INSTITUT DE FRANCE
Académie des sciences

Comptes Rendus
Mécanique

Christophe Zhang and Enrique Zuazua

A quantitative analysis of Koopman operator methods for system identification and predictions

Published online: 2 December 2022

https://doi.org/10.5802/crmeca.138

Part of Special Issue: The scientific legacy of Roland Glowinski

Guest editors: Gregoire Allaire (CMAP, Ecole Polytechnique, Institut Polytechnique de Paris, Palaiseau, France), Jean-Michel Coron (Laboratoire Jacques-Louis Lions, Sorbonne Université) and Vivette Girault (Laboratoire Jacques-Louis Lions, Sorbonne Université)

Les Comptes Rendus. Mécanique *sont membres du Centre Mersenne pour l'édition scientifique ouverte*
www.centre-mersenne.org
e-ISSN : 1873-7234

Comptes Rendus
Mécanique
Published online: 2 December 2022
https://doi.org/10.5802/crmeca.138

The scientific legacy of Roland Glowinski / *L'héritage scientifique de Roland Glowinski*

A quantitative analysis of Koopman operator methods for system identification and predictions

Christophe Zhang *, [a] **and Enrique Zuazua** [a, b, c]

[a] Chair in Dynamics, Control and Numerics, (Alexander von Humboldt Professorship), Department of Data Science, Friedrich Alexander Universität Erlangen-Nürnberg, 91058 Erlangen, Germany

[b] Chair of Computational Mathematics, Fundación Deusto, Avenida de las Universidades 24, 48007 Bilbao, Basque Country, Spain

[c] Departamento de Mateáticas, Universidad Autónoma de Madrid, 28049 Madrid, Spain

E-mails: chzhang92@gmail.com (C. Zhang), enrique.zuazua@fau.de (E. Zuazua)

Abstract. We give convergence and cost estimates for a data-driven system identification method: given an unknown dynamical system, the aim is to recover its vector field and its flow from trajectory data. It is based on the so-called Koopman operator, which uses the well-known link between diVerential equations and linear transport equations. Data-driven methods recover specific finite-dimensional approximations of the Koopman operator, which can be understood as a transport operator. We focus on such approximations given by classical finite element spaces, which allow us to give estimates on the approximation of the Koopman operator as well as the solutions of the associated linear transport equation. These approximations are thus relevant objects to solve the system identification problem.

We then analyze the convergence of a variant of the generator Extended Dynamic Mode Decomposition (gEDMD) algorithm, one of the main algorithms developed to compute approximations of the Koopman operator from data. We find however that, when combining this algorithm with classical finite element spaces, the results are not satisfactory numerically, as the convergence of the data-driven approximation is too slow for the method to benefit from the accuracy of finite element spaces. In particular, for problems in dimension 1 it is less e Y cient than direct interpolation methods to recover the vector field. We provide some numerical examples to illustrate this last point.

Keywords. Koopman operator, System identification, Finite element spaces, Data-driven approximation, Extended dynamic mode decomposition.

Published online: 2 December 2022

* Corresponding author.

1. Introduction

1.1. *Unknown dynamics and system identification*

Many problems in applied mathematics rely on the existence of a mathematical model for the system under consideration. Given the potential complexity of systems that arise in all scientific fields, from econometrics to molecular dynamics, finding relevant and numerically tractable models for applications such as forecasting and control is a central challenge.

The traditional approach to mathematical modelling consists in deriving governing equations from fundamental laws of physics such as Newton's second law or the principle of least action, after a series of assumptions and ideal simplifications. Models derived in this manner have been thoroughly studied, giving rise to rich theories and a profound understanding of some of the phenomena they describe. But these remain ideal models, which sometimes fall short of real-life observations and applications. Indeed, as real-life systems become more complex, the relevance of such models for applications comes into question, as there is a risk that usual, convenient simplifications could lead to essential features of the dynamics being overlooked.

On the other hand, the development of data-driven techniques has brought a new perspective to this challenge (see for example [1–3]): how can one complement traditional modelling approaches with data-driven considerations? Can relevant features of the dynamics of the system be recovered from data? More generally, how can one learn a model from data, and propose a system of governing equations that allow for reliable predictions, and eVective control?

Mathematically speaking, we can cast this as follows: suppose that, for a given system, its behaviour can be relevantly modelled with an autonomous, nonlinear diVerential equation

$$\dot{x} = f(x), \quad x \in \mathbb{R}^d. \tag{1}$$

How can one learn an approximation of the vector field f from observational data, that is, samples of trajectories? Given initial conditions, can one produce predictions on the evolution of the system in a data-driven manner?

This is a form of inverse problem, which diVers from more classical inverse problems (see for example [4–6]) in that the latter rely on the a priori choice of a class of models (heat equation, polynomial diVerential equations, balance laws...) so that one has to recover parameters of the given model (diVusivity, coeYcients...).

1.2. *Regression, interpolation and operator theoretic approach*

The task of recovering a function $f : \mathbb{R}^d \to \mathbb{R}^d$ from a data sample of values $\{x_i, y_i = f(x_i)\}_{i=1}^N$ in order to make predictions lies at the heart of statistical regression analysis (linear regression being perhaps the simplest example), with examples dating back to as early as Legendre's works on planetary orbits using a least squares method [7]. More recently, a variety of supervised learning and deep learning techniques have been developed to tackle this problem, such as kernel-based methods, neural networks, or deep neural networks. We refer to [8–10] for a comprehensive overview and further references.

The core idea of these regression techniques is to find a function $\hat{f}$ belonging to a certain class of functions $\mathscr{S}$ (linear functions, polynomials, wavelets...), that minimizes the loss

$$\sum_{i=1}^{N} \| \hat{f}(x_i) - y_i \|^2, \tag{2}$$

in such a manner that $\hat{f}$ will also be a good approximation of f on new data. The first approach to solving (2) would be to find, if it exists, a function in $\mathscr{S}$ that interpolates the data, that is,

$$\hat{f} \in \mathscr{S}, \quad \sum_{i=1}^{N} \| \hat{f}(x_i) - y_i \|^2 = 0 \quad \text{i.e. } \hat{f}(x_i) = y_i = f(x_i),\ i = 1, \dots, N. \tag{3}$$

A typical example of functions used for interpolation are polynomials, as there exists a unique polynomial of degree at most $N-1$ that satisfies (3). However, although $\hat{f}$ fits the given data perfectly, it has no reason a priori to be a good approximation of f on new data. Nonetheless under some assumptions on the x_i and on f, there are some well-known estimates for interpolation by polynomials and continuous piecewise polynomial functions. We refer to [11, Chapter 3] for more details on these interpolation results.

Another approach for regression that is worth mentioning is Sparse Identification of Nonlinear Dynamics (SINDy) [12–14]. It consists in finding a good approximation of f that is sparsely represented (in order to gain computational e Y ciency) in $\mathscr{S}$. In order for this method to work, the functions in $\mathscr{S}$ must be chosen carefully. Usually a basis of ordinary nonlinear functions (such as trigonometric functions, exponentials, polynomials) is used, although a more recent weak formulation, which highlights the analogy with Galerkin methods, obtains results using a specific family of piecewise polynomials [14].

As the above example shows, regression techniques are not always designed to interpolate the data exactly, as in the first example, but rather to reach a compromise between a small loss function in (2), and an e Y cient and accurate overall approximation of the target function f, so that $\hat{f}$ can be generalized to new data. There are then two important aspects to consider: first, the choice of the family of functions $\mathscr{S}$ and its representation, which can benefit from prior knowledge on f. Second, the method used to minimize the loss, which can involve for example regularization to avoid overfitting $\hat{f}$ to the data to the detriment of generalization.

In particular, if one has the prior knowledge that f is linear, then the above problem amounts to a multivariate linear regression, and the least squares problem (2) has a well-known solution. This, among other things, has motivated an alternative approach to system identification, the so-called operator theoretic approach. Instead of treating system identification directly as a general nonlinear problem, one can recast this problem as a linear one, using the well-known connection between diVerential equations and linear transport equations. This is appealing both on a theoretical level, as it gives an elegant framework to understand some intricate properties of nonlinear dynamical systems, and on a practical level, due to the many tools available both for forecasting and control of linear systems. As we will see, with this approach system identification then becomes a linear regression, but in a potentially much larger space.

