

Algorithmic (Semi-)Conjugacy via Koopman Operator Theory

William T. Redman^{1,2}, Maria Fonoberova², Ryan Mohr², Ioannis G. Kevrekidis³, and Igor Mezić^{2,4}

Abstract—Iterative algorithms are of utmost importance in decision and control. With an ever growing number of algorithms being developed, distributed, and proprietaryized, there is a similarly growing need for methods that can provide classification and comparison. By viewing iterative algorithms as discrete-time dynamical systems, we leverage Koopman operator theory to identify (semi-)conjugacies between algorithms using their spectral properties. This provides a general framework with which to classify and compare algorithms.

I. INTRODUCTION

For many problems that are commonly faced in control and decision settings, a variety of numerical algorithms exist to find approximate solutions. For instance, ordinary differential equations can be solved with either the forward or the backward Euler method. Roots to polynomial functions can be found with either the Newton-Raphson or Secant method. Deep neural networks can be optimized with stochastic gradient descent using either a fixed or an adaptive learning rate. In each case, differences in numerical stability, usage of computational resources, and speed, among other factors, must be taken into consideration, in order to decide which method to employ.

A well known result within the field of numerical analysis is that many of these seemingly different algorithms are, in fact, equivalent. That is, the output of one algorithm can be exactly mapped to the output of another. In its simplest form, such equivalency between algorithms involves the same number of variables and operations, but different values of free parameters, making an appropriate choice of initial conditions lead to the same output. A sufficiently straightforward example of such an equivalence could be identified by looking at the underlying equations. However, in more subtle scenarios, two algorithms can be equivalent, but take on different looking forms, making an identification challenging. The ability to recognize equivalency is important in assessing the novelty of proposed algorithms [1]. Additionally, classifying algorithms by their equivalent counterparts (i.e. defining equivalence classes) provides a way in which to better analyze them.

Recent work to automatically detect equivalencies has taken a control theoretic perspective [1]. By viewing iterative algorithms as linear control systems [2], and evaluating their corresponding transfer functions, it was possible to develop

an analytic method that could directly identify equivalent algorithms. While broadly successful, such an approach has several limitations. First, it was only able to identify linear mappings between algorithms, restricting the class of equivalencies that could be identified. Second, several different definitions of equivalence had to be introduced in order to capture the equivalencies of the different algorithms that were studied. Third, it was necessary to have access to the equations underlying the iterative algorithms. In cases where the algorithms are proprietary, this may not be available. Lastly, the control theoretic framework adds an additional layer of complexity that is not strictly necessary.

We study the equivalence of algorithms from a related, dynamical systems theoretic perspective. Dynamical systems have a long history of inter-relatedness with algorithms [3]–[5], and recent developments in Koopman operator theory [6]–[9] have been used to optimize and analyze algorithms [10], especially within the field of machine learning [11]–[16]. Here, we show that the different types of equivalence established by Zhao et al. (2021) [1] naturally fall under the idea of (semi-)conjugacy, which can be identified from spectral properties of the Koopman operator. Using several of the illustrative examples from Zhao et al. (2021), we find that different algorithms that are equivalent by: 1) linear, invertible mappings; 2) linear, embedded mappings; 3) nonlinear, invertible mappings; and 4) shifts of output values, can all be detected by signatures in their Koopman spectra. Taken together, our work provides evidence that Koopman operator theory is a general approach for studying algorithmic equivalence.

The remainder of the paper is organized as follows. We begin by discussing how iterative algorithms can be viewed as discrete-time dynamical systems (Sec. II-A) and review basics of Koopman operator theory (Sec. II-B). In particular, we highlight the importance of the Koopman mode decomposition. We proceed to define the notions of (semi-)conjugacy, and discuss results that allow the Koopman spectrum to be used in identifying (semi-)conjugate systems (Sec. II-C). In Sec. III, we show that the theory outlined in Sec. II can be used to identify equivalencies in various example algorithms. We conclude in Sec. IV.

II. KOOPMAN OPERATOR THEORY

A. Iterative algorithms as dynamical systems

The idea central to our framework is that iterative algorithms can be viewed as discrete-time dynamical systems. To see this, consider an algorithm, $A : X \rightarrow X$, whose state space is $X \subseteq \mathbb{R}^d$, for some $d \in \mathbb{N}$. Starting from an initial

¹W.T.R. is with the Interdepartmental Graduate Program in Dynamical Neuroscience, University of California, Santa Barbara.

