On the Koopman operator of algorithms*

Felix Dietrich†, Thomas N. Thiem‡, and Ioannis Kevrekidis§

Abstract. A systematic mathematical framework for the study of numerical algorithms would allow comparisons, facilitate conjugacy arguments, as well as enable the discovery of improved, accelerated, data-driven algorithms. Over the course of the last century, the Koopman operator has provided a mathematical framework for the study of dynamical systems, which facilitates conjugacy arguments and can provide efficient reduced descriptions. More recently, numerical approximations of the operator have enabled the analysis of a large number of deterministic and stochastic dynamical systems in a completely data-driven, essentially equation-free pipeline. Discrete or continuous time numerical algorithms (integrators, nonlinear equation solvers, optimization algorithms) are themselves dynamical systems. In this paper, we use this insight to leverage the Koopman operator framework in the data-driven study of such algorithms and discuss benefits for analysis and acceleration of numerical computation. For algorithms acting on high-dimensional spaces by quickly contracting them towards low-dimensional manifolds, we demonstrate how basis functions adapted to the data help to construct efficient reduced representations of the operator. Our illustrative examples include the gradient descent and Nesterov optimization algorithms, as well as the Newton-Raphson algorithm.

Key words. Koopman operator, algorithm, gradient descent, Nesterov, Newton-Raphson

AMS subject classifications. 47B33, 68W40, 37C10

1. Introduction.

The relation between algorithms and dynamical systems has been studied for a long time. Due to the breadth of the topic, we cannot give an exhaustive review of the literature and will only briefly discuss some of the research areas important to this work. The book by Stuart and Humphries [42] and the article by Chu [8] provide a broad overview of the relation between algorithms and dynamical systems. Much recent research focuses on the discrete steps in the solution to linear systems and the relation to continuous dynamics on manifolds of matrices [8, 16, 25, 33]. In particular, algorithms for eigenvalue problems have been studied in this context, with the relation to integrable systems (Lax-pair formulation) [44]. Studying the connection between algorithms and dynamical systems led to many surprising results, such as the definition of systems that can "sort numbers" [4], or, more generally, solve optimization problems with constraints [15]. Chaotic dynamical systems that "paint", i.e., have stationary distributions according to the distribution of paint in an image, have been studied in relation to Markov Chain Monte Carlo algorithms [39]. Another connection relates particle dynamics to algorithmic solutions [11]. In the neural network research community, the idea of a (deep) neural network representing an iterated map of a

*Submitted to the editors on May 20, 2020.

Funding: This work was partially funded by the DARPA/Lagrange program (Drs. Fahroo and Lewis) and by the ARO through a MURI (Drs. S. Stanton and M. Munson).

†Department of Chemical and Biomolecular Engineering and Department of Applied Mathematics and Statistics, Johns Hopkins University and Department of Informatics, Technical University of Munich (felix.dietrich@tum.de).

‡Department of Chemical and Biomolecular Engineering, Princeton University (tthiem@princeton.edu).

§Department of Chemical and Biomolecular Engineering and Department of Applied Mathematics and Statistics, Johns Hopkins University and JHMI (yannis@princeton.edu).

discrete dynamical system has gained attention, particularly with respect to the continuous generator of this discrete map [36]. In general, optimizing the same loss function through different algorithms such as gradient descent or "stochastic" gradient descent may result in different trajectories, and different local or even global minima may be found. Optimization algorithms modeling second-order dynamical systems (for example, Nesterov's method, sometimes called the "heavy ball with friction" algorithm) are favoured over traditional gradient descent methods, because of their ability to overcome local minima [2, 35].

The work of Koopman and von Neumann on ergodic, mixing, and chaotic dynamical systems [20, 45] revealed that that there is a canonical, linear operator associated to each system. The operator acts on complex-valued functions of the system state, the result being evaluation of the function at a future time. Linearity of the operator makes it amenable to finite-dimensional matrix approximations on computers, which was heavily exploited in the last twenty years [29, 7]. The Koopman operator and its adjoint, the Frobenius-Perron operator, are also known to provide optimal basis functions for uncertainty quantification [1, 14].

In this paper, we discuss benefits of the operator viewpoint on algorithms, when considering them as dynamical systems. The main problem we are addressing with the new operator viewpoint is the data-driven analysis of complex algorithms acting on high-dimensional state spaces. The Koopman operator and related numerical approximation methods for it allow us to treat this challenge in a unified way, solving many pressing issues such as acceleration and formulation of data-driven surrogate models, discovery of (almost) invariant sets and regions of convergence, and high-dimensional state spaces through model order reduction. We demonstrate that a numerical approximation of the operator may be possible, even for systems with partially available or high-dimensional data. We also show how properties of the Koopman operator, such as ergodic quotients and spectral analysis, offer valuable insight into algorithmic behavior. Even if the only available data is a set of randomly distributed, single-iteration pairs of states, the operator (and thus a sense of the algorithm) can be approximately constructed. We demonstrate a particularly novel approach to algorithm analysis through the computation of the spectrum of the Koopman operator. Summarizing a complex, nonlinear algorithm in such a meaningful way–and in the language of linear algebra–may provide a missing link to more fundamental results. An example is the comparison of algorithms based on their Koopman spectrum, which may lead to a definition of metrics or distances between algorithms. Conjugacy arguments have also been used to analyze dynamical systems for a long time. The operator viewpoint on algorithms can bring this successful approach to algorithm analysis, comparison, and development, even in a data-driven setting [3].

The remainder of the paper is organized as follows. In section 2.1, we introduce the relation between algorithms and dynamical systems. In section 2.2, we describe the Koopman operator framework, followed by section 3 with the Extended Dynamic Mode Decomposition algorithm for numerical approximation of the operator. Acceleration and domain decomposition of algorithms is outlined in section 4.1. In section 4.2, we describe how to construct a continuous version of the iterative Nesterov algorithm in a data-driven way. Section 4.3 describes numerical approximations of the Koopman operator for algorithms that act on high-dimensional spaces, but move their state close to a low-dimensional manifold after a few iterations. We analyze the Newton-Raphson method for root finding of polynomials on the complex plane in

section 4.4, with explicit construction of the spectrum and eigenfunctions, as well as a numerical analysis of chaotic behavior of the method on the real line. Appendix A contains some preliminary computational experiments on the impact (on the numerical construction of the Koopman operator) of only partial information about the state space (due to finite sample size). We conclude with a summary of the results and an outlook on data-driven algorithm analysis in section 5.

2. The Koopman operator of algorithms.

The focus of this paper is on potential benefits of the data-driven study of algorithms as dynamical systems from the Koopman operator viewpoint. In this section, we describe the necessary mathematical framework for dynamical systems, state the algorithms that we will use as examples, and briefly introduce the Koopman operator.

2.1. Algorithms are dynamical systems.

In this paper, we consider algorithms that act on their state either continuously or in discrete steps (iterations). We assume that the algorithms' state spaces $X \subseteq \mathbb{R}^d$ are smooth, k-dimensional Riemannian submanifolds embedded in Euclidean space of dimension $d \in \mathbb{N}$, $d \geq k$. The Riemannian metric on X is induced by the embedding. We define iterative algorithms as differentiable maps $a : X \to X$, with iteration number $n \in \mathbb{N}$. For a single iteration of the map a on a state x_n, we write

$$x_{n+1} = a(x_n), \ n \in \mathbb{N}.$$

For some algorithms, it is useful to consider continuously evolving state variables. In this case, the algorithm is typically represented as a differentiable vector field $v : X \to \mathbb{R}^k$. Starting from a given initial state $x_0 \in X$, this vector field generates a curve $c : \mathbb{R}^+ \to X$ in the state space through

$$\frac{d}{ds}c(s) = v(c(s)), \ c(0) = x_0. \tag{2.1}$$

Equation (2.1) describes the action of a continuous algorithm. It can be reformulated as a discrete map through the definition of a flow: The map $S : \mathbb{R}^+ \times X \to X$ is called a (semi-)flow of the vector field v if, for all $t, s \in \mathbb{R}^+$ and all $x \in X$,

$$S(t+s, x) = S(t, S(s, x)), \ \left.\frac{d}{dt}S(t, x)\right|_{t=0} = v(x).$$

If we fix a time interval $\Delta t \in \mathbb{R}^+$, the map $S(\Delta t, \cdot) =: S_{\Delta t}(\cdot) : X \to X$ is an iterative algorithm (a discrete-time dynamical system). This construction is always possible: however, it is important to note that in general it is not possible to find a continuous algorithm v for a given iterative algorithm a. We will discuss the necessary properties of a for a continuous version to exist, and show how it can be approximated in a data-driven procedure through the consideration of the Koopman operator. In this paper, we study the prototypical forms of algorithms listed in Table 1, formulated in continuous and discrete time.

2.2. Introduction to the Koopman operator.

We start with a definition of the family of Koopman operators in the deterministic setting in section 2.2.1. Details about the operator in the stochastic setting, and the relation with its adjoint, the Frobenius–Perron operator, are discussed in section 2.2.2.

Table 1

Algorithms formulated as dynamical systems, with their continuous (left) and discrete versions (right). Starting from a point $x_0 \in \mathbb{R}^n$, the function $f : \mathbb{R}^n \to \mathbb{R}$ (or its absolute value, in case of root-finding algorithms) is to be minimized, where ∇f denotes its gradient, and $H_x f$ the Hessian matrix at x. If f is vector-valued, $J_x f$ denotes its Jacobian matrix at x. The symbol $h \in \mathbb{R}^+$ denotes a small, positive constant.