2. Koopman operators

2.1. *Definitions*

Consider a vector field $f : \mathbb{R}^d \to \mathbb{R}^d$. One can view it as the vector field of a nonlinear ordinary diVerential equation:

$$\dot{x} = f(x), \quad x \in \mathbb{R}^d, \tag{4}$$

but also as the velocity field of the linear transport equation:

$$\frac{\partial \varphi}{\partial t}(t,x) = f(x) \cdot \nabla \varphi. \tag{5}$$

The equivalence between these two points of view has been long known. Existence and uniqueness results for nonlinear diVerential equations with Sobolev vector fields were obtained by studying the corresponding transport equation [15, 16]. Conversely, the method of characteristics solves (5) using (4): if f is regular enough, it generates a flow Φ_f^t which solves (4), and the solutions of (5) are constant along the characteristics $(t, \Phi_{-f}^t(x))$, $t \geq 0$, $x \in \mathbb{R}^d$. The solution of (5) with initial condition $\varphi_0 \in L^2(\mathbb{R}^d)$ is then given by

$$\varphi(t,x) := \varphi_0(\Phi_f^t(x)). \tag{6}$$

This corresponds to the following semigroup:

$$\varphi_0 \in L^2(\mathbb{R}^d) \mapsto \varphi_0 \circ \Phi_f^t, \tag{7}$$

which is a simple composition operator with the following infinitesimal generator

$$\mathscr{K} := f(x) \cdot \nabla, \tag{8}$$

which is simply the transport operator with velocity field f, on the domain

$$\mathscr{D}(\mathscr{K}) = \{\varphi \in L^2(\mathbb{R}^d), f \cdot \nabla\varphi \in L^2(\mathbb{R}^d)\}. \tag{9}$$

In what follows, we will denote by $e^{t\mathscr{K}}$ the semigroup (7), generated by the Koopman operator $\mathscr{K}$.

The semigroup and its generator were studied and used by Koopman in [17] to study properties of fluid dynamics. They have since then been named after him: what is known as the *Koopman operator of* f is the operator $\mathscr{K}$ given by (8), and the *Koopman semigroup of* f is the semigroup $e^{t\mathscr{K}}$ given by (7). Practically speaking, for a function $\varphi \in L^2(\mathbb{R}^d)$ and a trajectory $x(t)$ of the diVerential equation (4), the Koopman semigroup describes the evolution of $t \mapsto \varphi(x(t))$.

Remark 2.1. In applications involving discrete-time dynamical systems

$$x^+ = T(x), \quad x \in \mathbb{R}^d, \tag{10}$$

the corresponding discrete-time Koopman operator is simply the composition operator

$$U: \quad g \mapsto g \circ T. \tag{11}$$

Equivalently, when dealing with continuous-time systems, the Koopman operator is sometimes defined as the bounded operator $e^{\tau\mathscr{K}}$ with the notations of (7), for some time increment $\tau > 0$.

Historically, Koopman used the linear composition operator (7) to study measure-preserving systems. It was notably instrumental in Von Neumann's proof of the mean ergodic theorem [18], and a first connection of chaotic behavior to the existence of continuous parts in the spectrum of the Koopman semigroup was noted in [19]. More recently, spectral properties of the Koopman semigroup have been shown to be related to a variety of aspects of the underlying dynamical system such as ergodicity, stability, existence of diVerent time scales, mixing and non-mixing properties (see for example [20–25]). An important feature in these works is that the spectral analysis of the Koopman operator is done in a rather geometrical spirit in combination with linearization results and conjugation techniques, focusing on local behaviour near equilibria or attractors.

2.2. *Koopman operator and data-driven modelling*

In addition to this renewed theoretical exploration, data-driven applications of Koopman operator theory have gained quite some interest in the last decade, in particular for system identification, modelling, predictions and control problems [26–28]. In these works, among others, the idea is that one can gain a deep understanding of the underlying nonlinear dynamical system, and in particular identify it, by identifying its Koopman operator, or its Koopman semigroup, or the discrete-time Koopman operator, depending on the nature of the system.

There are two potential benefits to this approach:

(1) The linearity of the Koopman operators, which oVers the possibility to describe nonlinear dynamics in a linear manner.
(2) As a consequence, the possibility to make predictions by computing solutions to the linear equations given by these operators, instead of integrating a nonlinear diVerential equation.

Numerically speaking, the focus of these methods lies on finite-dimensional reductions of the Koopman operator, as it is not possible to recover an infinite-dimensional operator from a finite amount of data.

The question then becomes: which finite-dimensional reductions are accurate and relevant for system identification and predictions? What is their dimension? How does one recover them from data? At what cost?

- **Exact reductions.** In some cases, there exist finite-dimensional subspaces V of functions which are invariant under the action of the Koopman operator. Then, the linear transport equation (5) reduces exactly to a linear diVerential equation on these subspaces:

$$\dot{\varphi} = \mathcal{K}\varphi, \quad \varphi \in V.$$

  The main example of such a situation can be found in [26], where a system of two polynomial diVerential equations are lifted to a system of three linear diVerential equations by considering a basis of multinomials (note that this is a particular case where the Carleman linearization [29] of a polynomial system is exact and does not need to be truncated).
  Although the existence of such subspaces holds great interest for applications, and there have been eVorts to recover them from data [30], generally speaking, this seems to be a very rare situation. Let us point out also that even though in this example, the dimension of the invariant subspace is small, in general it can be much larger than the dimension d of the underlying problem.
  Another way to find invariant subspaces is to find eigenfunctions of the Koopman operator. Then, any subspace spanned by a finite number of eigenfunctions $\varphi_1, \dots, \varphi_N$ with eigenvalues $\lambda_1, \dots, \lambda_N$ will be invariant under the action of the Koopman operator, and the linear transport equation (5) reduces exactly to the following diagonal system of linear diVerential equations:

$$\dot{\varphi}_i = \lambda_i \varphi_i, \quad i \in \{1, \dots, N\}.$$

  As we have mentioned earlier, eigenfunctions of the Koopman operator, when they exist, represent relevant quantities for the underlying system. However, when the goal is system identification per se, one must keep in mind that there is no reason a priori for them to span an approximation of the Koopman operator that is relevant for system identification.
- **Spectral properties of the Koopman operator.** On the other hand, the complete spectral analysis of the Koopman operator is a delicate topic, and depends heavily on the choice of a function space. On the whole space $\mathbb{R}^d$, the transport operator

$$\mathcal{K} = f(x) \cdot \nabla, \quad \mathcal{D}(\mathcal{K}) = H^1(\mathbb{R}^d)$$

  does not have a natural basis of eigenfunctions, sometimes no eigenfunctions at all. This is related to the fact that in general transport operators do not have a compact resolvent, and are not generally self-adjoint unless the velocity field f has zero divergence.
  Let us nevertheless point out an interesting recent development for the discrete-time Koopman operator introduced in Remark 2.1. In the case where Ω is the unit circle in $\mathbb{R}^2$, Ref. [31] shows that an adequate choice of a function space (namely, Hilbert–Hardy spaces) leads to interesting spectral results. Indeed, by restricting the function space to a class of analytical functions, the adjoint of the Koopman operator (the Perron–Frobenius operator) becomes compact. By duality, this means that in this case, by extending the discrete-time Koopman operator to a larger space containing L^2, it becomes compact. When it is self-adjoint, which is equivalent to the existence of an invariant measure, it then has a basis of eigenfunctions. This result however concerns a certain class of discrete-time dynamical systems, and analogous results in the continuous-time setting, regarding the Koopman generator (8), have yet to be explored.

In terms of data-driven methods, spectral analysis of the Koopman operator from data has received a lot of attention (see [32–36]), in particular the data-driven recovery of eigenfunctions when they exist. In [32], as is often the case in numerical analysis, only the eigenvalues with the smallest moduli can be accurately approximated. This is of interest to isolate components of the observed system that evolve slowly in time (or are invariant, if the eigenvalue is 0). However, these give only a very partial picture of how the system actually evolves. In [35], the study is restricted to measure-preserving systems, an important assumption which has since been sidestepped in [36].
As we have pointed out earlier, in general, transport operators (Koopman operators) do not have nice spectral properties. For system identification and predictions, there seems to be better hope in finding accurate and appropriate finite-dimensional approximations of the Koopman operator independently of such properties.

- **Galerkin projections and Extended Dynamic Mode Decomposition.** Classical Galerkin methods used in numerical analysis, which approximate solutions of partial diVerential equations using approximation spaces, seem a natural solution to provide reliable approximations of the Koopman operator. They have indeed proven to be an e Y cient way to obtain reduced-order models of partial diVerential equations without having to use spectral decompositions.
To recover a data-driven approximation of such a Galerkin projection, one of the main algorithms is the so-called Extended Dynamic Mode Decomposition (EDMD) (introduced in [37]). Its predecessor algorithm, Dynamic Mode Decomposition (DMD), computes the "best fit" system of linear diVerential equations for the unknown system (4) in the state space $\mathbb{R}^d$. EDMD does this in the subspace of functions given by the Galerkin projection. As this corresponds to recovering the best linear approximation of a linear infinite-dimensional operator, this has a better chance of yielding convincing results than DMD.
EDMD was first introduced in the discrete-time framework, to recover approximations of the discrete-time Koopman operator, that is, the classical composition operator. It has been actively explored as a way to perform data-driven system identification and predictions [38–40]. It can also provide approximations of so-called coarse-grain models (see [41]) for highly complex systems such as molecular dynamics [28]. It has been used in combination with model predictive control to achieve data-driven control of systems with unknown dynamics [27], and combined with deep learning techniques [42–46]. It has recently been modified to recover eigenfunctions from data in [36]. In this last reference, two Galerkin approximations are constructed, which gives robustness and convergence guarantees when computing spectra from data.
In continuous time, there are two main approaches using EDMD to recover an approximation of the Koopman operator.
  (1) The "Koopman lifting method", proposed in [47], where the authors apply EDMD to a continuous-time system to obtain an approximation of the Koopman semigroup $U(\tau)$ for a sampling time $\tau > 0$. Then, they compute its matrix logarithm to produce an approximation of the infinitesimal generator $\mathscr{K}$.
  (2) Generator Extended Dynamic Mode Decomposition (gEDMD), proposed in [28], where EDMD is adapted to directly produce an approximation of $\mathscr{K}$ by processing data in diVerent manner.