²W.T.R., M.F., R.M., and I.M. are with AIMdyn Inc., Santa Barbara.

³I.G.K. is with the Departments of Chemical and Biomolecular Engineering, and of Mathematics and Statistics, The Johns Hopkins University.

⁴I.M. is with the Departments of Mechanical Engineering and of Mathematics, University of California, Santa Barbara.

state $x_0 \in X$, each iterative application of A evolves the state-space input by

$$x_{k+1} = Ax_k = A^{k+1}x_0. \quad (1)$$

Comparing Eq. 1 to the classical perspective of dynamical systems, we see that A acts as a dynamical map, with the number of iterations, $k \in \mathbb{N}$, acting as “time”. The resulting sequence, $\{x_k\}_{k=1}^N$, is a trajectory through state-space. The existence of geometric objects that are studied in dynamical systems theory, such fixed points, limit cycles, quasi-periodic attractors, etc. depends on the precise nature of A and X . Indeed, the same A , applied to different domains of X , can have different properties. However, in order for an algorithm to be of general practical utility, it is necessary for it to converge, and converge in a finite number of iterations. Therefore, for the algorithms we study in this paper, we will assume that there is a large region of state space, $M \subseteq X$, where A evolves any initial condition $x_0 \in M$ to within ε of the fixed point x^* , in at most $N_{\max} \in \mathbb{N}$ iterations.

However, the key idea in this paper, the comparison of dynamical systems based on the spectrum of the associated Koopman operator, is applicable to a much broader set of systems with point, limit cycle, toroidal, and even chaotic attractors [17], [18].

B. Koopman mode decomposition

While the discrete-time dynamical systems view of algorithms outlined in Sec. II-A motivates the use of dynamical systems theory, the standard tools are difficult to leverage when the algorithms are nonlinear or when the equivalence between algorithms is nonlinear. Additionally, they are impossible to use when the equations underlying the algorithms are not known. Therefore, the course of action is not immediately apparent. Koopman operator theory, a data-driven dynamical systems theory [6]–[9] that is intimately related to the geometrical objects present in the classical theory [8], [18]–[21], offers a way forward.

The central object of interest in Koopman operator theory is the Koopman operator, U , an infinite dimensional linear operator that describes the time evolution of observables (i.e. functions of the underlying state space variables) that live in the functional space, \mathcal{F} . That is, after $t > 0$ amount of time, which can be continuous or discrete, the value of the observable $f \in \mathcal{F}$, which can be a scalar or a vector valued function, is given by

$$U^t f(x_0) = f [T^t(x_0)], \quad (2)$$

where T is the dynamical map evolving the system and $x_0 \in M$ is the initial condition in state space $M \subseteq X \in \mathbb{R}^d$. For the remainder of the paper, it will be assumed that M is of finite dimension and that \mathcal{F} is the suitably chosen space of functions in which the spectral expansion exists [18].

The action of the Koopman operator on the observable f can be decomposed as

$$Uf(x_0) = \sum_{r=0}^{\infty} \lambda_r \phi_r(x_0) v_r, \quad (3)$$

where the ϕ_r are eigenfunctions of U , with $\lambda_r \in \mathbb{C}$ as their eigenvalues and v_r as their Koopman modes [8]. For systems with chaotic or shear dynamics, there is an additional term in Eq. 3 arising from the continuous part of the spectrum [8]. As the algorithms we study are not expected to have such dynamics, for the remainder of this paper it will be assumed that the dynamical systems we are considering are such that the Koopman operator only has a point spectrum.

Spectrally decomposing the action of the Koopman operator is powerful because, for a discrete-time dynamical system, the value of f at time step, $k \in \mathbb{N}$, is given simply by

$$f [T^k(x_0)] = U^k f(x_0) = \sum_{r=0}^{\infty} \lambda_r^k \phi_r(x_0) v_r. \quad (4)$$

From Eq. 4, we see that the dynamics of the system in the directions v_r , scaled by $\phi_r(x_0)$, are given by the magnitude of the corresponding λ_r . Assuming that $|\lambda_r| \leq 1$ for all r , finding the long time behavior of f amounts to considering only the $\phi_r(x_0) v_r$ whose $|\lambda_r| \approx 1$.