Gradient Decent	
$\dot{x} = -\nabla f(x)$	$x_{n+1} = x_n - h\nabla f(x_n)$
Davidenko / Newton-Raphson method for optimization	
$(H_x f)\dot{x} + \nabla f(x_0) = 0$	$x_{n+1} = x_n - (H_{x_n} f)^{-1}\nabla f(x_n)$
Davidenko / Newton-Raphson method for root finding	
$(J_x f)\dot{x} + f(x_0) = 0$	$x_{n+1} = x_n - (J_{x_n} f)^{-1} f(x_n)$
Nesterov method	
$\ddot{x} + \frac{r}{t}\dot{x} + \nabla f(x) = 0, r \geq 3$	$x_{n+1} = y_n - h\nabla f(y_n)$ $y_{n+1} = x_{n+1} + \frac{n}{n+3}(x_{n+1} - x_n)$

2.2.1. Deterministic setting. Given a deterministic dynamical system with vector field $v : X \to \mathbb{R}^k$, flow $S_t : X \to X$, initial condition $x_0 \in X$, and

$$\frac{d}{dt} S_t(x)\Big|_{t=0} = v(x), \;\; S_0(x) = x_0, \tag{2.2}$$

the family of Koopman operators $\mathcal{K}^t$ indexed by $t \in \mathbb{R}$ is a family of linear operators acting on the function space $\mathcal{F}$ of observables $g : X \to \mathbb{C}$ such that

$$[\mathcal{K}^t g](x) := (g \circ S_t)(x).$$

The choice of function space $\mathcal{F}$ crucially determines the properties of $\mathcal{K}^t$. A typical choice is the space of complex-valued functions on the domain X that are square-integrable with respect to a measure μ, denoted

$$\mathcal{F} = L^2(X, \mathbb{C}, \mu) := \{g : X \to \mathbb{C}, \text{s.t.} \int_X |g(x)|^2 d\mu(x) < \infty\}. \tag{2.3}$$

The space $L^2(X, \mathbb{C}, \mu)$ is typically equipped with an inner product

$$\langle g_1, g_2 \rangle := \int_X g_1(x)\overline{g_2(x)}d\mu(x),$$

where the bar over $g_2(x)$ denotes complex conjugation. The set of all $\mathcal{K}^t$ forms a continuous group w.r.t the strong operator topology. It is a semi-group if S_t is not invertible for all t (e.g., if the system approaches spatial infinity in finite time). The strong operator topology is defined through pointwise convergence on $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$ [13]. The map $t \mapsto \mathcal{K}^t$ is continuous in the strong operator topology on a compact subset $B \subset \mathbb{R}$ if, for $t \in B$,

$$t \mapsto \mathcal{K}^t f$$

is continuous for every $f \in \mathcal{F}$. Continuous (semi-)groups are then defined as follows. The family $\{\mathcal{K}^t\}_{t\in\mathbb{R}}$ is a continuous (semi-)group of operators if, in the strong operator topology,

$$\lim_{t\to 0} \|\mathcal{K}^t - I\| = 0.$$

Typically, a fixed $t = t_0$ is chosen, and the associated Koopman operator $\mathcal{K}^{t_0}$ is studied. In the remaining text, if the specific choice of time is not important in the context, we will choose $t = 1$ and drop the superscript t to simplify notation. An important feature of the operator is the spectrum $\sigma(\mathcal{K}) \subset \mathbb{C}$, where we consider the atomic (or pure-point) part σ_{pp} and the absolutely continuous part σ_{ac} (see [7, 31] for discussions of the singularly continuous part). We can use the spectral theory for linear operators [34, 29] to write

$$\mathcal{K} = \sum_k \underbrace{\lambda_k}_{\in\sigma_{pp}} P_{\lambda_k} + \int_{\sigma_{ac}} \lambda \mathrm{d}E(\lambda), \tag{2.4}$$

where P_λ and $E(\lambda)$ are projection operators mapping functions to their projection in the eigenspaces associated to λ. Equation (2.4) decomposes the operator $\mathcal{K}^t$ at $t = 1$. Following functional calculus, the spectrum associated to other values of t can be obtained by exponentiation of λ, i.e. if $\lambda \in \sigma(\mathcal{K})$ then $\lambda^t \in \sigma(\mathcal{K}^t)$. For an eigenvalue $\lambda \in \sigma_{pp}(\mathcal{K})$, there are functions $\phi_\lambda \in \mathcal{F}$ such that

$$[\mathcal{K}^t\phi_\lambda](x) = (\phi_\lambda \circ S_t)(x) = \lambda^t\phi_\lambda(x),\ t \in \mathbb{R}^+.$$

These functions are called *eigenfunctions* associated to the eigenvalue λ. To predict the time evolution of a function $g \in \mathrm{span}\{\phi_{\lambda,k}\} \subset \mathcal{F}$ such that $g = \sum_k c_k\phi_{\lambda,k},\ c_k \in \mathbb{C}$, we can thus write

$$\mathcal{K}^t g = \mathcal{K}^t \sum_k c_k\phi_{\lambda,k} = \sum_k c_k\mathcal{K}^t\phi_{\lambda,k} = \sum_k c_k\lambda^t\phi_{\lambda,k}. \tag{2.5}$$

The coefficients c_k are sometimes called "Koopman modes" for the observable g, since they are used to reconstruct it. For many dynamical systems, the span of the eigenfunctions is a rich subspace of $\mathcal{F}$, and many observables can be defined through their Koopman modes.

For continuous-time dynamical systems, another important object related to the Koopman operator group $\mathcal{K}^t$ is the infinitesimal generator $A_\mathcal{K}$, where

$$A_\mathcal{K} g := \lim_{t \to 0} \frac{\mathcal{K}^t g - g}{t} \tag{2.6}$$

for all $g \in D(A_\mathcal{K}) \subset \mathcal{F}$. For continuous semi-groups, the domain $D(A_\mathcal{K})$ is dense in $\mathcal{F}$ [13]. The infinitesimal generator has the same eigenfunctions as any $\mathcal{K}^t$. For fixed $t > 0$ and eigenvalues λ^t of $\mathcal{K}^t$, the eigenvalues ω of $A_\mathcal{K}$ satisfy $\exp(t\omega) = \lambda^t$. The generator is important because it induces the following relation on the eigenfunctions ϕ_λ:

$$\langle \nabla \phi_\lambda(x), v(x) \rangle_{\mathbb{R}^n} = \omega \phi_\lambda(x) \tag{2.7}$$

for all $x \in \mathbb{R}^n$ where the eigenfunctions and the vector field are defined (see [3] for a discussion). Note that equations (2.6) and (2.7) reveal the relation between the generator $A_\mathcal{K}$ and the vector field v. If we interpret v as a map from a point $x \in X$ to the coefficients v_i for basis vectors ∂_i of the tangent space $T_x X$, then we can write

$$v(x) = \sum_i^d v_i(x) \partial_i.$$

With this reformulation, the vector field is an object that acts on functions $g : X \to \mathbb{R}$ from the left, i.e. $vg = \sum_i^d v_i \partial_i g$, which is exactly what equation (2.6) describes. Also, formally,

$$\mathcal{K}^t = \exp\left(t A_\mathcal{K}\right).$$

In this sense, the vector field v acts as the generator of the Koopman operator (semi)group. The generator $A_\mathcal{K}$ can be used to obtain an approximation of the vector field v from only discrete-time samplings of the dynamical system. This was used to identify vector fields from data of dynamical systems [26], and we will use it to identify vector fields for discrete algorithms in section 4.2.

The identification of vector fields using the Koopman operator is a global approach, taking into account all the data to obtain the vector field at every point. A different, more localized idea of identifying continuous vector fields on discrete data (e.g., [38, 37, 17]) is becoming increasingly popular in the machine learning literature.

2.2.2. Stochastic setting. We provide a brief introduction to the operator definitions in the stochastic setting. The manuscript mostly focuses on deterministic algorithms, and thus we refer to [41, 9, 19] for more theoretical details and applications of the stochastic case. Note that some numerical algorithms used to approximate the Koopman operator in the deterministic setting can be analyzed using a stochastic approximation of the system that accounts for numerical errors [46, 9].

Different from the deterministic systems based on equation (2.2), in this section, we start with a time-homogeneous stochastic process $\{x_t\}_{t \in \mathbb{R}_0^+}$ defined on the space X with a probability measure $\mathbb{P}$. In this setting, the formulation of the Koopman operator and its adjoint, the Frobenius-Perron operator, can be stated in terms of the transition density function [19].

Definition 2.1. *The transition density function* $p : \mathbb{R}_0^+ \times X \times X \to \mathbb{R}_0^+$ *of a process* $\{x_t\}_{t\in\mathbb{R}_0^+}$ *is defined by*

$$\mathbb{P}[x_{t+\tau} \in A | x_t = x] = \int_A p(\tau, x, y)dy \tag{2.8}$$

for all measurable sets $A \subseteq X$.

Note that for a deterministic system with flow ϕ, the transition density function collapses to a Dirac delta centered on the point $\phi(t, x)$:

$$p(t, x, y) = \delta_{\phi(t,x)}(y). \tag{2.9}$$

For $1 \leq p \leq \infty$, the spaces $L^p(X, \mu)$ denote the p-integrable functions w.r.t. μ (see equation (2.3) for $p = 2$). The Koopman operator and its adjoint can now be defined as follows.

Definition 2.2. *Let* $q(t, \cdot) \in L^1(X, \mu)$ *be a probability density and* $f(t, \cdot) \in L^\infty(X, \mu)$ *an observable of the system with flow* ϕ. *For a given time step* $\Delta t \in \mathbb{R}^+$, *the Koopman operator* $K^{\Delta t} : L^\infty(X, \mu) \to L^\infty(X, \mu)$ *is defined by*

$$K^{\Delta t} f(t, x) = \int_X p(\Delta t, x, y) f(t, y) d\mu(y) = \mathbb{E}\left[f(t, x_{t+\Delta t}) | x_t = x\right], \tag{2.10}$$

and the Frobenius–Perron operator $P^{\Delta t} : L^1(X, \mu) \to L^1(X, \mu)$ *is defined by*

$$P^{\Delta t} q(t, x) = \int_X p(\Delta t, y, x) q(t, y) d\mu(y). \tag{2.11}$$

The Koopman and Frobenius–Perron operators are adjoint w.r.t the inner product on a function space $\mathcal{F}$ if for all $f, g \in \mathcal{F}$,

$$\left\langle K^{\Delta t} f, g \right\rangle_{\mathcal{F}} = \left\langle f, P^{\Delta t} g \right\rangle_{\mathcal{F}}. \tag{2.12}$$

If for a given stochastic dynamical system the transition function satisfies the detailed balance condition

$$p(t, x, y) = p(t, y, x)$$

for all $t \in \mathbb{R}^+$ and all states $x, y \in X$, the Koopman operator is self-adjoint (and hence equal to the Frobenius–Perron operator) on $L^1(X) \cap L^\infty(X)$, which is a direct consequence of the definitions of the operators (equations (2.10–2.11)).

The Frobenius–Perron and Koopman operators are the solution operators of the forward (Fokker–Planck) and backward Kolmogorov equations, respectively [22, section 11]. That is why they are also often referred to as *forward* and *backward* (transfer) operators.

General Smoluchowski equations of a d-dimensional system are given by

$$\mathrm{dx}_t = -D\nabla V(\mathrm{x}_t)\mathrm{d}t + \sqrt{2dD}\mathrm{dW}_t, \tag{2.13}$$

with potential $V : X \to \mathbb{R}$, diffusion coefficient D, and Wiener process W_t. They are time-reversible [28, 23], and so their transition density function satisfies the detailed balance condition. Therefore, the two transfer operators are identical (self-adjoint). The fact that this holds for equations of type (2.13) is extremely strong and useful, particularly for the analysis of molecular dynamics simulators. As many algorithms can be reformulated in this setting, too, these strong results can be carried over directly.