2.3. *Main contributions and structure of the article*

In this article, we describe and implement a variant of gEDMD, using classical finite element spaces to define finite-dimensional approximations of the Koopman operator. Our central aim is

to quantify the quality and cost of system identification and predictions by this method, under generic hypotheses which allow for clear-cut estimates.

More precisely, we will focus on the case of an unknown vector field f of a certain regularity, of *known dimension* d, on a *bounded* domain $\Omega \in \mathbb{R}^d$ with Lipschitz continuous boundary, as it would be unreasonable to expect in general a good approximation of f on all of $\mathbb{R}^d$ from a finite number of observations. Moreover, to simplify matters and alleviate notations, we will focus on the case where the flow Φ_f^t satisfies

$$\Phi_f^t(\Omega) \subset \Omega. \tag{12}$$

Finally, for the sake of simplicity, we will not consider measurement noise. This simplification does not aVect the overall message (see below) and allows for a lighter presentation of the results.

In Section 3, we give estimates on the finite-dimensional approximations of the Koopman operator, thanks to properties of finite element spaces. We show how these approximations can be used to solve the system identification problem. The estimates depend on the size of the mesh and on the degree of the finite element functions we consider, which are related to the dimension of the finite element space. Accurate Galerkin projections require finite element spaces of considerable dimension, which increases exponentially with the dimension d of the system under consideration.

In Section 4 we devise a variant of gEDMD to compute a data-driven approximation of these Galerkin projections. The slight improvement we propose simplifies the algorithm thanks to the Galerkin projections being used. The convergence of this approximation depends on the number of data samples, and is essentially based on Monte-Carlo approximation of integrals. We then analyze the total approximation error of gEDMD with finite elements, and estimate its computational cost, in order to quantify whether this method can be eYcient for system identification and predictions in practice.

This allows us to give some clear insights on the challenges and limitations of this method. The overall conclusion is that, even in the favorable setting of the analysis, and even when using accurate approximation spaces such as finite elements, the approximation error decreases relatively slowly, not only because the convergence of the data-driven approximation is quite slow (in $m^{-1/2}$, m being the number of samples), but also because this convergence is further slowed if one chooses a finite element space of greater dimension (i.e., refines the mesh or considers higher-order finite elements).

Finally in Section 5 we focus on the case $d = 1$ to establish a comparison with direct interpolation methods, given the same data sample. Interpolation methods turn out to be advantageous, due to the slowness of the convergence of gEDMD. We give some numerical illustrations of this verdict.

2.4. *Notations*

We denote the usual Lebesgue measure by μ.

For the rest of this article, for $\Omega \subset \mathbb{R}^d$, when there is no ambiguity we will note $L^2(\Omega, \mu) = L^2$ to alleviate notations. Notice that if $f \in L^\infty(\mathbb{R}^d, \mathbb{R}^d)$, by the definition (9) we have

$$H^1(\mathbb{R}^d) \subset \mathscr{D}(\mathscr{K}). \tag{13}$$

We denote by $\|\cdot\|$ the usual L^2 norm, $\|\cdot\|_{H^s(\Omega)}$ the usual Sobolev norms for $s \geq 0$, and $|\cdot|_{H^k(\Omega)}$ the Sobolev seminorms for $k \in \mathbb{N}$:

$$|\varphi|^2_{H^k(\Omega)} = \sum_{\substack{\alpha \in \mathbb{N}^d \\ |\alpha| = k}} \|D^\alpha \varphi\|^2, \quad \varphi \in H^k(\Omega). \tag{14}$$

In Euclidean spaces, $\|\cdot\|$ denotes the usual Euclidean norm.

For matrices $9 \cdot 9$ denotes the operator norm induced by the Euclidean norm, $\|\cdot\|_F$ denotes the Frobenius norm, $\|\cdot\|_\infty$ denotes the max norm

$$\|A\|_\infty = \max_{ij} |A_{ij}|.$$

The sign † denotes the Moore–Penrose pseudo-inverse, and $\top$ will denote the usual transposition.

3. Finite-dimensional approximations of the Koopman operator

3.1. *Restricting the Koopman operator to a subdomain* Ω

As we have mentioned in the introduction, we will focus on studying the Koopman operator of a vector field f on a bounded domain $\Omega \subset \mathbb{R}^d$ with Lipschitz continuous boundary.

When dealing with a vector field that is indeed defined on all of $\mathbb{R}^d$, this means we are studying the Koopman operator $\mathcal{K}_\Omega$ of the restriction $f_{|\Omega}$:

$$\mathcal{K}_\Omega = f_{|\Omega} \cdot \nabla. \tag{15}$$

Again, $\mathcal{K}_\Omega$ is an unbounded operator, with domain

$$\mathcal{D}(\mathcal{K}_\Omega) = \{\varphi \in L^2(\Omega),\ f \cdot \nabla\varphi \in L^2(\Omega)\}. \tag{16}$$

However, with this domain, in general $\mathcal{K}_\Omega$ does not generate a semigroup as in (5). Indeed in general the flow Φ_f^t does not preserve Ω. Denoting the following

$$\partial\Omega^{\text{out}} = \{x \in \partial\Omega,\ f(x) \cdot \nu(x) > 0\}, \tag{17}$$

where $\nu(\cdot)$ denotes the outward normal unit vector, it is necessary to specify boundary conditions on $\partial\Omega^{\text{out}}$ in order for the corresponding linear transport equation to be well-posed. Typically, the partial diVerential equation

$$\begin{cases} \dfrac{\partial\varphi}{\partial t} = f(x) \cdot \nabla\varphi, \quad x \in \Omega, \\ \varphi(t,x) = 0 \quad x \in \partial\Omega^{\text{out}}, \end{cases} \tag{18}$$

is well-posed. Accordingly, the unbounded operator

$$\tilde{\mathcal{K}}_\Omega := f \cdot \nabla, \quad \mathcal{D}(\tilde{\mathcal{K}}_\Omega) = \{\varphi \in H^1(\Omega),\ \varphi = 0 \text{ on } \partial\Omega^{\text{out}}\}, \tag{19}$$

generates the C^0-semigroup

$$e^{t\tilde{\mathcal{K}}_\Omega}\varphi_0(x) = \begin{cases} \varphi_0(\Phi_f^t(x)) & \text{if } \Phi_f^t(x) \in \Omega, \\ 0 \quad \text{otherwise,} \end{cases} \qquad \forall \varphi_0 \in L^2(\Omega),\ \forall x \in \Omega. \tag{20}$$

3.2. *System identification and predictions with the Koopman operator*

We now show how the Koopman operator can be used to recover the vector field f, and in some cases, its flow Φ_f^t.

Let us note the coordinate functions

$$g_i(x) = x_i, \quad x \in \Omega,\ i = 1,\dots,d. \tag{21}$$

Then, as they are polynomial, and Ω is a bounded domain,

$$g_i \in \mathcal{D}(\mathcal{K}_\Omega), \quad i = 1,\dots,d. \tag{22}$$

Moreover,

$$\mathcal{K}_\Omega g_i = f \cdot \nabla g_i = f_i, \quad i = 1,\dots,d, \tag{23}$$

where the f_i denotes the coordinates of the vector field f in (4). It is thus possible, with the restricted Koopman operator $\mathscr{K}_\Omega$, to recover the vector field f. On the other hand, in general

$$g_i \notin \mathscr{D}(\tilde{\mathscr{K}}_\Omega)$$

due to the boundary conditions in (19). However, we can still apply the semigroup defined in (20):

$$e^{t\tilde{\mathscr{K}}_\Omega} g_i(x) = \begin{cases} (\Phi_f^t(x))_i & \text{if } \Phi_f^t(x) \in \Omega, \\ 0 \quad \text{otherwise}, & \forall x \in \Omega, \end{cases} \tag{24}$$

where the $(\Phi_f^t(\cdot))_i$ denotes the coordinates of the flow (4). Thus, it is possible, with the restricted Koopman operator $\tilde{\mathscr{K}}_\Omega$, to recover the flow Φ_f^t, and thus make predictions, as long as the flow does not exit Ω.