While the number of triplets (v_r, λ_r, ϕ_r) needed to fully capture the action of U is, in principle, infinite, in many applied settings it has been found that a finite number, $R \in \mathbb{N}$, of them allows for a good approximation [9]. Namely, for a generic n -dimensional system that is stable in the basin of attraction to a fixed point x^* , there is the set of principal eigenvalues $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_R)$, with real part less than or equal to 1 and such that $|\lambda_1| < |\lambda_2| < \dots < |\lambda_R|$. All the other eigenvalues are obtained by

$$\lambda^{\mathbf{k}} = \lambda_1^{k_1} \cdot \lambda_2^{k_2} \dots \cdot \lambda_R^{k_R}, \quad (5)$$

where $\mathbf{k} = (k_1, \dots, k_R) \in \mathbb{N}^R$. Thus, we can often select a finite number, R , of eigenvalues that dominate the spectral expansion after a sufficient number of iterates, as their magnitudes are closer to 1. That is,

$$U^k f(x_0) \approx \sum_{r=0}^{R-1} \lambda_r^k \phi_r(x_0) v_r. \quad (6)$$

Many numerical methods exist to compute the Koopman mode decomposition, which have allowed it to be applied to complex problems across a wide range of scientific domains. The most popular method is dynamic mode decomposition (DMD) [22]–[24], and its variants [21], [25], [26]. Here we make use of DMD and Extended-DMD [26] because of their ubiquity.

C. (Semi-)Conjugacy

As discussed in the previous section, in a suitably chosen functional space, all eigenvalues of the Koopman operator are obtained from the R principal ones. The key idea in comparing algorithms then becomes one of comparing principal eigenvalues. This is justified by utilizing the notion of (semi-)conjugacy.

Definition 1: Let $T : M \rightarrow M$ and $S : N \rightarrow N$ be two discrete-time dynamical systems on sets $M \subset \mathbb{R}^m$ and $N \subset \mathbb{R}^n$, with the associated Koopman operators U^T and

U^S . They are conjugate provided $m = n$ and there exists a smooth invertible mapping $h : M \rightarrow K$, such that $h \circ T = S \circ h$. In other words, $U^T h = S \circ h$. If $n < m$ and h is smooth but not invertible, then the mapping is called semi-conjugate.

Clearly, if T and S are semi-conjugate, then $h(x^*)$ is a fixed point of S if and only if x^* is a fixed point of T . We have the following Lemma.

Lemma 1: If T and S are conjugate and T is stable in the basin of attraction to x^* , then they have the same principal eigenvalues. If T and S are semi-conjugate, then the set of principal eigenvalues of U^S is a subset of the set of principal eigenvalues of U^T .

Proof: Let λ, ϕ be an eigenvalue, eigenfunction pair of U^S . Then

$$\begin{aligned} \lambda \phi[h(x)] &= U^S \phi[h(x)] = \phi(S[h(x)]) = \dots \\ &= \phi(h[T(x)]) = U^T \phi(h). \end{aligned} \quad (7)$$

Thus, λ is an eigenvalue of U^T , associated with the eigenfunction $\phi \circ h$. The converse to Lemma 1, that is, if U^T and U^S have discrete spectra that are equivalent, then T and S are conjugate, can also be shown to be true [17].

III. RESULTS

Having developed a dynamical systems framework for studying algorithmic equivalence, via comparing the spectra of the Koopman operators associated with the algorithms being applied to a given problem (Sec. II), we tested whether standard numerical methods could properly identify instances of (semi-)conjugacy. To do this, we made use of several of the illustrative examples that were presented by Zhao et al. (2021), examining equivalencies defined by various types of mappings (i.e. linear/nonlinear, invertible/embedded) [1]. In each case, we find that implementations of Koopman mode decomposition can identify the underlying conjugacy, and thus, provide a general framework for probing algorithmic equivalency.

A. Equivalence by linear, invertible mappings

We begin by examining the toy Algorithms 1 and 2 [1]. Each algorithm's behavior is determined by the choice of the function f , as ∇f is an operation that occurs in both algorithms. For any f , there exists an invertible, linear mapping between the two algorithms' outputs, such that the two sequences are equivalent, assuming the initial conditions have been properly set. Namely,

$$\begin{aligned} \xi_1^k &= 2x_1^k - x_2^k \\ \xi_2^k &= -x_1^k + x_2^k \end{aligned} \quad (8)$$

defines a mapping between the output of Algorithm 1 (x_i^k) and the output of Algorithm 2 (ξ_i^k). In this sense, the two algorithms are "oracle equivalent" [1], where ∇f is considered an oracle.