2.3. Illustrative example: Koopman operator of the Euler method. We use a typical example from the stability analysis of algorithms to familiarize the reader with the concept of Koopman operators of algorithms. Consider the following ordinary differential equation (ODE) with parameter $a > 0$,

$$\frac{d}{dt} S_t(x)\Big|_{t=0} = -ax, \ x \in \mathbb{R}, \ t \geq 0, \tag{2.14}$$

and solution $S_t(x) = \exp(-at)x$. A trajectory $S_t(x_0)$ can be approximated with a numerical algorithm, an initial value solver, given only the ODE (2.14) and an initial condition $x_0 \in \mathbb{R}$. Here, we choose the forward Euler method, compute its Koopman operator, and show how stability analysis of the method is related to the spectrum of the operator. For a given step size $\Delta t > 0$, the forward Euler method is given through the following iterative scheme, starting at x_0 for $n = 0$:

$$x_{n+1} = x_n + \Delta t \ \frac{d}{dt} S_t(x_n)\Big|_{t=0}. \tag{2.15}$$

For the ODE (2.14), we thus have the linear, discrete-time dynamical system

$$x_{n+1} = x_n - a\Delta t x_n = (1 - a\Delta t)x_n. \tag{2.16}$$

We now choose the function space $\mathcal{F} = \text{span}\{x^k, x \in \mathbb{R}, k \in \mathbb{N}\}$, with the identity function $g(x) = x$ as the generator of the basis. The Koopman operator $\mathcal{K}^n : \mathcal{F} \to \mathcal{F}$ applied to the generator is

$$[\mathcal{K}^n g](x) = g((1 - a\Delta t)^n x) = (1 - a\Delta t)^n g(x),$$

which shows that g is an eigenfunction of $\mathcal{K}^n$ associated to the eigenvalue $(1 - a\Delta t)$. On the space $\mathcal{F}$, the spectrum of $\mathcal{K}$ consists of the eigenvalues $\omega_k = (1-a\Delta t)^k$, $k \in \mathbb{N}$, associated to the eigenfunctions $\phi_k(x) = g(x)^k = x^k$. It is easy to see that for $a\Delta t \in [0, 2]$, $\mathcal{K}^n$ is a contraction operator on $\mathcal{F}$: for all k, $|\lambda_k| \leq 1$. This is a sufficient condition for the numerical stability of the Euler method in this region. Note that system (2.14) has eigenvalues $\lambda_k = \exp(-ka)$ on $\mathcal{F}$, and so the discrete system (2.16) defined by the Euler method is not conjugate to the continuous system for any $\Delta t > 0$ (for conjugacy, the eigenvalues would have to be the same). Consistency–and thus, convergence–of the algorithm can now also be interpreted in the spectral sense, by setting $n := \lceil 1/\Delta t \rceil$ (rounding to the next largest integer) and applying the formula

$$\lim_{n\to\infty} \left(1 - \frac{a}{n}\right)^n = \exp(-a).$$

After this pedagogical example, we show how Koopman operators can provide a useful framework to understand more general algorithms. In a more applied setting, "the algorithm" is usually a large, complex piece of software that is treated as a black box data generator (for example, an optimization procedure for a supply chain). Then, numerical approximations of the Koopman operator can be used. In the next section, we briefly describe Extended Dynamic Mode Decomposition as an example for such an approximation procedure from data.

3. Data-driven approximation of the Koopman operator. We briefly cover the approximation methods employed in the computational experiments of later sections. The methods include an approximation of the Koopman operator with pure-point spectrum (Extended Dynamic Mode Decomposition), extracting the infinitesimal generator of the family of Koopman operators, as well as an approximation algorithm in case the Koopman operator is unitary and has continuous spectrum.

3.1. Extended Dynamic Mode Decomposition. One extension of Dynamic Mode Decomposition (DMD) [40] to nonlinear systems with pure-point spectrum is Extended DMD [46, 48, 47] (EDMD). Many more numerical methods exist, such as Generalized Laplace Averaging [30, 27], EDMD with dictionary learning through neural networks [24], dictionary selection through L^1-optimization [5], etc.

We consider the state space of a given system–usually, a smooth manifold M embedded in Euclidean space $\mathbb{R}^d$–sampled with a finite number of points $X = \{x_k \in M \subset \mathbb{R}^d | k = 1, \dots, N_X\}$, $N_X \in \mathbb{N}$. The main idea of (Extended) Dynamic Mode Decomposition is to approximate the action of the operator on a function space $\mathcal{F}$ over M by choosing a finite-dimensional subspace spanned by a finite set of real-valued functions of M (called a "dictionary", with "observables" as elements). For $\mathcal{F}$, we typically use the space $L^2(M, \mathbb{C}, \mu)$, see equation (2.3). The EDMD algorithm computes the action of the Koopman operator on the dictionary at the points in X, and then approximates the Koopman operator by solving a least-squares problem:

1. Given a dictionary $D = \{d_k : M \to \mathbb{R} | k = 1, \dots, N_D\} \subset \mathcal{F}$ with N_D observables, construct

$$G = D(X) = [d_1(X), d_2(X), \dots, d_{N_D}(X)] \in \mathbb{R}^{N_X \times N_D}.$$

2. For a fixed $t > 0$, compute the action of the Koopman operator $\mathcal{K}^t$ on the dictionary elements using the flow map $S_t : M \to M$ of the given system:

$$[\mathcal{K}^t d_k](x) = (d_k \circ S_t)(x),$$

and construct the matrix

$$A = [\mathcal{K}^t D](X) = \left[\mathcal{K}^t d_1(X), \mathcal{K}^t d_2(X), \dots, \mathcal{K}^t d_{N_D}(X)\right] \in \mathbb{R}^{N_X \times N_D}.$$

3. Approximate the operator $\mathcal{K}^t$ through the matrix

$$K = \frac{1}{N_D^2}(G^T G)^\dagger (A^T A) \in \mathbb{R}^{N_D \times N_D},$$

where $(G^T G)^\dagger$ denotes the pseudo-inverse of the matrix $(G^T G)$.

The choice of the dictionary D is crucial for a successful approximation of the operator, see [24, 46, 47] and references therein. In this paper, we employ thin-plate radial basis functions [10],

$$d_x(y) = \|x - y\|^2 \ln(\|x - y\| + \delta), \ x, y \in X, \ \delta > 0, \tag{3.1}$$

where the point x is the center of the radial basis function and δ is a small, positive constant to extend the function to points $y = x$ (here: $\delta = 10^{-3}$). The centers x are uniformly distributed over the data set X, where the uniform distribution is constructed through an appropriate k-means algorithm, clustering the data in N_D clusters and using the centers of the clusters as the centers for the basis functions.

3.2. Approximation of the infinitesimal generator. After some of the eigenfunctions $\hat{\phi}$ and eigenvalues λ of $\mathcal{K}$ are available through their numerical approximations, we can represent the approximation of the infinitesimal generator $A_{\mathcal{K}}$ for the group $\mathcal{K}^t$ as a matrix $L = V \frac{\ln[\Lambda]}{\Delta t} V^{-1}$ (see [26]), where V contain the eigenvectors of the matrix representation of a particular $\mathcal{K}^{\Delta t}$, obtained with EDMD from snapshot data $(x, S_{\Delta t}(x))$. For the coordinate functions $x_i \in \mathcal{F}$ and a finite step size Δt,

$$A_{\mathcal{K}} x_i = \lim_{t \to 0} \frac{(\mathcal{K}^t - I) x_i}{t} \approx \frac{\ln[\Lambda]}{\Delta t} \hat{\phi}(x)\, C_i, \tag{3.2}$$

where Λ is the diagonal matrix of eigenvalues of $\mathcal{K}$, ln is the complex logarithm (where we pick the principal branch), and C are the "Koopman modes" associated to the coordinate functions $x_i : X \to \mathbb{R}$. The approximation of the vector field v generating the flow S_t is thus

$$v(x) \approx \frac{\ln[\Lambda]}{\Delta t} \hat{\phi}(x) C. \tag{3.3}$$

Note that the approximation of the generator through the logarithm of the eigenvalues involves several challenges: numerical issues with eigenvalues of Λ close to zero, and non-invertibility (or rather, non-uniqueness) of the complex logarithm. The issue with non-uniqueness is noted by Mauroy and Goncalves [26]. In the example where we construct L (section 4.2), the spectrum of the Koopman operator lies in the unit disk, so the complex logarithm is unique when picking its principal branch. To mitigate numerical issues, we set all eigenvalues with $\mathrm{Re}(\ln \lambda) < -2/\Delta t$ to zero.

3.3. Approximation of the continuous spectrum. The EDMD algorithm is applicable for approximating the Koopman operator if its spectrum only consists of eigenvalues (pure-point spectrum). Not many numerical approximations are available for operators with continuous spectra. Here, we briefly describe the results from Korda, Putinar and Mezić [21] for unitary Koopman operators, where the spectrum is concentrated on the unit circle $\mathbb{T}$ in the complex plane. We will use this approximation to analyze the Newton algorithm in section 4.4. If the Koopman operator of a system is unitary, we can write it as an integral over a projection operator-valued measure E on the unit circle $\mathbb{T} \subset \mathbb{C}$,

$$\mathcal{K} = \int_{\mathbb{T}} z dE(z).$$

For any observation function g in the domain $\mathcal{F}$ of $\mathcal{K}$, the measure E defines a real-valued, positive measure μ_g on $\mathbb{T}$ through

$$\mu_g(A) := \langle E(A) g, g \rangle_{\mathcal{F}}$$

for all Borel sets $A \subset \mathbb{T}$. The moments m_k, $k \in \mathbb{Z}$ of μ_g are defined by

$$m_k := \int_{\mathbb{T}} z^k d\mu_g(z) = \left\langle \mathcal{K}^k g, g \right\rangle_{\mathcal{F}},$$

where the last identity follows from the spectral theorem. The measure μ_g depends on the choice of observable g, but under certain conditions on g, μ_g fully determines the operator $\mathcal{K}$. The conditions are satisfied if g is $\star$-cyclic, meaning that repeated applications of $\mathcal{K}$ to g span the function space $\mathcal{F}$, see [21]. In this case, approximating the measure μ_g not only reveals information about the specific observable g, but also about the underlying system (and its operator $\mathcal{K}$). The density of μ_g on the unit circle can be visualized after a numerical approximation of the moments. If we have N observations $y_i = g(x_i)$ along a trajectory of an ergodic system, we can estimate the moments through

$$m_k \approx \frac{1}{N-k} \sum_{i=1}^{N-k} y_{i+k} \bar{y}_i.$$

Define $\psi_N(z) = [1, z, z^2, \ldots, z^N]^T$, and write the Christoffel–Darboux kernel as

$$K_N(z, w) = \psi_N(w)^H M^{-1} \psi_N(z),$$

with a semi-positive definite, Hermitian Toeplitz matrix M defined through

$$M := \begin{bmatrix} m_0 & \overline{m}_1 & \cdots & \cdots & \cdots & \overline{m}_N \\ m_1 & m_0 & \overline{m}_1 & \ddots & \ddots & \overline{m}_{N-1} \\ \vdots & m_1 & m_0 & \ddots & \ddots & \overline{m}_{N-2} \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & m_0 & \overline{m}_1 \\ m_N & \cdots & \cdots & \cdots & m_1 & m_0 \end{bmatrix}.$$

Inverting the matrix M is non-trivial, see [21] for details. Instead of inverting M directly, the authors propose to invert a matrix $\tilde{M}$ where the elements m_k in M are replaced by elements

$$\tilde{m}_k := \begin{cases} m_k + 1 & \text{if } k = 0, \\ m_k & \text{if } k > 0. \end{cases}$$

Given the kernel $\tilde{K}_N$ (now using $\tilde{M}$ instead of M), the continuous density ρ of μ_g at a given angle $\theta \in \mathbb{T} \cong [0, 2\pi)$ can be approximated through

$$\frac{N+1}{\tilde{K}_N(\exp(i\theta), \exp(i\theta))} - 1 \approx \rho(\theta).$$

4. Analysis of algorithms through their Koopman operator. This section contains illustrative examples of algorithm analysis through their Koopman operator. We demonstrate several aspects of our solution to the main challenge: how to analyze complex, nonlinear algorithms in a data-driven setting within a unified framework. In particular, we show

1. how to accelerate algorithms by constructing data-driven surrogate models,
2. approximations of infinitesimal generators for discrete-time algorithms,
3. approximations of the (asymptotic, long-time) operator in high-dimensional state spaces,
4. and a combination of an explicit operator analysis for the Newton-Raphson method with a data-driven approximation of its continuous spectrum.