To make matters simpler, in order to bring forth our conclusions with more clarity, we will for the rest of this article focus on the case where

$$\mu_{\partial\Omega}(\partial\Omega^{\text{out}}) = 0,$$

i.e., the case where Ω is forward invariant under Φ_f^t. Then, we will set, to alleviate notations,

$$\mathscr{K} := \mathscr{K}_\Omega = f \cdot \nabla, \quad \mathscr{D}(\mathscr{K}) := \mathscr{D}(\mathscr{K}_\Omega) = \mathscr{D}(\tilde{\mathscr{K}}_\Omega).$$

With this domain, $\mathscr{K}$ generates the semigroup:

$$e^{t\mathscr{K}} \varphi_0(x) = \varphi_0(\Phi_f^t(x)), \quad \forall \varphi_0 \in L^2(\Omega), \ \forall x \in \Omega. \tag{25}$$

i.e., the linear transport equation

$$\begin{cases} \dfrac{\partial \varphi}{\partial t}(t,x) = f(x) \cdot \nabla \varphi(t,x), \quad t \geq 0, x \in \Omega, \\ \varphi(0) = \varphi_0 \in L^2(\Omega), \end{cases} \tag{26}$$

is well-posed.

In particular, Equation (23) remains unchanged, and (24) becomes

$$e^{t\mathscr{K}} g_i(x) = (\Phi_f^t(x))_i, \quad \forall x \in \Omega. \tag{27}$$

3.3. *Galerkin projection of the Koopman operator*

We now consider a bounded domain $\Omega \subset \mathbb{R}^d$, and we focus on the case where f is such that

$$\Phi_f^t(\Omega) \subset \Omega.$$

On this domain, the Koopman operator of f

$$\mathscr{K} = f \cdot \nabla$$

with domain

$$\mathscr{D}(\mathscr{K}) = H^1(\Omega)$$

generates the semigroup

$$e^{t\mathscr{K}} \varphi_0 = \varphi_0 \circ \Phi_f^t, \quad \forall \varphi_0 \in L^2.$$

The idea of Koopman operator methods is to observe trajectories of the dynamical system (4), but instead of trying to identify the vector field f directly, to apply linear regression methods to identify the linear operator $\mathscr{K}$.

In practice, identifying $\mathscr{K}$ numerically is impossible as it is infinite dimensional. Instead, the goal is to identify a Galerkin projection of $\mathscr{K}$, which corresponds to a reduced-order model of the linear transport equation (5). The projection is determined by a family of linearly independent

functions $\Psi := \{\psi_1, \dots, \psi_N\} \subset L^2$, often called *dictionary* in the literature, which span a finite-dimensional subspace $V_N \subset L^2$ which should satisfy

$$V_N \subset \mathscr{D}(\mathscr{K}). \tag{28}$$

Then, the Galerkin projection of $\mathscr{K}$ is given by

$$\mathscr{K}_N := \Pi_N \mathscr{K} \Pi_N, \tag{29}$$

where Π_N denotes the L^2-orthogonal projection on V_N. In particular,

$$\mathscr{K}_N \varphi = \Pi_N \mathscr{K} \varphi, \quad \varphi \in V_N. \tag{30}$$

In this article we will focus on a specific choice of V_N, given by continuous finite elements. Classical approximation results for these spaces then ensure the Galerkin projection is a relevant approximation of the actual Koopman operator.

3.4. *Galerkin projection on finite element spaces*

In order to alleviate notations, and simplify matters pertaining to the choice and regularity of meshes, we fix for the remainder of this article

$$\overline{\Omega} = \mathscr{Q} := [0, L]^d. \tag{31}$$

Following [11, Section 3.4], we then define the family of rectangulations of $\mathscr{Q}$:

$$\mathscr{T}_M = \left\{ \prod_{i=1}^{d} \left[p_i \frac{L}{M}, (p_i + 1) \frac{L}{M} \right], \ 0 \leq p_i \leq M - 1, \ i \in \{1, \dots, d\} \right\}, \quad M \in \mathbb{N} \setminus \{0\}. \tag{32}$$

In what follows, to bring forth the dependence in the mesh size, we will denote $\mathscr{T}_h := \mathscr{T}_M$ where

$$h = \frac{L}{M}.$$

These rectangulations form a regular family of rectangulations of $\mathscr{Q}$.

For $k \geq 1$, we define $\mathbb{Q}_k$ the space of polynomials of degree at most k *in each variable*, and the following continuous finite element spaces:

$$V_h^k = \{v \in C^0(\mathscr{Q}), \ v_{|K} \in \mathbb{Q}_k, \ \forall K \in \mathscr{T}_h\}. \tag{33}$$

We consider the nodes

$$\hat{x}_\alpha := \frac{h}{k} \alpha, \quad \alpha \in \left\{ 0, \dots, k \frac{L}{h} \right\}^d,$$

and define

$$N_h^k := (1 + kM)^d = \left(1 + \frac{kL}{h} \right)^d \tag{34}$$

in order to re-index the nodes by

$$\hat{x}_j, \quad j \in \{1, \dots, N_h^k\}.$$

Then the classical shape functions ψ_j which satisfy

$$\psi_j(\hat{x}_k) = \delta_{jk} \quad \forall j, \ k \in \{1, \dots, N_h^k\},$$

form a basis of V_h^k, so that

$$\dim V_h^k = N_h^k, \tag{35}$$

i.e.,

$$h = \frac{kL}{(N_h^k)^{\frac{1}{d}} - 1}. \tag{36}$$

One can see from (35) that both refining the mesh ($h \to 0$) and taking finite elements of higher degree ($k \to \infty$) will increase the dimension of the approximation subspace at the same rate.

We now recall classical projection inequalities for finite element spaces. Note r_h^k the classical finite element interpolation operator

$$r_h^k(\varphi) = \sum_{j=1}^{N_h^k} \varphi(x_j)\psi_j, \quad \varphi \in C^0(\mathcal{Q}). \tag{37}$$

The classical interpolation inequalities for triangular finite elements (see for example [11, Section 3.4]) can be adapted to the rectangulation $\mathcal{T}_h$:

Proposition 3.1. *Let $\mathcal{T}_h$ be the rectangulation defined by* (32)*. Let $s > d/2$, $\nu \in \{0,1\}$, and define $l = \min(k, s-1)$. Then, for $\varphi \in H^s(\mathcal{Q})$, $\varphi \in C^0(\mathcal{Q})$ so that $r_h^k(\varphi)$ is well defined, and there exists $C > 0$ independent of h such that*

$$\|r_h^k(\varphi) - \varphi\|_{H^\nu(\mathcal{Q})} \leq C h^{l+1-\nu} |\varphi|_{H^{l+1}(\mathcal{Q})}, \quad \forall \varphi \in H^s(\mathcal{Q}). \tag{38}$$

Denoting Π_h^k the L^2-orthogonal projection on V_h^k, one gets, by property of orthogonal projections in Hilbert spaces:

Corollary 3.1. *With the notations above,*

$$\|\Pi_h^k(\varphi) - \varphi\|_{H^\nu(\mathcal{Q})} \leq C h^{l+1-\nu} |\varphi|_{H^{l+1}(\mathcal{Q})}, \quad \forall \varphi \in H^s(\mathcal{Q}). \tag{39}$$

Now, we have

$$V_h^k \subset \mathcal{D}(\mathcal{K}). \tag{40}$$

Following (29), the Galerkin projection of $\mathcal{K}$ is then given by

$$\mathcal{K}_h^k := \Pi_h^k \mathcal{K} \Pi_h^k.$$

3.5. *System identification and predictions*

We now show how the approximation power of finite elements can be combined with (23) and (27) to obtain approximations of the vector field f and its flow Φ_f^t.

First, as the g_i are linear, we clearly have

$$g_i \in V_h^k, \quad \forall i \in \{1, \dots, d\}. \tag{41}$$

We then have, from (23),

$$\mathcal{K}_h^k g_i = \Pi_h^k \mathcal{K} g_i = \Pi_h^k f_i, \tag{42}$$

which can be understood as a finite element approximation of (23).

In the same spirit, we obtain a finite element approximation of (26) by the Galerkin method, given by the following ODE in the finite-dimensional space V_h^k:

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\varphi_i^{k,h} = \mathcal{K}_h^k \varphi_i^{k,h}, \quad \varphi_i^{k,h} \in V_h^k, \\ \varphi_i^{k,h}(0) = g_i, \quad i \in \{1, \dots, d\}. \end{cases} \tag{43}$$

Remark 3.1. In accordance with the assumptions we have made on the domain Ω and the behavior of the flow, there is no need here to specify boundary conditions for the Galerkin method, as the actual linear transport equation (26) is well-posed without having to specify any boundary conditions.

Denoting

$$\varphi_i^{k,h}(t) = \sum_{j=1}^{N_h^k} (\varphi_i^{k,h}(t))_j \psi_j, \quad g_i = \sum_{j=1}^{N_h^k} (\mathbf{g}_i)_j \psi_j.$$

We have the following system of linear diVerential equations

$$\begin{cases} M_h^k \dfrac{\mathrm{d}}{\mathrm{d}t}\varphi_i^{k,h} = R_h^k \varphi_i^{k,h}, \\ \varphi_i^{k,h}(0) = \mathbf{g}_i, \end{cases} \tag{44}$$

where $M_h^k = (\langle \psi_i, \psi_j \rangle)_{i,j}$, $R_h^k = (\langle \psi_i, \mathscr{K}_h^k \psi_j \rangle)_{i,j}$ are the *mass and stiVness matrices* of the basis (ψ_j), for the operator $\mathscr{K}_h^k$: noting K_h^k its matrix in the basis (ψ_j), we have

$$K_h^k = (M_h^k)^{-1} R_h^k. \tag{45}$$

We see from (42) and (43) that solving the system identification problem with a Galerkin projection corresponds to projecting the vector field on the finite element space V_h^k, and approximating the solutions of the linear transport equation (26) as in classical numerical analysis. With this in mind, the choice of classical approximation spaces used in numerical analysis for the approximation of partial diVerential equations seems adequate.