We examine the Koopman spectra of the two algorithms applied to pairs of initial conditions, related via Eq. 8, for different choices of the function f . Examples of $f(x) = x^2$ and $f(x) = -\cos(x)$ are shown in Fig. 1.

Algorithm 1

```

for  $k = 0, 1, 2, \dots, K$  do
   $x_1^{k+1} = 2x_1^k - x_2^k - \frac{1}{10} \nabla f(2x_1^k - x_2^k)$ 
   $x_2^{k+1} = x_1^k$ 
end for

```

Algorithm 2

```

for  $k = 0, 1, 2, \dots, K$  do
   $\xi_1^{k+1} = \xi_1^k - \xi_2^k - \frac{1}{5} \nabla f(\xi_1^k)$ 
   $\xi_2^{k+1} = \xi_2^k + \frac{1}{10} \nabla f(\xi_1^k)$ 
end for

```

In both cases, we find that the Koopman spectra are nearly exactly overlapping. Thus, as predicted from the theory discussed in Sec. II, conjugacy via linear, invertible mapping can be identified by the two algorithms having the same Koopman eigenvalues.

While Algorithms 1 and 2 are conjugate by Eq. 8 for both choices of $f(x)$ we considered, the nature of this conjugacy differs. When $f(x) = x^2$, the two algorithms are linear, as $\nabla f = 2x$, and the matrices governing their dynamics have the same eigenvalues, corresponding to the same single attractor. Therefore, Algorithms 1 and 2 are *globally conjugate*, as any two initial conditions will evolve, with the same dynamics, to $(0, 0)$.

In contrast, when $f(x) = -\cos(x)$, the algorithms are no longer linear, as $\nabla f = \sin(x)$, and there exist multiple attractors. For instance, when $2x_1^0 - x_2^0$ is small, Algorithm 1 converges to $(0, 0)$, but when $2x_1^0 - x_2^0$ is large, x_1^k and x_2^k grow approximately linearly. A similar transition occurs for Algorithm 2. Therefore, Algorithms 1 and 2 are

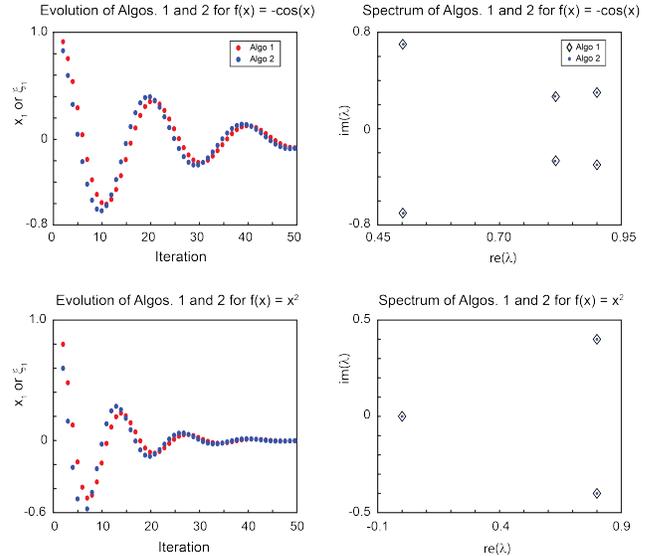


Fig. 1. **Conjugacy of Algorithms 1 and 2.** Left column, trajectory of state variables x_1 and ξ_1 . Right column, corresponding Koopman spectra. Top row, $f = -\cos(x)$. Bottom row, $f = x^2$. The nearly identical spectra correctly illustrates that the two algorithms are conjugate.