4.1. Acceleration. We can employ the predictive capabilities of the Koopman operator framework to accelerate the application of algorithms. First, the Koopman operator eigenvalues, eigenfunctions, and modes are constructed in an offline phase, using EDMD (see section 3). Second, given a new initial condition close to the ones in the data set used for constructing the operator, we can use the approximation as a data-driven surrogate model for the original algorithm.

Such a surrogate model may be useful in accelerating the training of neural networks. There, the state space is often very high-dimensional, because each state consists of all weights and biases of the network. The typical optimization algorithm, stochastic gradient descent, employed to change the state in order to minimize a certain loss function, is based on the gradient descent algorithm we study in this section. The stochastic version can be analyzed similarly, using the setting described in section 2.2.2. Different types of training (e.g. through mini-batches, or dropout) will induce different biases, and the Koopman operator based surrogate may provide a path towards quantifying and comparing such inductive biases.

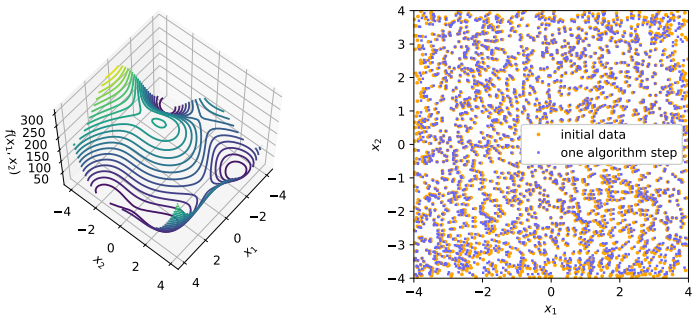
Figure 2 shows the setting used for the example in this section, where we study the Gradient Descent algorithm minimizing Himmelblau's function [18]—a standard test function for optimization algorithms:

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$

This function has one local maximum and four local minima with three connecting saddle points (see figure 1, a). As a dictionary for EDMD, we employ 500 thin-plate radial basis functions (see equation (3.1)), chosen uniformly distributed on $M = [-4, 4]^2$. In addition to the radial basis functions, we add the two coordinate functions x_1, x_2 as well as the constant function with value 1 to the dictionary.

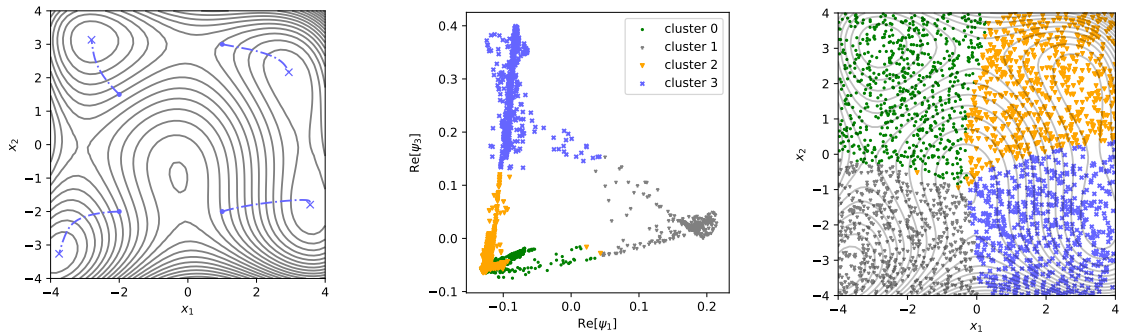
Figure 2 shows four predicted trajectories using the approximated Koopman operator. To generate a trajectory, we follow the prediction steps outlined in section 2.2 (equation (2.5)), using all 503 eigenvalues and eigenfunctions.

The eigenfunctions of $\mathcal{K}^t$ at eigenvalue $\lambda = 1$ are special, as they are preserved under the flow of the system—they do not decay, expand, or oscillate. This can be used to construct an *ergodic decomposition* [6, 7] of the state space, separating the basins of attraction. In the system discussed here, we expect to see four basins of attraction, corresponding to four attracting fixed points, and accordingly, at least four eigenfunctions associated to eigenvalue $\lambda = 1$. To approximate the ergodic decomposition, we consider the map $M \ni p \mapsto [\phi_1(p), \dots, \phi_4(p)]^T \in \mathbb{C}^4$, i.e. from the states M into the values of eigenfunctions associated to $\lambda = 1$. In this space (shown projected to two dimensions in figure 2, b), there should be four clusters of values corresponding to the different values the eigenfunctions take on over the corresponding basins of attraction. Clustering the data using the k-means



(a) (b)

Figure 1. *Level sets of Himmelblau's function (a), and sample data* (y_n, y_{n+1}) *from Gradient Descent (b), where* y_{n+1} *is generated through one iteration step with* $\Delta t = .001$.



(a) (b) (c)

Figure 2. *Panel (a) shows predicted trajectories from four initial points. All four trajectories approach the correct minima in their respective basin of attraction. An ergodic decomposition of the state space through k-means clustering of values from four complex-valued Koopman eigenfunctions associated to eigenvalues close to 1 (a projection of the eight-dimensional data to two dimensions is shown in panel b) separates the four basins of attraction (c).*

algorithm then allows us to distinguish the original points in M (plot c in figure 2).

4.2. Continuous versions of discrete algorithms. In recent years, extensive research has focused on re-deriving established algorithms as discrete versions of continuous dynamical systems, in this way "explaining" the original, iterative algorithm. In the case of the Nesterov method, the appropriate continuous system derived by Candès et al. [43] is a time-dependent, second-order ordinary differential equation. This ODE can be (trivially) transformed to a first-order system of autonomous equations, given through

$$\dot{x} = v, \ \dot{v} = -\frac{r}{t}v - \nabla f(x), \ \dot{t} = 1, \tag{4.1}$$

where $\dot{x}$ now refers to a derivative with respect to a new time variable. In the example we discuss below, the function to minimize is $f(x) = \frac{1}{2}x^2$, $x \in \mathbb{R}$. As in the previous section,

we use EDMD to approximate the eigensystem of the Koopman operator with 125 thin-plate radial basis functions (see equation (3.1)) as well as the x, v, t coordinates and the constant function as a basis for the function space of observables on $\mathbb{R}^3$ ((x, v, t)-space). The operator approximation is performed with data (z_n, z_{n+1}) from the discrete algorithm with step size $h = 0.01$ (see table 1), where $z_n := (x(n), (x(n) - x(n-1))/h, t(n))$. We then approximate the corresponding vector field on the three coordinates of z through equations (3.2) and (3.3), which should be similar to the vector field in equation (4.1). Figure 3 shows streamlines of the vector field (4.1) for three different values of t, with a distinguished sample trajectory, compared to the vector field approximated by using the Koopman operator. This example illustrates that it is possible to extract a continuous version (an infinitesimal generator) of a discrete algorithm.

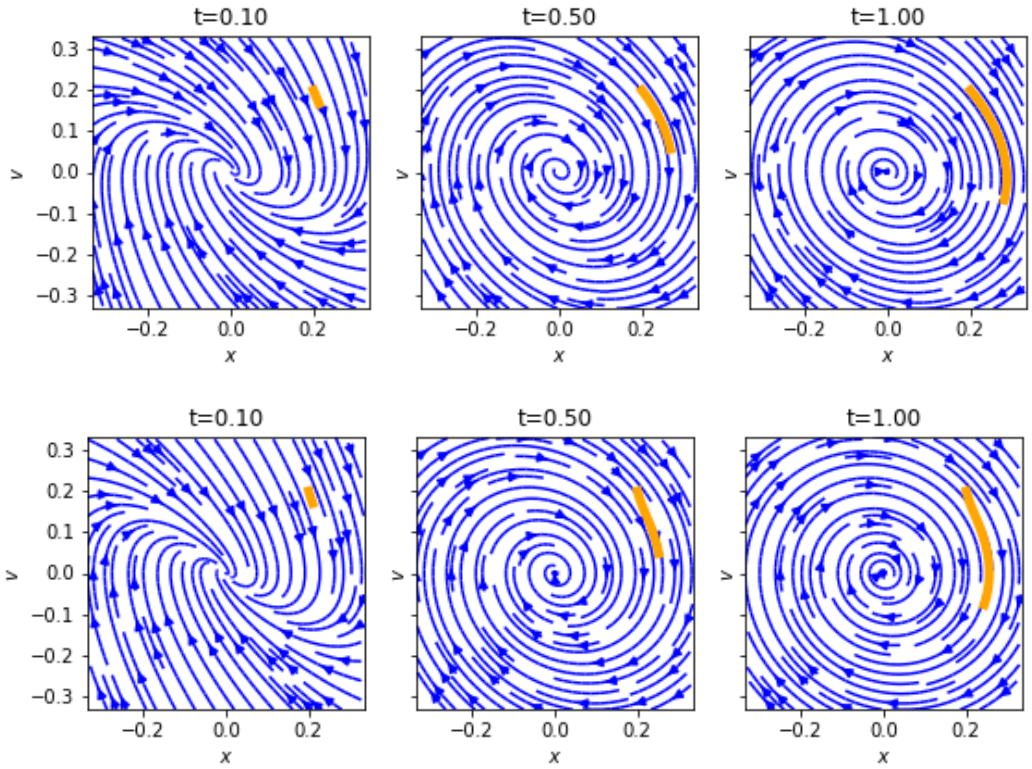


Figure 3. *Approximation of the vector field through EDMD (top row), and the ODE found by Candès et al. [43] (bottom row), at times $t \in \{\frac{1}{10}, \frac{1}{2}, 1\}$. The orange trajectory shows an example solution over this (time-dependent!) vector field.*

4.3. High-dimensional state spaces. Many algorithms are operating on points in high-dimensional state spaces that are parametrized through a large number of variables. In this case, sampling the full state space to obtain an accurate approximation of the Koopman operator for the algorithm is difficult, especially when the action of the algorithm on the state is nonlinear. However, if a few iterations of the algorithm quickly contract the whole state

space towards a low-dimensional subset, then numerical approximations can be successful in reducing the overall computational effort. The EDMD algorithm does not need an explicit expression for this low-dimensional subset if, for example, radial basis functions are used for the dictionary.