Accordingly, under some conditions on the order k of the finite elements, and the regularity of the vector field f, we have the following estimates both on the approximation of f and its flow Φ_f^t:

Proposition 3.2. *Let $k+1 > d/2$. There exist constants $C_1, C_2(f) > 0$ such that, for $f \in C^{k+1}(\mathscr{Q}, \mathbb{R}^d)$, the following error estimates hold:*

$$\|\mathscr{K}_h^k g_i - f_i\| \leq C_1 h^{k+1} |f_i|_{H^{k+1}(\mathscr{Q})}. \tag{46}$$

$$\sup_{t\in[0,T]} \|(\Phi_f^t)_i - \varphi_i^{k,h}(t)\| \leq C_2(f) h^k. \tag{47}$$

Proof. From (42), we have

$$\|\mathscr{K}_h^k g_i - f_i\| = \|\Pi_h^k f_i - f_i\|. \tag{48}$$

To estimate the right hand-term, we apply Corollary 3.1 with $s = k+1$ to the $f_i \in C^{k+1}(\mathscr{Q}) \subset H^{k+1}(\mathscr{Q})$ (as $\mathscr{Q}$ is bounded), which directly yields (46).

Now consider the solutions of the following linear transport equations:

$$\begin{cases} \dfrac{\partial \varphi_i}{\partial t}(t,x) = f(x) \cdot \nabla \varphi_i(t,x), \quad t \geq 0,\ x \in \mathscr{Q}, \\ \varphi_i(0) = g_i, \quad i \in \{1,\dots,d\}. \end{cases} \tag{49}$$

Recalling (27), these solutions are given by

$$\varphi_i(t,x) = g_i(\Phi_f^t(x)) = (\Phi_f^t(x))_i. \tag{50}$$

As $f \in C^{k+1}(\mathscr{Q}, \mathbb{R}^d)$, by the Cauchy–Lipschitz theorem, for all $i \in \{1,\dots,d\}$,

$$(t,x) \mapsto (\Phi_f^t(x))_i \tag{51}$$

is in $C^{k+1}([0,T] \times \mathscr{Q})$. Hence, $\varphi_i \in L^2(0,T; H^{k+1}(\mathscr{Q})) \cap H^1(0,T; H^k(\mathscr{Q}))$. Adapting [11, Chapter 14, Section 3] to our setting, which is simplified by the absence of boundary conditions and the fact that $\varphi_i^{k,h}(0) \in V_h^k$, we can then estimate the convergence of solutions of (43) to the φ_i: there exists a constant $C_2(f) > 0$ such that

$$\sup_{t\in[0,T]} \|\varphi_i(t) - \varphi_i^{k,h}(t)\| \leq C_2(f) h^k. \tag{52}$$

Together with (50), this directly yields (47).

Thus, Galerkin projections of the Koopman operator on finite element spaces are relevant objects to approximately solve the system identification problem, both to recover the vector field f and to approximate its flow Φ_f^t. However the accuracy obtained in the estimates (46) and (47) comes at a cost, as we will see in the next section.

3.6. *Curse of dimensionality*

In Proposition 3.2 we have given the error estimate (46) for system identification using the Galerkin projection $\mathscr{K}_h^k$ of $\mathscr{K}$ on finite element spaces. In terms of the dimension of the finite element space, we have the following:

Proposition 3.3. *The dimension N_h^k of the finite element spaces needed to ensure the quality of approximation $\varepsilon > 0$ for system identification grows exponentially with the dimension d of the underlying space:*

$$N_h^k \geq C''(L, f) k^d \varepsilon^{-\frac{d}{k+1}}, \tag{53}$$

for some constant $C'''(L, f) > 0$.

Proof. If one requires the quality of approximation in (46):

$$C_1 h^{k+1} |f|_{H^{k+1}(\mathscr{Q})} \leq \varepsilon, \tag{54}$$

then

$$h \leq C'(f) \varepsilon^{\frac{1}{k+1}}. \tag{55}$$

From (34) we then get

$$N_h^k = \left(1 + \frac{kL}{h}\right)^d \geq C''(L, f) k^d \varepsilon^{-\frac{d}{k+1}}, \tag{56}$$

which proves (53).

This is a form of curse of dimensionality, a phenomenon first pointed out in [48] in the context of dynamic programming. It is present in a wide range of situations such as function approximation [49, 50], integral computation [51], approximation of parametric PDEs [52, 53], where the complexity of a problem increases exponentially with its underlying dimension. On the other hand, for a fixed dimension d, the required dimension N_h^k is polynomial in the error ε.

This curse of dimensionality is a common feature in approximation theory. It stems from the use of a mesh to define the approximation spaces (here, continuous functions that are polynomial on each cell of the mesh). This allows a very accurate approximation, but the complexity of the mesh increases exponentially with the dimension, hence the curse of dimensionality.

Recalling that predictions are made by integrating the system of N_h^k linear diVerential equations (44), this means that in high dimension d, making predictions by using Galerkin projections of the Koopman operator becomes prohibitively expensive.

In the next sections we will also see that this curse of dimensionality has an impact on the data-driven approximation of the Galerkin projections.

4. A variant of generator Extended Dynamic Mode Decomposition

Having given estimates on how the Galerkin projection $\mathscr{K}_h^k$ converges to the Koopman operator $\mathscr{K}$ under suitable regularity assumptions on the vector field f, it remains to compute a data-driven approximation of $\mathscr{K}_h^k$. To achieve this, we present a modified version of the gEDMD algorithm.

4.1. *Data sampling*

The idea of data-driven system identification is to compute an approximation of $\mathscr{K}_h^k$ from a finite number of measurements (the so-called data). Practically speaking, we consider a collection of points $\{x^l \in \mathscr{Q}, l \in \{1, \dots, m\}\}$, and for each $l \in \{1, \dots, m\}$. The measurements are obtained from the solutions $x^l(t)$ of (4) with initial condition x^l, by approximating the initial velocities $\dot{x}^k(0) = f(x^k)$. This is usually done by total variation regularized diVerentiation, which is also used in other methods such as SINDy (see [12, 54]).

The idea is then to use this measurement data to approximate $\mathscr{K}_h^k$, more precisely by computing an approximation of its rigidity matrix. Essentially, this boils down to approximating the integrals in the coeYcients of this matrix. To handle the potentially large dimension d, and to account for the potentially random nature of measurements for real-life systems, the x^l are assumed to be drawn independently and uniformly with respect to the Lebesgue measure on $\mathscr{Q}$. The integrals are then approximated by Monte-Carlo integration, the convergence of which does not depend on the dimension. Most of the algorithms (DMD, EDMD) presented in the literature on Koopman operators rely on this integration method. Thus, in a simultaneous endeavour to give a rigorous analysis of the existing Koopman operator methods (with only a minor improvement, see below in Section 4.2) and to cover the most general case, this will be the focus of this article.

4.2. *The algorithm*

Recalling the identity (45), we see that there are two distinct components in the Galerkin projection $\mathscr{K}_h^k$: the mass matrix M_h^k, which depends on the space V_h^k and the basis (ψ_j) (thus, essentially, on the domain $\mathscr{Q}$), and the stiVness matrix R_h^k which depends on the vector field f, the space V_h^k, and the basis (ψ_j). In general, the mass matrix can be approximated by numerical integration methods such as quadrature formulae, or Monte-Carlo integration. In the case of finite elements, in particular linear finite elements, it can even be given explicitly. In any case, this does not require trajectory data, as it does not depend on f, and can thus be done separately.

On the other hand, the stiVness matrix R_h^k depends on f. The algorithm we now lay out mainly consists in approximating this matrix from observations of trajectories of the system (4). To our knowledge, this approach, consisting in separating the approximation of the mass and stiVness matrices, is new. As we will see, it has the advantage of allowing the gEDMD method to be extended to localized bases such as finite elements, or wavelets.

We then proceed as follows:

Step 1. From the data set $\{x^l, y^l := f(x^l), l \in \{1,\dots,m\}\}$, define the following matrices:

$$G_{k,h}^m = (\psi_i(x^j))_{\substack{i\in\{1,\dots,N\}\\ j\in\{1,\dots,m\}}}, \quad A_{k,h}^m = (y^j \cdot \nabla\psi_i(x^j))_{\substack{i\in\{1,\dots,N\}\\ j\in\{1,\dots,m\}}}, \tag{57}$$

and compute the following:

$$R_{k,h}^m := \frac{\mu(\Omega)}{m} G_{k,h}^m (A_{k,h}^m)^\top = \frac{L^d}{m} G_{k,h}^m (A_{k,h}^m)^\top. \tag{58}$$

As we will establish later on, $R_{k,h}^m$ is an approximation of the stiVness matrix R_h^k of (44).

