polynomials, and a Gaussian kernel. However, many other kernels are available and are 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 currently must be chosen by the user.

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, though it is likely they
overestimate the decay rates associated with those eigenvalues due to the randomly
selected data and the presence of measurement noise. 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 eigenvalues to 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 approxi-
mations 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 [35], which is, for certain choices of basis functions, only computa-
tionally feasible in problems with small state dimension. The kernel method presented
here is conceptually identical to Extended Dynamic Mode Decomposition, but computa-
tionally 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
of the pseudoinverse 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] S. Bagheri, Koopman-mode decomposition of the cylinder wake, Journal of Fluid Mechanics,
726 (2013), 596–623.
[2] S. Bagheri, Effects of weak noise on oscillating flows: Linking quality factor, Floquet modes,
and Koopman spectrum, Physics of Fluids, 26 (2014), 094104.
[3] G. Baudat and F. Anouar, Kernel-based methods and function approximation, In Proceed-
ings of the International Joint Conference on Neural Networks, IEEE, 2 (2001), 1244–1249.
[4] C. M. Bishop et al, Pattern Recognition and Machine Learning, Springer, 2006.
[5] J. P. Boyd, Chebyshev and Fourier Spectral Methods, Courier Dover Publications, Mineola,
NY, 2001.
[6] M. Budišić, R. Mohr and I. Mezić, Applied Koopmanism, Chaos: An Interdisciplinary
Journal of Nonlinear Science, 22 (2012), 047510, 33pp.
[7] C. J. Burges, A tutorial on support vector machines for pattern recognition, Data Mining
and Knowledge Discovery, 2 (1998), 121–167.
[8] A. Chatterjee, An introduction to the proper orthogonal decomposition, Current Science, 78
(2000), 808–817.
[9] K. K. Chen, J. H. Tu and C. W. Rowley, Variants of dynamic mode decomposition: Boundary
condition, Koopman, and Fourier analyses, Journal of Nonlinear Science, 22 (2012), 887–915.
[10] R. R. Coifman and S. Lafon, Diffusion maps, Applied and Computational Harmonic
Analysis, 21 (2006), 5–30.
[11] N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other
Kernel-Based Learning Methods, Cambridge University Press, 2000.
[12] C. E. Elmer and E. S. Van Vleck, Spatially discrete FitzHugh–Nagumo equations, SIAM
Journal on Applied Mathematics, 65 (2005), 1153–1174.
[13] P. Gaspard and S. Tasaki, Liouvillian dynamics of the Hopf bifurcation, Physical Review E, 64 (2001), 056232, 17pp.
[14] M. S. Hemati, M. O. Williams and C. W. Rowley, Dynamic mode decomposition for large and streaming datasets, Physics of Fluids, 26 (2014), 111701.
[15] P. Holmes, J. L. Lumley, G. Berkooz and C. W. Rowley, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, 2nd edition, 2012.
[16] M. R. Jovanović, P. J. Schmid and J. W. Nichols, Sparsity-promoting dynamic mode decomposition, Physics of Fluids, 26 (2014), 024103.
[17] J.-N. Juang, Applied System Identification, Prentice Hall, 1994.
[18] 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, 18 (1932), 255–163.
[19] B. O. Koopman, Hamiltonian systems and transformation in Hilbert space, Proceedings of the National Academy of Sciences of the United States of America, 17 (1931), 315–318.
[20] J. A. Lee and M. Verleysen, Nonlinear Dimensionality Reduction, Springer, 2007.
[21] R. B. Lehoucq, D. C. Sorensen and C. Yang, ARPACK Users’ Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, volume 6. SIAM, 1998.
[22] A. Mauroy and I. Mezić, A spectral operator-theoretic framework for global stability, In 52nd IEEE Conference on Decision and Control, (2013), 5234–5239.
[23] I. Mezić, Spectral properties of dynamical systems, model reduction and decompositions, Nonlinear Dynamics, 41 (2005), 309–325.
[24] I. Mezić, Analysis of fluid flows via spectral properties of the Koopman operator, Annual Review of Fluid Mechanics, 45 (2013), 357–378.
[25] C. E. Rasmussen, Gaussian Processes for Machine Learning, MIT Press, 2006.
[26] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter and D. S. Henningson, Spectral analysis of nonlinear flows, Journal of Fluid Mechanics, 641 (2009), 115–127.
[27] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, Journal of Fluid Mechanics, 656 (2010), 5–28.
[28] P. J. Schmid, D. Violato and F. Scarano, Decomposition of time-resolved tomographic PIV, Experiments in Fluids, 52 (2012), 1567–1579.
[29] P. J. Schmid, L. Li, M. P. Juniper and O. Pust, Applications of the dynamic mode decomposition, Theoretical and Computational Fluid Dynamics, 25 (2011), 249–259.
[30] B. Scholkopf, The kernel trick for distances, Advances in Neural Information Processing Systems, (2001), 301–307.
[31] L. Sirovich, Turbulence and the dynamics of coherent structures. part I: Coherent structures, Quarterly of applied mathematics, 45 (1987), 561–571.
[32] G. Tissot, L. Cordier, N. Benard and B. R. Noack, Model reduction using dynamic mode decomposition, Comptes Rendus Mécanique, 342 (2014), 410–416.
[33] 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, Experiments in Fluids, 55 (2014), 1–13.
[34] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton and J. N. Kutz, On dynamic mode decomposition: Theory and applications, Journal of Computational Dynamics, 1 (2014), 391–421.
[35] M. O. Williams, I. G. Kevrekidis and C. W. Rowley, A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition, Journal of Nonlinear Science, 25 (2015), 1307–1346.
[36] A. Wynn, D. S. Pearson, B. Ganapathisubramani and P. J. Goulart, Optimal mode decomposition for unsteady flows, Journal of Fluid Mechanics, 733 (2013), 473–503.

Received February 2015; revised February 2016.

E-mail address: william1@utrc.utc.com
E-mail address: cwrowley@princeton.edu
E-mail address: yannis@princeton.edu