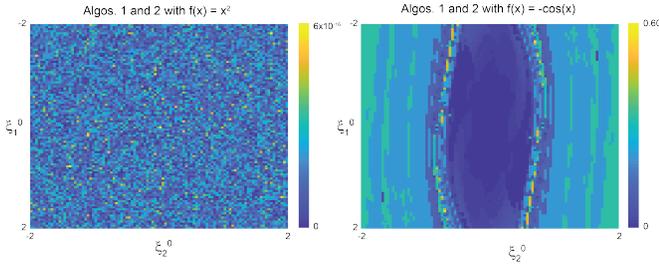


Fig. 2. **Global and local conjugacy.** Wasserstein distance of the Koopman spectra when the initial condition for Algorithm 1 is fixed at $x_1^0 = x_2^0 = 0.1$ and the initial condition of Algorithm 2, ξ_1^0 and ξ_2^0 , is sampled over the range $[-2, 2] \times [-2, 2]$. Left, the Koopman spectra are nearly identical for every pair of initial conditions when $f(x) = x^2$. Right, the Koopman spectra can be similar or different when $f(x) = -\cos(x)$, depending on the initial conditions.

locally conjugate, as initializing them differently can lead to trajectories in different dynamical regimes.

To illustrate this, and demonstrate that Koopman operator theory can identify such differences in the behavior of the conjugacies, we fix $x_1^0 = x_2^0 = 0.1$ (making $2x_1^0 - x_2^0 = 0$ small), and vary (ξ_1^0, ξ_2^0) over the interval $[-2, 2] \times [-2, 2]$. For each pair of (ξ_1^0, ξ_2^0) , we evolve Algorithm 2, approximate the Koopman spectrum, and compute the Wasserstein distance between it and the spectrum found when applying Algorithm 1 to $(x_1^0, x_2^0) = (0.1, 0.1)$. The Wasserstein distance is a metric originally developed in the context of optimal transport, and provides a sense of how far apart two distributions are. As expected, when $f(x) = x^2$, all (ξ_1^0, ξ_2^0) we evaluated led to the same dynamics of Algorithm 2 as compared to Algorithm 1, with Koopman eigenvalues that were separated by less than 10^{-15} (Fig. 2, left). However, when $f(x) = -\cos(x)$, there is a region where the Wasserstein distance increases, as the dynamics of Algorithm 2 transition to a different attractor (Fig. 2, right).

These results highlight the fact that our Koopman framework can provide a view on the relationship between two algorithms that is broader than the binary classification of whether they are conjugate or not. Given that the nature of equivalence is an important factor to take into account when classifying and comparing algorithms, this is a useful feature.

B. Equivalence by linear, embedded mappings

We next consider Algorithms 3 and 4 [1]. There exists a linear, embedded mapping between their outputs. Namely,

$$\xi^k = -x_1^k + 2x_2^k \quad (9)$$

defines a mapping of the two-dimensional output of Algorithm 3 (x_i^k) to the one-dimensional output of Algorithm 4 (ξ^k).

From the theory developed in Sec. II, we can recognize Eq. 9 as describing a semi-conjugacy. We therefore expect that the Koopman spectrum associated with the smaller system (Algorithm 4) should be a subset of the spectrum associated with the larger system (Algorithm 3). Indeed, we find this to be the case for $f(x) = x^2$ and $f(x) = -\cos(x)$ (Fig.

Algorithm 3

```

for  $k = 0, 1, 2, \dots, K$  do
   $x_1^{k+1} = 3x_1^k - 2x_2^k + \frac{1}{5}\nabla f(-x_1^k + 2x_2^k)$ 
   $x_2^{k+1} = x_1^k$ 
end for

```

Algorithm 4

```

for  $k = 0, 1, 2, \dots, K$  do
   $\xi^{k+1} = \xi^k - \frac{1}{5}\nabla f(\xi^k)$ 
end for

```

3). In both cases, the spectra have the same eigenvalues corresponding to decaying modes (i.e. $|\lambda| < 1$). However, the eigenvalue corresponding to the exponentially growing mode (i.e. $|\lambda| > 1$) is only present in the spectrum of Algorithm 3, which indeed matches the unbounded growth of x_1^k and x_2^k .

C. Equivalence by nonlinear, invertible mappings

We next tackle an equivalence noted by Zhao et al. (2021) that is given by a nonlinear, invertible mapping. This is a non-trivial problem to identify, and cannot be done within a linear control framework [1], [2].

Algorithm 5 is equivalent to Algorithm 4 via,

$$x^k = \exp(\xi^k). \quad (10)$$

From the theory discussed in Sec. II, we expect that the Koopman spectra of such conjugate systems should be equivalent. We indeed find this to be true (Fig. 4).