Step 2. Numerically solve the following least-squares problem:

$$\min_{K\in\mathbb{R}^{N\times N}} \|M_h^k K - R_{k,h}^m\|_F^2, \tag{59}$$

the unique solution of which is given by:

$$K_{k,h}^m := (M_h^k)^{-1} R_{k,h}^m. \tag{60}$$

4.3. *Convergence*

We will now prove that $K_{k,h}^m$ converges to the matrix K_h^k given by (45).

Now, as we have considered random points $(x^l)_{l=1}^m$, this means that the coeYcients of

$$K_h^k - K_{k,h}^m$$

are random variables. The notion of convergence of random variables we consider here is then the notion of convergence in distribution:

Definition 4.1 (Convergence in distribution of a sequence of random variables). A sequence $(X_n)_{n\in\mathbb{N}}$ of real random variables is said to converge in distribution to a real random variable X:

$$X_n \xrightarrow[n\to\infty]{\mathscr{D}} X$$

if their cumulative distribution functions converge pointwise to the cumulative distribution function of X on its continuity set.

We can now state the following convergence result (see also [38, 55, 56]), which focuses on the convergence of the coefficients of the approximate stiffness matrix $R^m_{k,h}$:

Proposition 4.1 (Convergence and complexity of gEDMD). *The computational complexity of gEDMD has the upper bound $\mathcal{O}(mN^2+N^3)$.*

For $f\in L^\infty(\mathcal{Q},\mathbb{R}^d)$, the following convergence in distribution holds:

$$\sqrt{m}((R^m_{k,h})_{ij}-(R^k_h)_{ij})\xrightarrow[m\to\infty]{\mathscr{D}}\mathcal{N}(0,\sigma^{k,h}_{ij}), \tag{61}$$

where

$$(\sigma^{k,h}_{ij})^2=\int_{\mathcal{Q}}\psi_i^2(x)(f(x)\cdot\nabla\psi_j(x))^2\,\mathrm{d}x-\left(\int_{\mathcal{Q}}\psi_i(x)f(x)\cdot\nabla\psi_j(x)\,\mathrm{d}x\right)^2, \tag{62}$$

and $\mathcal{N}(0,\sigma^{k,h}_{ij})$ is the normal distribution with mean 0 and standard deviation $\sigma^{k,h}_{ij}$.

Proof. The upper bound for the complexity is given by the complexity of the linear regression that is performed to compute $K^m_{k,h}$. It is well-known (see [57, Appendix D]) that this complexity is bounded by $\mathcal{O}(mN^2+N^3)$ due to the matrix product operation, and the pseudo-inversion that follows.

As for the convergence in distribution, notice that the coefficients of $R^m_{k,h}$ are given by

$$(R^m_{k,h})_{ij}=\frac{L^d}{m}\sum_{l=1}^m\psi_i(x^l)f(x^l)\cdot\nabla\psi_j(x^l),$$

which is none other than the classical Monte-Carlo approximation of the integral

$$\langle\psi_i,f\cdot\nabla\psi_j\rangle=\int_{\mathcal{Q}}\psi_i(x)f(x)\cdot\nabla\psi_j(x)\,\mathrm{d}x=(R^k_h)_{ij},$$

as the sampling points x^l are drawn independently and uniformly with respect to the Lebesgue measure.

Convergence (61) then follows directly from a classical result of Monte-Carlo integration (see [58, Theorem 2.1]:

$$\sqrt{m}\left(\left(\frac{1}{m}\sum_{l=1}^m\psi_i(x^l)f(x^l)\cdot\nabla\psi_j(x^l)\right)-\frac{1}{\mu(\Omega)}\int_{\mathcal{Q}}\psi_i(x)f(x)\cdot\nabla\psi_j(x)\,\mathrm{d}x\right)\xrightarrow[m\to\infty]{\mathscr{D}}\mathcal{N}(0,\sigma^{k,h}_{ij}). \tag{63}$$

Remark 4.1. In the usual implementation of gEDMD, the mass matrix is not computed offline, but also approximated with the Monte-Carlo approximation

$$M^m_{k,h}:=\frac{L^d}{m}G^m_{k,h}(G^m_{k,h})^\top.$$

The matrix $K^m_{k,h}$ is then given by

$$K^m_{k,h}=(M^m_{k,h})^\dagger R^m_{k,h},$$

which converges to K^k_h provided the pseudo-inverses converge. By a classical result due to Penrose [59], this is the case if, for m large enough, the $M^m_{k,h}$ are invertible with probability 1. This corresponds to the following condition:

$$\varphi(x^1)=\cdots=\varphi(x^m)=0\implies\varphi=0,\quad\forall\varphi\in V^k_h. \tag{64}$$

When the $(x^j)_j$ are drawn uniformly, this is equivalent to

$$\forall \varphi \in V_N \setminus \{0\}, \quad \mu\{x \in \mathscr{Q},\, \varphi(x) = 0\} = 0, \tag{65}$$

which is not satisfied by the shape functions $(\psi_j)_j$, nor by other classical localized approximation functions such as wavelets.

In our case, to ensure (64), it would be necessary to draw sample points in every element of the rectangulation $\mathscr{T}_h$, which would be extremely impractical.

4.4. *Convergence in matrix norms*

Proposition 4.1 essentially means that, for a given confidence rate $\alpha \in [0,1)$, there exists a constant $C_{ij}^{k,h}(\alpha)$ given by the normal distribution $\mathscr{N}(0, \sigma_{ij}^{k,h})$, such that for large enough m the following estimate holds with probability at least α:

$$|(R_{k,h}^m)_{ij} - (R_h^k)_{ij}| \leq C_{ij}^{k,h}(\alpha) m^{-\frac{1}{2}}. \tag{66}$$

This probabilistic estimate tells us that the coefficients converge in $m^{-1/2}$, with a constant which depends on the standard deviation $\sigma_{ij}^{k,h}$, that is, recalling its expression (62), on the shape functions (ψ_j) and f. We then note

$$C_h^k(f, \alpha) := \max_{ij}(C_{ij}^{k,h}(\alpha)). \tag{67}$$

Now as we work with L^2 functions, we need to relate this matrix norm to the operator norm on $L^2(\mathscr{Q})$. We define the following norm on $\mathbb{R}^{N_h^k}$:

$$\|c\|_{L^2}^2 = \left\| \sum_{j=1}^{N_h^k} c_j \psi_j \right\|^2 = c^\top M_h^k c = \left\| (M_h^k)^{\frac{1}{2}} c \right\|^2. \tag{68}$$

The matrix M_h^k is symmetric positive-definite, we note $\rho_{k,h}^+$ and $\rho_{k,h}^-$ its largest and smallest eigenvalue respectively. We then have

$$\sqrt{\rho_{k,h}^-}\|c\| \leq \|c\|_{L^2} \leq \sqrt{\rho_{k,h}^+}\|c\|, \quad \forall c \in \mathbb{R}^{N_h^k}. \tag{69}$$

With this norm we then define the induced operator norm on $\mathbb{R}^{N_h^k \times N_h^k}$:

$$|\!|\!| A |\!|\!|_{L^2} = \sup_{\|c\|_{L^2}=1} \|Ac\|_{L^2}. \tag{70}$$

Applying (69), we then have

$$\sqrt{\frac{\rho_{k,h}^-}{\rho_{k,h}^+}}|\!|\!| A |\!|\!|_{L^2} \leq |\!|\!| A |\!|\!| \leq \sqrt{\frac{\rho_{k,h}^+}{\rho_{k,h}^-}}|\!|\!| A |\!|\!|_{L^2}, \quad \forall A \in \mathbb{R}^{N_h^k \times N_h^k}. \tag{71}$$

We can now state the following proposition:

Proposition 4.2 (Convergence of gEDMD in matrix norm). *Let $\alpha \in [0,1)$. For $f \in L^\infty(\Omega, \mathbb{R}^d)$, there exists a constant $C_h^k(f,\alpha)$ such that, for m large enough and with probability at least α,*

$$|\!|\!| K_h^k - K_{k,h}^m |\!|\!|_{L^2} \leq \frac{N_h^k}{\rho_{k,h}^-}\sqrt{\frac{\rho_{k,h}^+}{\rho_{k,h}^-}} C_h^k(f,\alpha) m^{-\frac{1}{2}}. \tag{72}$$

Proof. From the probabilistic estimate (66) we can then deduce the following estimate on the max matrix norm: for m large enough, with probability at least α,

$$\|R_{k,h}^m - R_h^k\|_\infty \leq C_h^k(f,\alpha) m^{-\frac{1}{2}}. \tag{73}$$

Now, we recall the result of [60] on sharp equivalence constants for matrix norms:

$$\frac{1}{N_h^k}\,9A9 \le \|A\|_\infty \le 9A9, \quad \forall A \in \mathbb{R}^{N_h^k \times N_h^k}. \tag{74}$$

Putting (74), (71) and (73) together, we get, for m large enough and with probability at least α:

$$9R_h^k - R_{k,h}^m 9_{L^2} \le N_h^k \sqrt{\frac{\rho_{k,h}^+}{\rho_{k,h}^-}} C_h^k(f,\alpha) m^{-\frac{1}{2}}. \tag{75}$$

Finally, by submultiplicativity of operator norms, and the expression of $K_{k,h}^m$, we have

$$9K_h^k - K_{k,h}^m 9_{L^2} = 9(M_h^k)^{-1}(R_h^k - R_{k,h}^m)9_{L^2} \le 9(M_h^k)^{-1}9_{L^2}\, 9R_h^k - R_{k,h}^m 9_{L^2}. \tag{76}$$

Now, as

$$9(M_h^k)^{-1}9_{L^2} = 9(M_h^k)^{-1}9 = \frac{1}{\rho_{k,h}^-}, \tag{77}$$

we finally deduce (72) from (75).