As an example, we construct a high-dimensional optimization problem with a low-dimensional, attracting manifold, where gradient descent is used as an optimizer. On the manifold, starting from the Müller-Brown potential, we add quadratic terms to obtain asymptotically stable, attractive behavior in the directions transverse to the manifold. On the manifold (here, a two-dimensional subspace of $\mathbb{R}^{100}$ parametrized by x, y), the potential is defined by

$$V(x,y) = \sum_{k=1}^{4} A_k \exp\left(a_k(x-x_k^0)^2 + b_k(x-x_k^0)(y-y_k^0) + c_k(y-y_k^0)^2\right),$$

with

$$\begin{aligned} A &= (-200, -100, -170, 15), \\ a &= (-1, -1, -6.5, 0.7), \\ b &= (0, 0, 11, 0.6), \\ c &= (-10, -10, -6.5, 0.7), \\ x^0 &= (1, 0, -0.5, -1), \\ y^0 &= (0, 0.5, 1.5, 1). \end{aligned}$$

The cost function $C : \mathbb{R}^{100} \to \mathbb{R}$ on the embedding space is then defined through

$$C(\mathbf{x}) = V([U\mathbf{x}]_1, [U\mathbf{x}]_2) + \sum_{i=0}^{98} (U\mathbf{x})_i^2, \ \mathbf{x} \in \mathbb{R}^{100},$$

where we rotate the data by a random, unitary matrix $U \in \mathbb{R}^{100\times 100}$ into the 100 dimensional space to demonstrate that the EDMD algorithm with radial basis functions does not depend on the chosen embedding coordinate system. Now, all coordinates have non-trivial dynamics, but the low-dimensional, attractive manifold is still present. To approximate the operator, we use 625 thin-plate radial basis functions with centers randomly distributed over the initial data set in the 100-dimensional space, as well as the 100 coordinates $x_1, \ldots, x_{100}$ and the constant function (i.e., the final dictionary has 726 elements). The initial data consists of 2500 data points, sampled in a 100-dimensional standard normal distribution around zero, and then evolved forward with five iterations of the algorithm to converge towards the low-dimensional structure. *Note that this number of points is–by far–not enough to densely sample the high-dimensional space,* but the attracting behavior of the algorithm towards the low-dimensional structure is nonetheless sufficient to numerically approximate the operator. We use radial basis functions for EDMD to circumvent the need for an explicit approximation of the low-dimensional space: the functions only depend on the distance between the points, not the ambient dimension. As in section 4.1, we can again accurately predict trajectories of gradient descent (see figure 4).

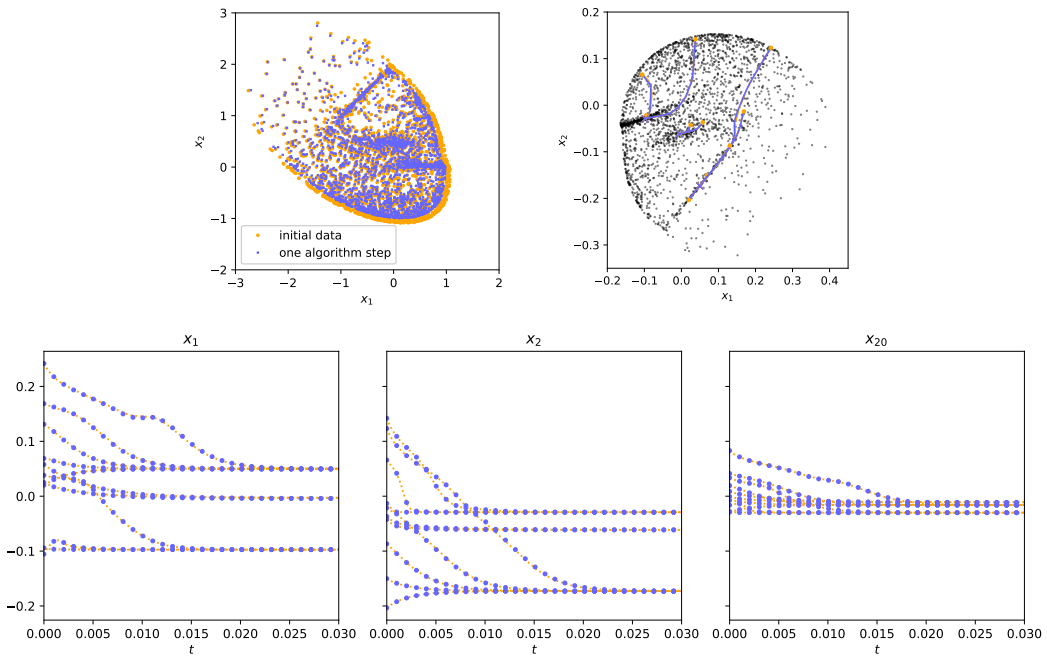


Figure 4. *Top-Left: Data gathered by sampling the 100-dimensional state space with a Gaussian distribution, converging to the low-dimensional manifold in five iterations of gradient descent (end points marked in orange), and then recording one additional iteration (blue points). The coordinates x_1 and x_2 here are only the first two of all 100 present in the data set. Top-Right: Predicted trajectories of the algorithm (blue) that were computed after the construction of the operator, for ten initial conditions chosen at random from the initial data set (orange points). The bottom panels show predicted (blue, dots) and actual (orange, dashed) trajectories for the coordinates x_1, x_2, and x_{20}, to demonstrate that all coordinate trajectories are reconstructed accurately.*

4.4. Koopman operators for the Newton-Raphson method. In this section, we mostly use the conjugacy results from [32] to construct eigenfunctions for the Koopman operator of the Newton-Raphson method for root finding. We only consider polynomial functions of degree two in this analysis, and at the end give a hint of more complicated dynamics present for cubic polynomials. The Newton-Raphson method to find roots of a function f is defined as the iterative scheme

$$x_{n+1} = x_n - [Jf]^{-1}(x_n) f(x_n) =: N_f(x_n),$$

where $[Jf]_{ij} = \frac{\partial f_i}{\partial x_j}$ is the Jacobian matrix of f. In the example we consider here, the function $f : \mathbb{C} \to \mathbb{C}$ is a complex polynomial of degree two, for which the Newton-Raphson method simplifies to

$$N_f(z) = z - \frac{f(z)}{f'(z)}.$$

Without loss of generality, we assume f has the form $f(z) = az^2 + bz + d$ with $a, b, d \in \mathbb{C}$, $a \neq 0$. Any polynomial f of degree two is conjugate to exactly one polynomial $g_c(z) = z^2 + c$, $c \in \mathbb{C}$ (see [32]), where the conjugacy is given through a linear transformation $h(z) = az + \frac{b}{2}$, such that

$$f \circ h = h \circ g_c.$$

The constant c is related to a, b, d through $c = ad + \frac{b}{2} - \frac{b^2}{4}$. The map h scales and translates the argument so that g_c has the desired form. The conjugacy of f and g_c implies that we only need to analyze the Newton-Raphson method applied to functions of the form g_c, with a given c. For this simpler form of f, the Newton-Raphson method becomes

$$N_c(z) = z - \frac{z^2 + c}{2z} = \frac{z^2 - c}{2z}.$$

In the Koopman operator picture, the map N_c defines the dynamical system in discrete time with an associated Koopman operator $\mathcal{K}_{N_c}$. An important result in [32] is that the map N_c is conjugate to the polynomial $g_0(z) = z^2$ through the map

$$h_0(z) = \frac{z + i\sqrt{c}}{z - i\sqrt{c}},$$

which is a Möbius transformation, and hence invertible on $\mathbb{C} \cup \{\infty\}$. Thus, eigenfunctions of $\mathcal{K}_{N_c}$ can be constructed through eigenfunctions of $\mathcal{K}_{g_0}$, which are defined through

$$[\mathcal{K}_{g_0}\phi_k](z) = (\phi_k \circ g_0)(z) = \lambda_k \phi_k(z), \ k \in \mathbb{Z}.$$

Since $g_0(z) = z^2$, there are eigenfunctions ϕ_k of the form $\phi_k(z) = \ln(|z|)^k$, associated to the eigenvalues $\lambda_k = 2^k$. Through the given conjugacy h_0, we have that

$$\psi_k(z) = (\phi_k \circ h_0)(z) = \ln\left(\left|\frac{z + i\sqrt{c}}{z - i\sqrt{c}}\right|\right)^k \tag{4.2}$$

are eigenfunctions for $\mathcal{K}_{N_c}$, associated to the eigenvalues λ_k. Fig. 5 shows ψ_1 for $c = 1$, i.e. for the polynomial $f(z) = z^2 + 1$. The roots at $\pm i$ are clearly visible as the points where the eigenfunction diverges.

4.4.1. An example with chaotic behavior. Consider the function $f(z) = z^2 + 1$ from the previous section, with its two roots at $\pm i$. Interpreting the Newton-Raphson method as a discrete dynamical system, the real line is the basin boundary separating the two basins of attraction on the complex plane. When started exactly on the real line, the Newton-Raphson method exhibits chaotic behavior, which we analyze in this section. The Newton-Raphson method applied to f is defined by the map

$$N_f(z) = z - \frac{z^2 + 1}{2z} = \frac{z^2 - 1}{2z}, \tag{4.3}$$

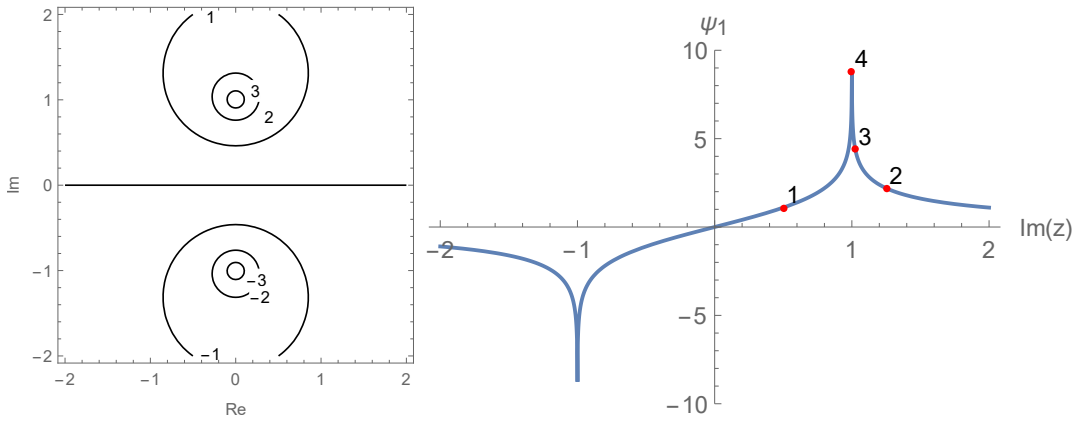


Figure 5. *The eigenfunction ψ_1 of $\mathcal{K}_{N_c}$ for $a = 1$, $b = 0$, $c = 1$, and $d = 1$, with corresponding polynomial $f(z) = z^2 + 1$. Left: Plot on the complex plane, with contour lines labeled by the function values. Right: Eigenfunction evaluated on $[-2i, 2i]$. The numbered, red dots show four evaluations of the Newton-Raphson method with starting point $z_1 = 0.5i$, $\psi_1(z_1) \approx 1.0986$. As expected, $\psi_1(z_{k+1}) = 2^k \psi_1(z_1)$ and the function diverges at the roots $\pm i$ of f.*

so that $z_{n+1} = N_f(z_n)$ yields a new iterate, starting at $z_0 \in \mathbb{C}$. For $z_0 \in \mathbb{C}/\mathbb{R}$, the limit points of the iterative scheme are the extremal points, where

$$\begin{array}{rcll} \lim_{n\to\infty} z_n & = & i & \mathrm{Im}(z_0) > 0, \\ \lim_{n\to\infty} z_n & = & -i & \mathrm{Im}(z_0) < 0. \end{array}$$

For $z_0 \in \mathbb{R}$, however, the scheme does not converge (see Fig. 6) and gives rise to the chaotic recurrence (which can be solved explicitly in this case)

$$z(n) = -\cot\left(c_1 2^n\right), \; c_1 = \arctan(-1/z_0). \tag{4.4}$$

A similar discrete system with explicit solution is given by the logistic map at parameter $r = 4$, with recurrence relation $y_{n+1} = r y_n(1 - y_n)$ and solution $y_n = \sin(c_0\, 2^n)^2$, $c_0 = \arcsin\left(\sqrt{y_0}\right)$. Accordingly, except for the constant function, the continuous eigenfunctions of the Koopman operator computed in the previous section are all zero (or infinity, for $k < 0$) on the real line, see equation (4.2). For the map N_f, the real line is an invariant, repelling limit set of measure zero.