Algorithm 5

```

for  $k = 0, 1, 2, \dots, K$  do
   $x^{k+1} = x^k \exp(-\frac{1}{5}\nabla f(\log x^k))$ 
end for

```

We note that, since we used a simple numerical scheme to compute the Koopman mode decomposition on a small set of data points, we do not see a perfect overlap of the spectra. In particular, the smallest eigenvalues differ the most between the two. Given the plethora of robust, precise numerical methods, we imagine it would be possible to

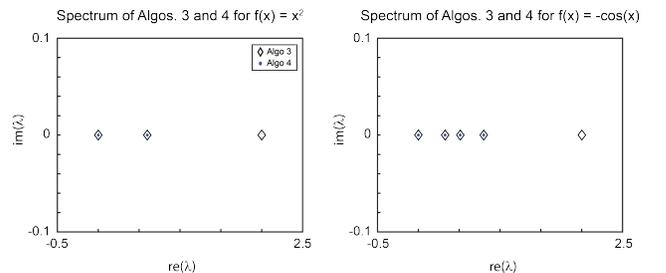


Fig. 3. **Semi-conjugacy of Algorithms 3 and 4.** The spectrum of Algorithm 4 is a subset of the spectrum of Algorithm 3, correctly implying that the two systems are semi-conjugate, for both $f(x) = x^2$ (left) and $f(x) = -\cos(x)$.

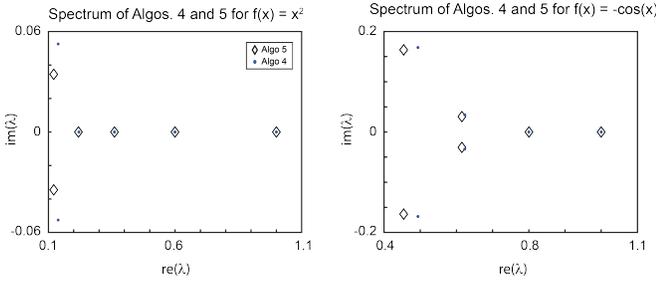


Fig. 4. **Conjugacy of Algorithms 4 and 5.** The nearly overlapping spectra, up to some differences in the smallest eigenvalues, correctly implies that Algorithms 4 and 5 are conjugate.

achieve better overlap. Regardless, the fact that the largest eigenvalues overlap suggests a clear connection between the two algorithms. Given that any numerical approach will find some non-zero difference, there will necessarily have to be some discretion practiced by the user.

D. Equivalence by shifting

Because algorithms can perform the same sequence of operations, but do so in differing orders, Zhao et al. (2021) considered “shift equivalence”. By permuting the transfer operators associated with each algorithm, it became possible to identify shift equivalent algorithms under the linear control framework [1]. However, this required some additional theory. We reasoned that our Koopman operator theoretic approach may be able to identify shift equivalent algorithms, without any new machinery.

To explore this, we considered Algorithms 6 and 7 [1]. The two make use of the proximal operator, prox_f . To evaluate this operation numerically, we make use of the UNLocBoX package [27]. The two algorithms are equivalent via the shift

$$\begin{aligned} \xi_1^k &= x_3^k \\ \xi_2^k &= x_1^{k+1}, \end{aligned} \quad (11)$$

which relates the initial conditions of Algorithm 7 (ξ_i^0) to iterates of Algorithm 6 (x_i^k), for $k \geq 0$. After performing such a shift, the two algorithms make the same calls on the proximal operators.

Algorithm 6

for $k = 0, 1, 2, \dots, K$ **do**
 $x_1^{k+1} = \text{prox}_f(x_3^k)$
 $x_2^{k+1} = \text{prox}_g(2x_1^{k+1} - x_3^k)$
 $x_3^{k+1} = x_3^k + x_2^{k+1} - x_1^{k+1}$
end for

Looking at the sequence of outputs, x_1^k and ξ_1^k , we see that indeed, ignoring x_1^0 , the two have the same dynamics, for different choices of f and g (Fig. 5 left column). Therefore, we expect that the Koopman eigenvalues will be similar between the two shifted systems. Computing the Koopman spectra of Algorithms 6 and 7, we confirm that it is indeed

Algorithm 7

for $k = 0, 1, 2, \dots, K$ **do**
 $\xi_1^{k+1} = \text{prox}_g(-\xi_1^k + 2\xi_2^k) + \xi_1^k - \xi_2^k$
 $\xi_2^{k+1} = \text{prox}_f(\xi_1^{k+1})$
end for