This shows that the convergence of K_h^k to $K_{k,h}^m$, i.e., the convergence of gEDMD, is asymptotically in $m^{-1/2}$, which is quite slow but does not depend on the underlying dimension d of the problem (4). It is however further slowed down by numerous factors: the standard deviation $\sigma_{ij}^{k,h}$, which represents the oscillatory behavior of the shape functions, the conditioning number and smallest eigenvalue of the mass matrix M_h^k, and the dimension of the finite element space V_h^k.

4.5. *Using gEDMD for system identification and predictions: convergence analysis*

We now use the matrix $K_{k,h}^m$ computed by gEDMD to give an approximation of f and its flow Φ_f^t.

Recalling (23) and (42), an approximation of the f_i is given by

$$\mathcal{K}_{k,h}^m g_i, \quad i \in \{1,\dots,d\}. \tag{78}$$

Recalling (26), (27), (43), (45) and (60), an approximation of the $(\Phi_f^t)_i$ is given by the solutions to the ODE

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\varphi_{i,m}^{k,h} = \mathcal{K}_{k,h}^m \varphi_{i,m}^{k,h}, \quad \varphi_{i,m}^{k,h} \in V_h^k, \\ \varphi_{i,m}^{k,h}(0) = g_i, \quad i \in \{1,\dots,d\}. \end{cases} \tag{79}$$

In the basis (ψ_j), noting $g_i = \sum_{j=1}^{N_h^k} \gamma_j^i \psi_j$, the components of the $\mathcal{K}_{k,h}^m g_i$ are given by

$$K_{k,h}^m \gamma^i \in \mathbb{R}^{N_h^k}, \quad i \in \{1,\dots,d\} \tag{80}$$

and the components of the $\varphi_{i,m}^{k,h}(t)$ are given by the solutions to

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\varphi_{i,m}^{k,h} = K_{k,h}^m \varphi_{i,m}^{k,h}, \quad \varphi_{i,m}^{k,h} \in \mathbb{R}^{N_h^k}, \\ \varphi_{i,m}^{k,h}(0) = \gamma^i, \quad i \in \{1,\dots,d\}. \end{cases} \tag{81}$$

We now give quantitative estimates for system identification and predictions, using gEDMD with finite element spaces:

Proposition 4.3. *Let $k+1 > d/2$, $f \in C^{k+1}(\mathscr{Q})$. There exist constants $C_1, C_2(f), C_3 > 0$ such that, for a given probability rate $\alpha \in [0,1]$, one has for m large enough and with probability at least α:*

$$\begin{aligned} &\|\mathscr{K}_{k,h}^m g_i - f_i\| \leq \frac{N_h^k}{\rho_{k,h}^-}\sqrt{\frac{\rho_{k,h}^+}{\rho_{k,h}^-}}C_h^k(f,\alpha)m^{-\frac{1}{2}}\|g_i\| + C_1 h^{k+1}|f_i|_{H^{k+1}(\mathscr{Q})}, \\ &\|\varphi_{i,m}^{k,h}(t) - (\Phi_f^t)_i\| \leq C_3 t\, \mathrm{e}^{t|||K_h^k|||_{L^2}}\frac{N_h^k}{\rho_{k,h}^-}\sqrt{\frac{\rho_{k,h}^+}{\rho_{k,h}^-}}C_h^k(f,\alpha)m^{-\frac{1}{2}}\|g_i\| + C_2(f)h^k, \quad \forall i \in \{1,\dots,d\}. \end{aligned} \tag{82}$$

Proof. Recalling (23) and (42), we have the following: for $i \in \{1,\dots,d\}$,

$$\begin{aligned} \|\mathscr{K}_{k,h}^m g_i - f_i\| &\leq \|\mathscr{K}_{k,h}^m g_i - \mathscr{K}_h^k g_i\| + \|\mathscr{K}_h^k g_i - f_i\| \\ &\leq 9K_{k,h}^m - K_h^k 9_{L^2}\|g_i\| + \|\mathscr{K}_h^k g_i - f_i\|. \end{aligned} \tag{83}$$

Applying (72) and (46) yields the first inequality of (82).

Recalling (27) and (43), we have the following: for $i \in \{1,\dots,d\}$,

$$\begin{aligned} \|\varphi_{i,m}^{k,h}(t) - (\Phi_f^t)_i\| &\leq \|\varphi_{i,m}^{k,h}(t) - \varphi_i^{k,h}(t)\| + \|\varphi_i^{k,h}(t) - \varphi_i(t)\| \\ &\leq 9\mathrm{e}^{tK_{k,h}^m} - \mathrm{e}^{tK_h^k}9_{L^2}\|g_i\| + \|(\Phi_f^t)_i - \varphi_i^{k,h}(t)\|. \end{aligned} \tag{84}$$

By property of the matrix exponential,

$$9\mathrm{e}^{tK_{k,h}^m} - \mathrm{e}^{tK_h^k}9_{L^2} \leq t\,9K_{k,h}^m - K_h^k 9_{L^2}\mathrm{e}^{t|||K_h^k|||_{L^2}}\mathrm{e}^{t|||K_{k,h}^m - K_h^k|||_{L^2}}. \tag{85}$$

Combining (85) with (84) and (72) then yields the second inequality of (82).

With these inequalities it is now clear that the "slowest" term is the data-driven approximation error term in $m^{-1/2}$. It depends both on the number of samples m and the dimension of the Galerkin projection space V_h^k. More precisely, the first inequality of (82) tells us that, for given k and h, the data-driven approximation error becomes comparable to the finite element approximation error, if m satisfies

$$m \geq \left(\frac{N_h^k}{\rho_{k,h}^-}\right)^2 \frac{\rho_{k,h}^+}{\rho_{k,h}^-}C_h^k(f,\alpha)^2 h^{-2k-2}. \tag{86}$$

For a fixed k, choosing more accurate finite element spaces by decreasing the mesh size h has a double impact on the required number of samples m: on the h^{-2k-2} term, on the one hand, and on the quantity $(N_h^k/\rho_{k,h}^-)^2(\rho_{k,h}^+/\rho_{k,h}^-)C_h^k(f,\alpha)^2$ on the other hand. Moreover, due to the presence of N_h^k we see that this number of samples must also increase exponentially with d. Thus, indirectly, the convergence of gEDMD is also subject to the curse of dimensionality.

Overall, the slowness of convergence of classical Monte-Carlo approximations usually employed in gEDMD conceals the high accuracy one can obtain from classical approximation spaces, and is impacted by the typical "curse of dimensionality" that usually comes with these spaces.

In the next section we will focus on the case $d = 1$ and fix $k = 1$, in order to give precise estimates of the terms in (86). This will in turn allow us to give a precise estimate of the computational cost of gEDMD and give a clear comparison with direct interpolation methods.

4.6. *Remarks on the choice of Monte-Carlo integration*

As mentioned above, the key point in data-driven identification of the Koopman operator is the approximation of the integrals

$$\langle \psi_i, \mathscr{K}\psi_j\rangle = \int_{\mathscr{Q}} \psi_i(x)\, f(x)\cdot\nabla\psi_j(x)\,\mathrm{d}x.$$

In the above we have assumed the sampling points x^l to be chosen at random, as is usually done in the EDMD literature. However, there are alternatives:

- The first is to choose the x^l along a single trajectory. It is preferred when the system being identified is known to be ergodic [61,62], i.e., has a preserved measure. For our purposes, BirkhoV's theorem would then guarantee that the integrals in the rigidity matrix can be approximated by averaging on the sample points, given enough sample points and a long enough trajectory (see [63] for details and convergence rates).
  However, in the general case, there is no reason to have such knowledge *a priori* on the system. It could nonetheless be of interest to look for such preserved measures in a data-driven manner (perhaps by adapting [35,61] to the Perron–Frobenius operator), and then implement the ergodic strategy. This would require a thorough quantitative analysis.
- When the x^l are chosen independently from each other, it is important to note that if the x^l can be chosen freely, then a good solution, at least in low dimension, is to choose them as a quadrature rule (as is done in [36]). This yields deterministic estimates on the approximation of the integrals. In higher dimensions, this can be combined with kernel techniques to reduce computational complexity. We refer again to [36, Sections 4 and 6] for more details, and a comparison of convergence rates, which can be as high as exponential.
  In higher dimension, choosing the x^l as a low-discrepancy sequence (such as Halton, Faure or Sobol sequences) would then allow to use quasi-Monte-Carlo integration (see [58]). The idea is that these sequences improve on the purely random sampling points of the Monte-Carlo method, and allow for a better convergence rate, typically m^{-1} instead of $m^{-1/2}$.