The results so far have been constructed explicitly. Addressing the main challenge of the manuscript, we now want to demonstrate that numerical algorithms can also be used to analyze the behavior of the Newton-Raphson algorithm. To this end, we employ the numerical method by Korda, Putinar and Mezić [21] (as described in section 3.3) to approximate the continuous spectrum of the Koopman operator associated to the map N_f. Figure 7 shows how different numbers of points (i.e. lengths of trajectories) approximate the measure μ_g for $g(z) = 1 + \exp(2\pi i z)$, with $z \in \mathbb{R}$ from a trajectory of the Newton-Raphson method starting at $z_0 = 1/2$. The point spectrum is $\sigma_{pp} = \{1\}$ with eigenvalue 1 associated to the constant function, as expected for a chaotic system. The absolutely continuous spectrum appears to be $\sigma_c = \mathbb{T}/\{1\}$, with no singularly continuous part. The trajectories were computed with

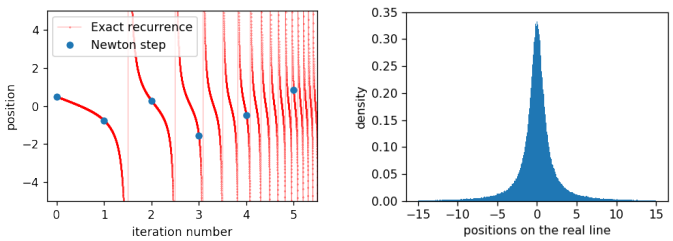


Figure 6. *Left: The first five iterations of the Newton-Raphson method (blue dots) on the polynomial $f(z) = z^2 + 1$, with the initial condition $z(0) = 1/2$. The exact recurrence relation is shown as red lines, evaluated over a continuous space, showing that the expansion and folding leads to faster and faster oscillations, a sign for chaotic behavior on the discrete iterations. Right: Invariant density over the real line, approximated from a single trajectory of length $n = 10^6$, starting at $z_0 = 0.3$.*

the iteration of N_f (equation (4.3)) with 16 digits of accuracy, and with the explicit formula (equation (4.4)) with 35,000 digits of accuracy, using the Wolfram Mathematica 11 software. The high accuracy is necessary to exactly represent the number $2^{100,000} \approx 10^{31,000}$ for the last point in the longest trajectory.

4.4.2. Fractal eigenfunctions. For cubic polynomial functions, application of the Newton-Raphson method is known to exhibit an even richer structure compared to the quadratic polynomials described in the previous section. We only give numerical results, inspired by [32]. Figure 8 shows the number of iterations (in color) of the Newton-Raphson method until the derivative of the objective function is smaller than 0.01. Figure 9 shows eight iterations of the Newton-Raphson method on a subset of points sampled on the complex plane, for a polynomial of degree two (left three columns), and degree three (right three columns). The convergence to the two (and three) roots is visible in the third and sixth column. The first/second and fourth/fifth columns show the real/imaginary parts of the final positions as functions on the initial positions, which makes the fractal structure for degree three (and higher) apparent.

5. Conclusions. Many modern algorithms (e.g. those associated with training neural networks) can difficult to analyze in a classical sense: they are complex, interconnected, can be deterministic or stochastic, act on high-dimensional states, and are often only accessible as input-output blax box systems. To address this challenge, we propose to employ the unifying view of the Koopman operator framework, with several numerical approximations readily available. In a series of examples, we demonstrated that the Koopman operator provides such a unifying view of many important issues in the analysis of algorithms: (spectral) convergence, algorithm acceleration, state space decomposition, high-dimensional state spaces, partial information, and generating processes for discrete algorithms. In the last section, we gave an outlook to chaotic behavior of algorithms and how the behavior of statistics of observables can be analyzed through the spectrum of the operator. We discussed the possibility of using this approach to accelerate algorithms in high-dimensional embedding spaces whose long-term dynamics lie on low-dimensional manifolds. Another possibility is to use this approach to create surrogates of complicated "black box" algorithms that are difficult to analyze mathematically; the data-driven surrogates obtained by sampling iterated

algorithm states may provide useful insights in their nature and behavior. The Koopman operator has been shown to provide a convenient framework for constructing data-driven homeomorphisms between dynamical systems; it is an intriguing possibility that we can use this framework to realize homeomorphisms between different algorithms for solving the same problem.

REFERENCES

[1] T. Berry and J. Harlim, *Nonparametric uncertainty quantification for stochastic gradient flows*, SIAM/ASA Journal on Uncertainty Quantification, 3 (2015), pp. 484–508, https://doi.org/10.1137/14097940x.

[2] A. Bhaya, F. A. Pazos, and E. Kaszkurewicz, *Comparative study of the CG and HBF ODEs used in the global minimization of nonconvex functions*, in Artificial Neural Networks – ICANN 2009, Springer Berlin Heidelberg, 2009, pp. 668–677, https://doi.org/10.1007/978-3-642-04274-4_69.

[3] E. M. Bollt, Q. Li, F. Dietrich, and I. Kevrekidis, *On matching, and even rectifying, dynamical systems through Koopman operator eigenfunctions*, SIAM Journal on Applied Dynamical Systems, 17 (2018), pp. 1925–1960, https://doi.org/10.1137/17m116207x.

[4] R. Brockett, *Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems*, Linear Algebra and its Applications, 146 (1991), pp. 79–91, https://doi.org/10.1016/0024-3795(91)90021-n.

[5] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proceedings of the National Academy of Sciences of the United States of America, 113 (2016), pp. 3932–3937, https://doi.org/10.1073/pnas.1517384113.

[6] M. Budisic and I. Mezic, *An approximate parametrization of the ergodic partition using time averaged observables*, in Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, IEEE, Dec. 2009, https://doi.org/10.1109/cdc.2009.5400512.

[7] M. Budišić, R. Mohr, and I. Mezić, *Applied Koopmanism*, Chaos, 22 (2012), p. 047510, https://doi.org/10.1063/1.4772195.

[8] M. T. Chu, *Linear algebra algorithms as dynamical systems*, Acta Numerica, 17 (2008), pp. 1–86, https://doi.org/10.1017/s0962492906340019.

[9] N. Črnjarić-Žic, S. Maćešić, and I. Mezić, *Koopman operator spectrum for random dynamical systems*, Journal of Nonlinear Science, (2019), https://doi.org/10.1007/s00332-019-09582-z.

[10] J. Duchon, *Splines minimizing rotation-invariant semi-norms in sobolev spaces*, in Constructive Theory of Functions of Several Variables, Springer Berlin Heidelberg, 1977, pp. 85–100, https://doi.org/10.1007/bfb0086566.

[11] S. Edvardsson, M. Neuman, P. Edström, and H. Olin, *Solving equations through particle dynamics*, Computer Physics Communications, 197 (2015), pp. 169–181, https://doi.org/10.1016/j.cpc.2015.08.028.

[12] J. Eldering, M. Kvalheim, and S. Revzen, *Global linearization and fiber bundle structure of invariant manifolds*, Nonlinearity, 31 (2018), pp. 4202–4245, https://doi.org/10.1088/1361-6544/aaca8d.

[13] K.-J. Engel and R. Nagel, *A Short Course on Operator Semigroups*, Springer New York, 2006, https://doi.org/10.1007/0-387-36619-9.

[14] D. Giannakis, *Data-driven spectral decomposition and forecasting of ergodic dynamical systems*, Applied and Computational Harmonic Analysis, (2017), https://doi.org/10.1016/j.acha.2017.09.001.

[15] A. A. Gladkikh and G. G. Malinetskii, *Study of dynamical systems from the viewpoint of complexity and computational capabilities*, Differential Equations, 52 (2016), pp. 897–905, https://doi.org/10.1134/s0012266116070090.

[16] N. Guglielmi and C. Lubich, *Differential equations for roaming pseudospectra: Paths to extremal points and boundary tracking*, SIAM Journal on Numerical Analysis, 49 (2011), pp. 1194–1209, https://doi.org/10.1137/100817851.

[17] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in 2016 IEEE

Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, June 2016, https://doi.org/10.1109/cvpr.2016.90.

[18] D. Himmelblau, *Applied nonlinear programming*, McGraw-Hill, 1972.

[19] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, *Data-driven model reduction and transfer operator approximation*, Journal of Nonlinear Science, 28 (2018), pp. 985–1010, https://doi.org/10.1007/s00332-017-9437-7.

[20] B. O. Koopman and J. von Neumann, *Hamiltonian systems and transformation in Hilbert space*, Proceedings of the National Academy of Sciences of the United States of America, (1932), p. 315.

[21] M. Korda, M. Putinar, and I. Mezić, *Data-driven spectral analysis of the koopman operator*, Applied and Computational Harmonic Analysis, (2018), https://doi.org/10.1016/j.acha.2018.08.002.

[22] A. Lasota and M. C. Mackey, *Chaos, Fractals, and Noise*, Springer New York, 1994, https://doi.org/10.1007/978-1-4612-4286-4.

[23] B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, and R. Skeel, eds., *New Algorithms for Macromolecular Simulation*, Springer Berlin Heidelberg, 2005, https://www.ebook.de/de/product/3996213/new_algorithms_for_macromolecular_simulation.html.

[24] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, *Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 27 (2017), p. 103111, https://doi.org/10.1063/1.4993854.

[25] K. Maeda and S. Tsujimoto, *A generalized eigenvalue algorithm for tridiagonal matrix pencils based on a nonautonomous discrete integrable system*, Journal of Computational and Applied Mathematics, 300 (2016), pp. 134–154, https://doi.org/10.1016/j.cam.2015.12.032.