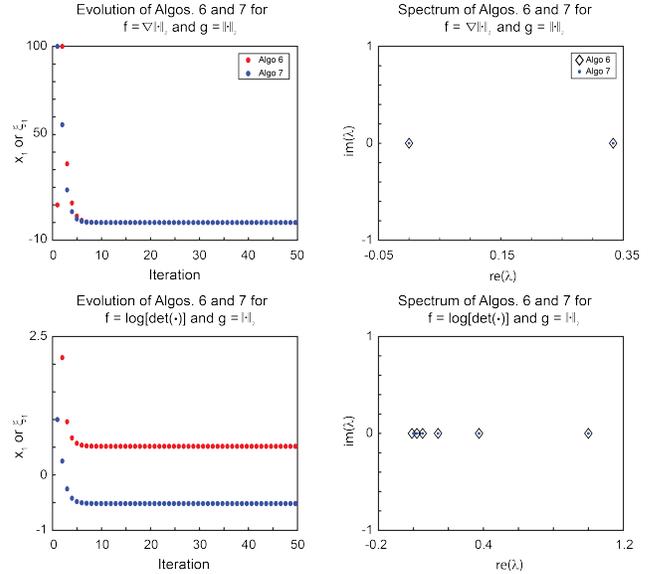


Fig. 5. **Shift equivalence of Algorithms 6 and 7.** Left column, trajectories of x_1^k and ξ_1^k for choices of $f = \nabla \|\cdot\|_2$ and $g = \|\cdot\|_2$ (top row) and $f = \log|\det(\cdot)|$ and $g = \|\cdot\|_2$ (bottom row). Right column, corresponding Koopman spectra. In both cases, the overlapping spectra correctly identify the “shift” equivalence [1].

possible to identify this kind of equivalence with no changes to our framework (Fig. 5, right column), as the Koopman spectra are largely overlapping.

Note that, as long as a sufficient amount of data snapshots are present, we can expect the two Koopman spectra to converge [28], [29]. An insufficient amount of data could, however, lead to differing decompositions. This could be overcome by using methods designed for the sparse data regime [30], [31].

IV. DISCUSSION

By viewing iterative algorithms applied to a given problem as discrete-time dynamical systems [10], we developed a framework for identifying equivalent algorithms via the spectra of the associated Koopman operators. The key to this approach relies on the fact that two dissipative systems that are conjugate have the same Koopman eigenvalues [18]. Similarly, two dissipative systems that are semi-conjugate have Koopman eigenvalues, where one is a subset of the other.

That we were able to make comparisons between dynamical systems in an applied setting using these mathematical results illustrates the wide implications that the rigorous theoretical development of Koopman operator theory can have. Indeed, results connecting Koopman spectral objects

to geometric properties of state space are indispensable for gaining insight when applying Koopman tools [8], [18]–[21].

Our method for identifying algorithmic equivalence necessarily requires the computation of the Koopman eigenvalues from data, which makes it necessary to have a sufficient amount of data and an appropriate choice of numerical scheme for a good approximation. Its conclusion of equivalence can depend on the domain chosen, as in the case of algorithms that are locally conjugate. This is in contrast to the linear control based method developed by Zhao et al. (2021), which was analytic and enabled identification of equivalences from the equations defining the algorithms, independent of initial conditions [1].

However, our framework avoids several of the limitations of the linear control framework [1], [2]. First, it was possible to identify nonlinear equivalences between algorithms. Second, by considering just the Koopman eigenvalues, it was possible to identify (semi-)conjugacy, unifying the several different definitions of equivalence introduced by Zhao et al. (2021). Third, it was possible to identify equivalence without the underlying equations. As long as a sequence of outputs from the algorithms are available, the Koopman framework can be used, making it especially useful in the case of proprietary software. Fourth, it can be used to distinguish between conjugacies with different properties (such as local and global conjugacy), an important distinction when comparing algorithms. Finally, this was all possible solely by viewing algorithms as discrete-time dynamical systems. No additional considerations, such as control, were required.

Taken together, these result illustrate that Koopman operator theory is a useful framework with which to study algorithmic equivalence, as well as algorithms more generally.

ACKNOWLEDGEMENT

W.T.R. was partially supported by a UC Chancellor's Fellowship. W.T.R., M.F., R.M and I.M were partially supported by the Air Force Office of Scientific Research project FA9550-17-C-0012. I. G. K. was partially supported by the U.S. DOE (ASCR) and DARPA ATLAS programs.