[26] A. Mauroy and J. Goncalves, *Koopman-based lifting techniques for nonlinear systems identification*, IEEE Transactions on Automatic Control, (2019), https://doi.org/10.1109/tac.2019.2941433, https://arxiv.org/abs/1709.02003v4.

[27] A. Mauroy, I. Mezić, and J. Moehlis, *Isostables, isochrons, and koopman spectrum for the action–angle representation of stable fixed point dynamics*, Physica D: Nonlinear Phenomena, 261 (2013), pp. 19–30, https://doi.org/10.1016/j.physd.2013.06.004.

[28] R. M. Mazo, *Brownian Motion*, Oxford University Press, Oct. 2008, https://doi.org/10.1093/acprof:oso/9780199556441.001.0001.

[29] I. Mezić, *Spectral properties of dynamical systems, model reduction and decompositions*, Nonlinear Dynamics, 41 (2005), pp. 309–325, https://doi.org/10.1007/s11071-005-2824-x.

[30] I. Mezić, *Analysis of fluid flows via spectral properties of the koopman operator*, Annual Review of Fluid Mechanics, 45 (2013), pp. 357–378, https://doi.org/10.1146/annurev-fluid-011212-140652.

[31] I. Mezic, *Koopman operator spectrum and data analysis*, arXiv, (2017), https://arxiv.org/abs/1702.07597v1.

[32] S. N. Miller, *The dynamics of newton's method on cubic polynomials*, master's thesis, 2006.

[33] Y. Miyatake, T. Sogabe, and S.-L. Zhang, *On the equivalence between SOR-type methods for linear systems and the discrete gradient methods for gradient systems*, Journal of Computational and Applied Mathematics, 342 (2018), pp. 58–69, https://doi.org/10.1016/j.cam.2018.04.013.

[34] J. V. Neumann, *Eine spektraltheorie für allgemeine operatoren eines unitären raumes.*, Mathematische Nachrichten, 4 (1950), pp. 258–281, https://doi.org/10.1002/mana.3210040124.

[35] N. Noroozi, P. Karimaghaee, A. A. Safavi, and A. Bhaya, *Real-time robust and adaptive solutions to zero finding problems with uncertainty*, in Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, IEEE, Dec. 2009, https://doi.org/10.1109/cdc.2009.5400043.

[36] F. A. Pazos, A. Bhaya, and E. Kaszkurewicz, *Design of second order neural networks as dynamical control systems that aim to minimize nonconvex scalar functions*, Neurocomputing, 97 (2012), pp. 174–191, https://doi.org/10.1016/j.neucom.2012.05.007.

[37] R. Rico-Martínez and I. Kevrekidis, *Nonlinear system identification using neural networks: dynamics and instabilities*, Elsevier Science, 1995, ch. 16, pp. 409–442.

[38] R. Rico-Martínez, K. Krischer, I. Kevrekidis, M. Kube, and J. Hudson, *Discrete- vs. continuous-time nonlinear signal processing of Cu electrodissolution data*, Chemical Engineering Communications, 118 (1992), pp. 25–48, https://doi.org/10.1080/00986449208936084.

[39] T. Sahai, G. Mathew, and A. Surana, *A chaotic dynamical system that paints and samples*, IFAC-PapersOnLine, 50 (2017), pp. 10760–10765, https://doi.org/10.1016/j.ifacol.2017.08.2278.
[40] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, Journal of Fluid Mechanics, 656 (2010), pp. 5–28, https://doi.org/10.1017/s0022112010001217.
[41] C. Schütte and S. Klus, *Towards tensor-based methods for the numerical approximation of the perron–frobenius and koopman operator*, Journal of Computational Dynamics, 3 (2016), pp. 139–161, https://doi.org/10.3934/jcd.2016007.
[42] A. M. Stuart and A. R. Humphries, *Dynamical Systems and Numerical Analysis*, Cambridge University Press, 1996.
[43] W. Su, S. Boyd, and E. Candès, *A differential equation for modeling nesterov's accelerated gradient method: Theory and insights*, in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds., Curran Associates, Inc., 2014, pp. 2510–2518.
[44] J.-Q. Sun, X.-B. Hu, and H.-W. Tam, *Short note: An integrable numerical algorithm for computing eigenvalues of a specially structured matrix*, Numerical Linear Algebra with Applications, 18 (2010), pp. 261–274, https://doi.org/10.1002/nla.754.
[45] J. v. Neumann, *Zur operatorenmethode in der klassischen mechanik*, The Annals of Mathematics, 33 (1932), p. 587, https://doi.org/10.2307/1968537.
[46] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, *A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition*, J. Nonlinear Sci., 25 (2015), pp. 1307–1346, https://doi.org/10.1007/s00332-015-9258-5.
[47] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, *A kernel-based method for data-driven koopman spectral analysis*, Journal of Computational Dynamics, 2 (2015), pp. 247–265, https://doi.org/10.3934/jcd.2015005.
[48] M. O. Williams, C. W. Rowley, I. Mezić, and I. G. Kevrekidis, *Data fusion via intrinsic dynamic variables: An application of data-driven koopman spectral analysis*, EPL (Europhysics Letters), 109 (2015), https://doi.org/10.1209/0295-5075/109/40007.

Appendix A. Computational experiments with partial information. In most applications, the data will only partially cover the state space of an algorithm. In this section, we explore the behavior of the approximated operator spectrum on such partial domains. The theoretical underpinning of this experiment is provided by the concept of "open eigenfunctions" [31], i.e. eigenfunctions defined only on parts of the state space. Not all the results from the computational experiments can be readily explained by the current theory, however, as we will detail below.

We study discrete gradient descent on the function

$$f(x_1, x_2) = x_1^4 - x_1^2 + x_1/4 + x_2^2.$$

Figure 10 illustrates the function values on $[-1, 1]^2$ through contour lines (color), and the sample points used for the approximation of the operator (red area). When using gradient descent to minimize f, we obtain a (discrete) dynamical system with two attracting steady states, and one saddle point in between them (see figure 10). The spectrum obtained by EDMD for the given sample domain (red) is shown on the right: all eigenvalues are inside the unit disk, indicating that the system behavior is purely attracting. This result is reasonable, because in the limit of infinite applications of the Koopman operator, all observable functions $g \in \mathcal{F}$ of the data set can be expressed by two piecewise constant eigenfunctions $\phi_1^{(1)}, \phi_1^{(2)}$

associated to the eigenvalue $\lambda = 1$:

$$\lim_{n\to\infty} \mathcal{K}^n g = a_1 \phi_1^{(1)} + a_2 \phi_1^{(2)};\ a_1, a_2 \in \mathbb{C},\ g \in \mathcal{F}, \tag{A.1}$$

$$\phi_1^{(1)}(x) = \begin{cases} c_1^{(1)} & \text{if } x_1 < x_1^{(s)} \\ c_2^{(1)} & \text{if } x_1 \geq x_1^{(s)} \end{cases}, \quad \phi_1^{(2)}(x) = \begin{cases} c_1^{(2)} & \text{if } x_1 < x_1^{(s)} \\ c_2^{(2)} & \text{if } x_1 \geq x_1^{(s)} \end{cases}, \tag{A.2}$$

where $x_1^{(s)}$ is the x_1 coordinate of the saddle point between the two steady states. The constants a_1, a_2 are determined by the value of g on the two attracting steady states and the four constants $c_1^{(1)}, c_1^{(2)}, c_2^{(1)}, c_2^{(2)} \in \mathbb{C}$ associated to the two eigenfunctions. Equation (A.1) illustrates that, in the function space $\mathcal{F}$, there is a four-dimensional (real and imaginary parts of two complex numbers a_1, a_2), attracting subspace that is the limit set of the system $g_{n+1} = \mathcal{K} g_n$.

If we shift the domain to the right (figure 11, C), the left attracting steady state leaves the sampling domain. This causes trajectories starting in the sampling domain to leave it after some time, which typically indicates a repelling set—explaining the numerically obtained eigenvalues outside the unit disk in parts A and C of figure 11. Note that since many trajectories leave the domain after a finite number N of iterations (depending on the initial state x_n), the dynamical system is not defined for iteration numbers $n > N$. For points on the boundary of the domain, $N = 0$, so the flow map at these points is only defined for iterations backwards in time. Since there is a saddle inside the domain, the system is also not defined for some states $x \in X$ on the boundary that would leave the domain backwards in time (iteration numbers $n < 0$). This poses problems with the definition of the Koopman operator family $\{\mathcal{K}^n\}$, since it acts on functions defined on all of X—and for all $n \neq 0$, some elements leave X, so that the flow map on the entire set X is not defined for $n \neq 0$. If all vectors on the boundary of the data domain pointed inward (outward), the flow would exist for all forward (backward) time, and the problem would be related to inflowing (overflowing) invariant manifolds [12]. Eigenvalues with a real part larger than one indicate unstable behavior, indicating the existence of unstable nodes or saddles in the data set.