REFERENCES

- [1] Shipu Zhao, Laurent Lessard, and Madeleine Udell. An automatic system to detect equivalence between iterative algorithms, 2021.
- [2] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.
- [3] A. M. Stuart and A. R. Humphries. *Dynamical Systems and Numerical ANalysis*. Cambridge University Press, 1996.
- [4] Moody T. Chu. Linear algebra algorithms as dynamical systems. *Acta Numerica*, 17:1–86, 2008.
- [5] Tuhin Sahai. Dynamical systems theory and algorithms for np-hard problems. In Oliver Junge, Oliver Schütze, Gary Froyland, Sina Ober-Blöbaum, and Kathrin Padberg-Gehle, editors, *Advances in Dynamics, Optimization and Computation*, pages 183–206, Cham, 2020. Springer International Publishing.
- [6] B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.
- [7] B. O. Koopman and J. v. Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263, 1932.
- [8] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.

- [9] Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2020/10/24 2012.
- [10] Felix Dietrich, Thomas N. Thiem, and Ioannis G. Kevrekidis. On the koopman operator of algorithms. *SIAM Journal on Applied Dynamical Systems*, 19(2):860–885, 2020.
- [11] Akshunna S. Dogra and William T. Redman. Optimizing neural networks via koopman operator theory. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2087–2097. Curran Associates, Inc., 2020.
- [12] Mauricio E. Tano, Gavin D. Portwood, and Jean C. Ragusa. Accelerating training in artificial neural networks with dynamic mode decomposition, 2020.
- [13] Iva Manojlović, Maria Fonoberova, Ryan Mohr, Aleksandr Andrejčuk, Zlatko Drmač, Yannis Kevrekidis, and Igor Mezić. Applications of koopman mode analysis to neural networks, 2020.
- [14] Ilan Naiman and Omri Azencot. A koopman approach to understanding sequence neural models, 2021.
- [15] Ryan Mohr, Maria Fonoberova, Zlatko Drmač, Iva Manojlović, and Igor Mezić. Predicting the critical number of layers for hierarchical support vector regression. *Entropy*, 23(1), 2021.
- [16] William T. Redman, Maria Fonoberova, Ryan Mohr, Ioannis G. Kevrekidis, and Igor Mezić. An operator theoretic perspective on pruning deep neural networks, 2021.
- [17] Igor Mezić. On comparison of dynamics of dissipative and finite-time systems using koopman operator methods. *IFAC-PapersOnLine*, 49(18):454–461, 2016. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016.
- [18] Igor Mezić. Spectrum of the koopman operator, spectral expansions in functional spaces, and state-space geometry. *Journal of Nonlinear Science*, 30(5):2091–2145, Oct 2020.
- [19] 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*, 261:19–30, 2013.
- [20] Yueheng Lan and Igor Mezić. Linearization in the large of nonlinear systems and koopman operator spectrum. *Physica D: Nonlinear Phenomena*, 242(1):42–53, 2013.
- [21] Hassan Arbabi and Igor Mezić. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.
- [22] Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.
- [23] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.
- [24] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.
- [25] Kevin K. Chen, Jonathan H. Tu, and Clarence W. Rowley. Variants of dynamic mode decomposition: Boundary condition, koopman, and fourier analyses. *Journal of Nonlinear Science*, 22(6):887–915, 2012.
- [26] Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, Dec 2015.
- [27] N. Perraudin, D. Shuman, G. Puy, and P. Vandergheynst. UNLocBoX A matlab convex optimization toolbox using proximal splitting methods. *ArXiv e-prints*, February 2014.
- [28] Stefan Klus, Péter Koltai, and Christof Schütte. On the numerical approximation of the perron-frobenius and koopman operator. *Journal of Computational Dynamics*, 3(1):51–79, 2016.
- [29] Milan Korda and Igor Mezić. On convergence of extended dynamic mode decomposition to the koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, Apr 2018.
- [30] Subhrajit Sinha, Umesh Vaidya, and Enoch Yeung. On computation of koopman operator from sparse data. In *2019 American Control Conference (ACC)*, pages 5519–5524, 2019.
- [31] Subhrajit Sinha, Umesh Vaidya, and Enoch Yeung. On few shot learning of dynamical systems: A koopman operator theoretic approach, 2021.