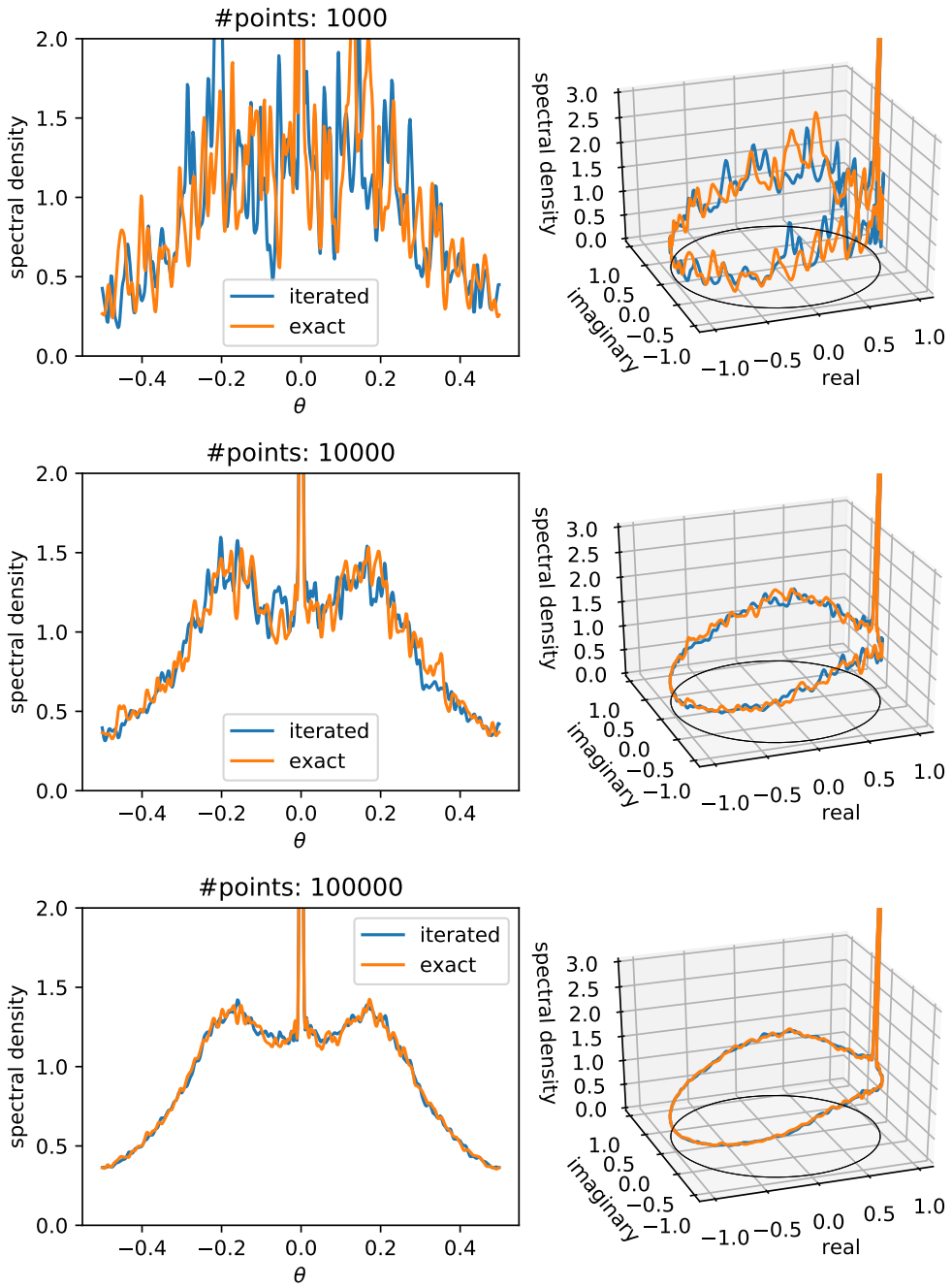


Figure 7. *The spectral density on the unit circle $\mathbb{T}$, associated to the observable $g(z) = 1 + \exp(2\pi i z)$, is approximated with single trajectories of Newton's method with length 10^3, 10^4, and 10^5 (rows) starting at $z_0 = 1/2 + 0i$. The point spectrum with eigenvalue at $1 + 0i$ associated to the constant function and the continuous spectrum on $\mathbb{T}/\{1\}$ can be distinguished. The blue curves are obtained from iterating the map N_f with 16 digits of accuracy, the orange curves by using the exact formula and 35,000 digits of accuracy.*

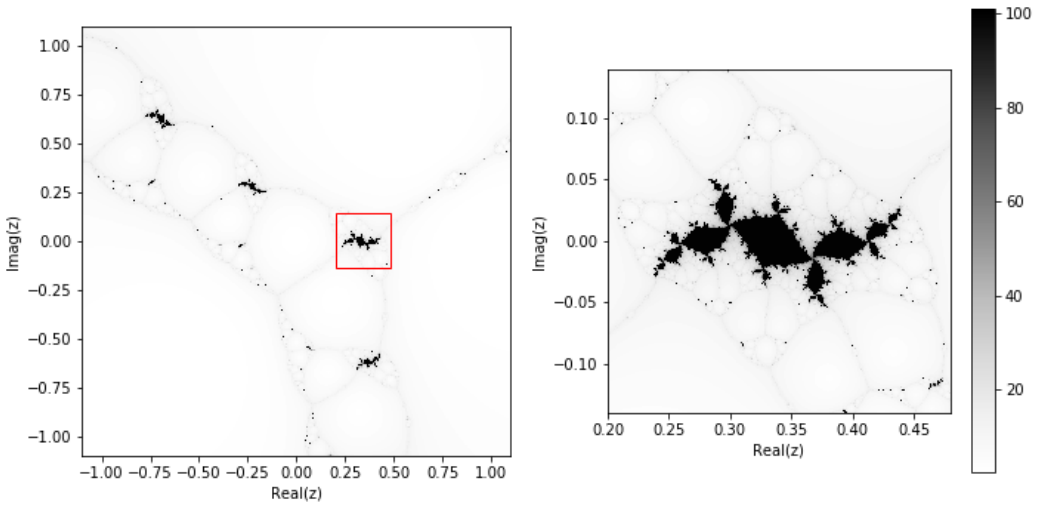


Figure 8. *Number of iterations n (colorbar) until convergence up to $|N_f(z^*) - z^*| < 0.01$ or $n \geq 100$, for a cubic polynomial function $f(z) = (z + w)(z - w)(z - 1)$ with $w = .589 + .605j$. The fractal structure of the number of iterates is visible in both panels. The right panel shows a zoomed in version of the left (marked with a box on the left).*

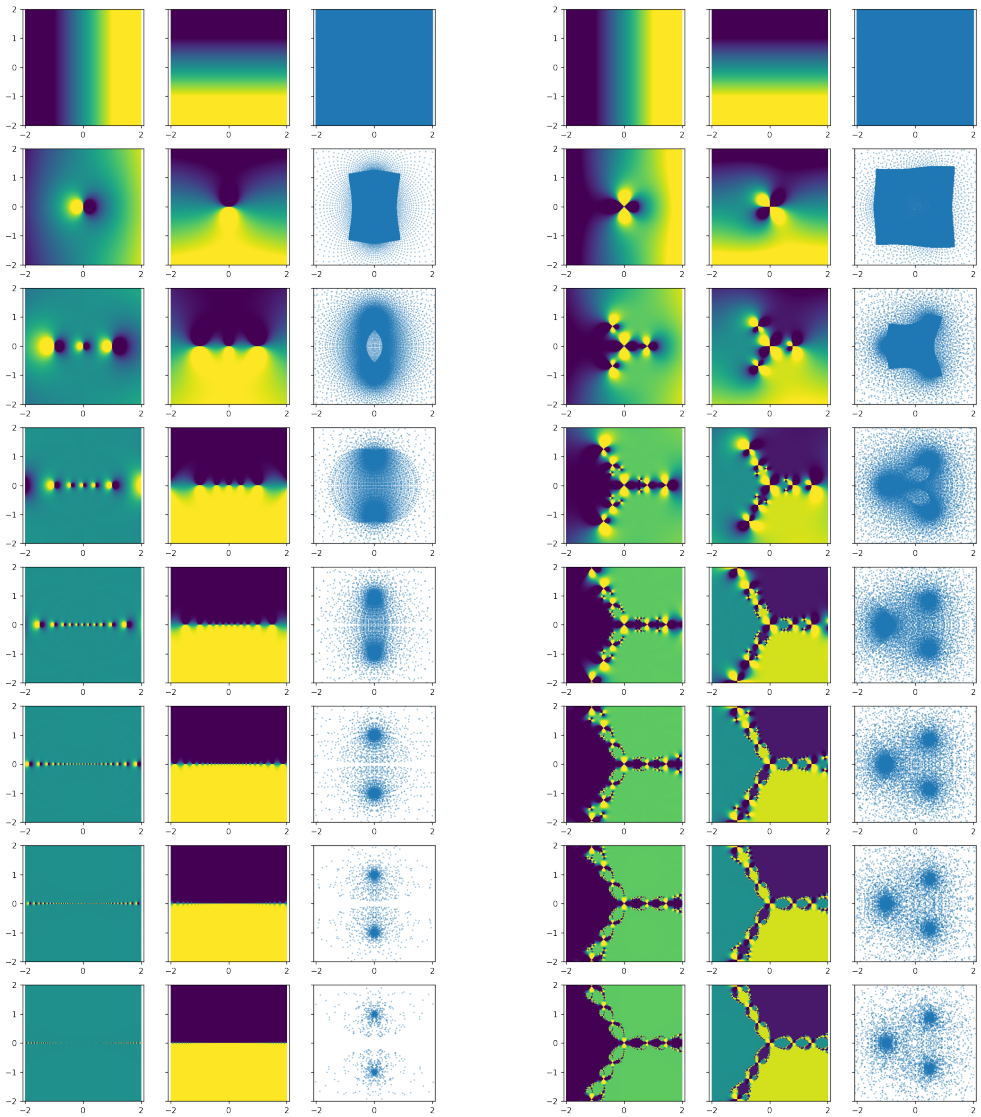


Figure 9. *Iterations on the complex plane for a polynomial function of degree two (left) and degree three (right). The eigenfunctions at eigenvalue 1 are limits of the process. On the left, the eigenfunction has two separate values for the two basins of attraction. On the right, the eigenfunction has a persistent fractal structure.*

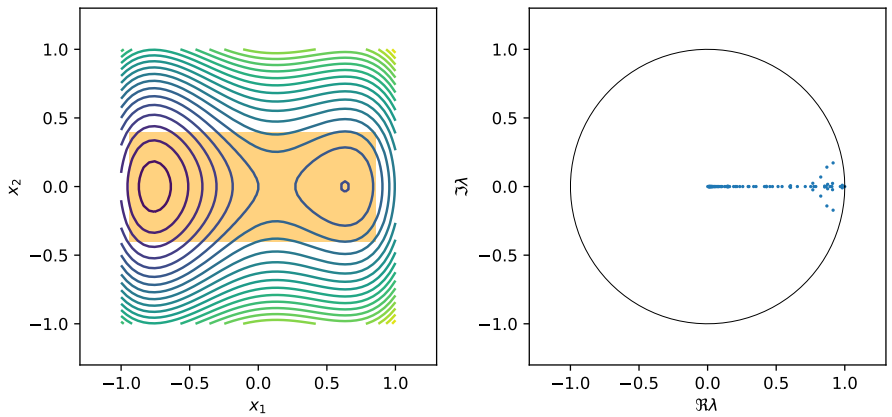


Figure 10. *Partial sampling (orange rectangle) of a two-dimensional domain. The objective function is shown as a countour plot. The two attracting steady states are both inside the rectangle, and the spectrum is fully inside the unit disk (right plot).*

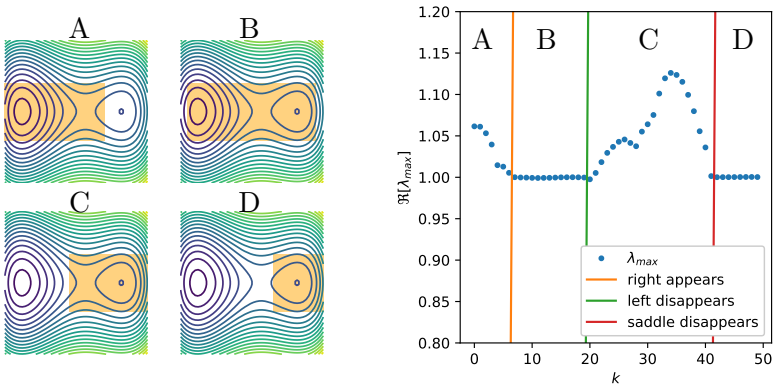


Figure 11. *At the beginning, the right steady state is not inside the data domain (orange rectangle, A). When both steady states are contained in the data, the maximum of the real part of all eigenvalues is close to one (B). As the available data set moves to the right, the left steady state is no longer available to the approximation (C). When the rectangle is moved even further (D), the saddle between the two steady states also disappears and only one, attracting steady state is contained in the data.*