It is important to note, however, that given the computations in (82) and (86), unless the integration method converges exponentially (see Remark 5.1), the "curse of dimensionality" drawback on the sample size, which we pointed to at the end of Section 4.5, still holds.

5. Illustration in the 1D case

5.1. *Comparison with direct interpolation in 1D for linear finite element*

In this subsection, $d = 1$ and $k = 1$. That is, we focus on gEDMD for systems on the interval $[0,L]$, using Galerkin projections on linear finite elements.

System identification using the Koopman operator appears as a quite indirect method, as it consists in observing trajectories of the diVerential equation, working with the associated transport equation, then coming back to the diVerential equation.

In the gEDMD algorithm, the values $f(x^l)$ at the sample points are approximated. Instead of using them to compute an approximation of the Koopman operator, one could also use them to interpolate f directly, without resorting to Koopman operator methods. In that case, predictions are then made by integrating the nonlinear ODE

$$\dot{x} = \tilde{f}(x), \quad x \in [0,L],$$

where $\tilde{f}$ is the interpolant of f, instead of a Galerkin method of the linear transport equation such as (81). We will focus here on linear and cubic spline interpolation.

In theory, the linear interpolant $\ell(f,x_i)$ and the natural cubic spline interpolant $\sigma(f,x_i)$ (with derivative 0 at the endpoints) satisfy the following inequalities

$$\begin{aligned} \|\ell(f,x_i) - f\| &\leq C_1\delta^2|f|_2, \quad f \in H^2(0,L), \\ \|\sigma(f,x_i) - f\| &\leq C_2\delta^4|f|_4, \quad f \in H^4(0,L), \end{aligned} \tag{87}$$

where

$$\delta = \max\{x_{n+1} - x_n, x_1, L - x_m, n \in \{1, \dots, m\}\}. \tag{88}$$

The inequality for the linear interpolant follows from classical inequalities on C^2 functions, for the cubic spline interpolation error bounds, we refer to [64].

The $\{x_i\}$ are uniformly distributed on $[0, L]$. For the random variable

$$\delta_m := \max\{x_{n+1} - x_n, x_1 + L, L - x_m, n \in \{1, \dots, m\}\}$$

we have the following asymptotics (see [65]):

$$\mathbb{E}[\delta_m] \sim_m \frac{\log(m+1)}{m+1}$$

so that (3) becomes

$$\begin{aligned} \mathbb{E}[\|\ell(f, x_i) - f\|] &\leq C_2 \left(\frac{\log(m)}{m}\right)^2 |f|_2, \quad f \in H^2(0, L), \\ \mathbb{E}[\|\sigma(f, x_i) - f\|] &\leq C_2 \left(\frac{\log(m)}{m}\right)^4 |f|_4, \quad f \in H^4(0, L). \end{aligned} \tag{89}$$

Comparing with the $m^{-1/2}$ rate of convergence of gEDMD, we get

$$\begin{aligned} m^{\frac{1}{2}} \left(\frac{\log(m)}{m}\right)^2 &= \left(\frac{\log(m)}{m^{\frac{3}{4}}}\right)^2 \xrightarrow[m\to\infty]{} 0 \\ m^{\frac{1}{2}} \left(\frac{\log(m)}{m}\right)^4 &= \left(\frac{\log(m)}{m^{\frac{7}{8}}}\right)^4 \xrightarrow[m\to\infty]{} 0. \end{aligned} \tag{90}$$

Hence, for a given large number of samples m, one can expect more accuracy from direct interpolation methods than from gEDMD.

Remark 5.1. As we have mentioned in Section 4.6, in some applications the sampling points x^l can be chosen freely. Choosing them as a quadrature rule then drastically improves the convergence rates of gEDMD. In particular, [36] shows an exponential convergence rate using Gauss–Legendre quadrature, which is considerably better than the convergence rates of direct interpolation. Thus, in terms of performance with a given number of samples m, gEDMD with Gauss–Legendre quadrature would perform better than classical interpolation methods. However, we shall see (Remark 5.2) that it is still the more computationally complex method.

Moreover, recalling the computations of (82) and (86), exponential convergence of the integrals would also mean that, in order for the data-driven approximation error to be comparable to the finite element approximation error, m would have to be of the order of

$$\log(N_h^k) - (k+1)\log(h) \sim d\log(kL) - (d+k+1)\log(h) = d\log\left(\frac{kL}{h}\right) + (k+1)\log\left(\frac{1}{h}\right),$$

which is clearly linear in d for a fixed mesh size h. Hence, in that case, the drawback mentioned in Section 4.5 no longer holds: not only is the convergence of the data-driven approximation faster, but the required number of samples to achieve a certain accuracy is no longer subject to the curse of dimensionality.

To give a more thorough comparison, let $\varepsilon > 0$. We now compare the number m of samples required in each method in order to achieve an accuracy of at most ε.

For linear interpolation, the required number of samples satisfies:

$$\frac{\log(m)}{m} \sim \varepsilon^{\frac{1}{2}} \implies \varepsilon^{-\frac{1}{2}} \leq m \leq C\varepsilon^{-\frac{1}{2}+\epsilon}, \quad \forall \epsilon > 0.$$

For cubic spline interpolation, the required number of samples satisfies:

$$\frac{\log(m)}{m} \sim \varepsilon^{\frac{1}{4}} \implies \varepsilon^{-\frac{1}{4}} \leq m \leq C\varepsilon^{-\frac{1}{4}+\epsilon}, \quad \forall \epsilon > 0.$$

For modified gEDMD, first consider the Galerkin projection. From (53), h must satisfy:

$$h \leq C\varepsilon^{\frac{1}{2}}. \tag{91}$$

Recalling (86):

$$m \geq \left(C_h^1(f,\alpha)\frac{1+\frac{1}{h}}{\rho_{1,h}^-}\right)^2 c_h^1 h^{-4}$$

and using the estimates (A 12), (A 13), and (A 22) from Appendix A, one has:

$$\begin{aligned} m &\geq C'\left(C(\alpha)\|f\|_{L^\infty}h^{-2}\frac{1+\frac{1}{h}}{h}\right)^2\varepsilon^{-2} \\ &\geq C''C(\alpha)^2\|f\|_{L^\infty}^2 h^{-8}\varepsilon^{-2} \\ &\geq C'''C(\alpha)^2\|f\|_{L^\infty}^2\varepsilon^{-6}. \end{aligned} \tag{92}$$

Now, it is well-known that interpolation methods have linear complexity (see [66]). On the other hand, we have seen in Proposition 4.1 that the computational complexity of gEDMD is in $\mathcal{O}(mN^2+N^3)$. Regarding N, (91) implies

$$N \geq C'\varepsilon^{-\frac{1}{2}}.$$

Together with (92), this yields the following complexity in terms of ε:

$$\mathcal{O}(\varepsilon^{-7}+\varepsilon^{-\frac{3}{2}}) = \mathcal{O}(\varepsilon^{-7}). \tag{93}$$

We summarize the above in the following table:

	Parameters	Cost
Koopman operator method	h k m	ε^{-7}
Linear interpolation	m	$\varepsilon^{-(1/2)+\epsilon}$
Splines interpolation	m	$\varepsilon^{-(1/4)+\epsilon}$

This clearly shows that interpolation methods are more cost-eVective than gEDMD with linear finite element.

Remark 5.2. Recalling Remark 5.1, the exponential convergence of Gauss–Legendre quadratures positively impacts the computational complexity of gEDMD in small dimensions. Indeed, (92) becomes, for small ε,

$$\begin{aligned} m &\geq \log\left(\frac{1+\frac{1}{h}}{h}\right) - 2\log(h) \\ &\geq -4C\log(h) \\ &\geq -2C\log(\varepsilon), \end{aligned}$$

with a constant $C>0$. Thus the complexity of gEDMD becomes

$$\mathcal{O}(2C\log(\varepsilon^{-1})\varepsilon^{-1}+\varepsilon^{-\frac{3}{2}}),$$

which is much better than (93), but still does not compare favorably with interpolation methods.

5.2. *Numerical illustration*

We show the results of some implementations of gEDMD, focusing on the approximation of the vector field f. Given our remarks on the curse of dimensionality in Section 3.6, we focus on the case $d = 1$. We have given precise estimates in the previous section that show how the convergence of the data-driven approximation is aVected by the choice of the Galerkin projection. Given (53) and (86), this problem worsens in higher dimensions.

We take f given by combinations of trigonometric functions, polynomials, and exponentials on the interval $[0,1]$. We implement gEDMD with linear finite elements (see Appendix A) to recover an approximation of f. We compare the results with linear interpolation and spline interpolation on the same data sample.

5.2.1. *Data sample*

We form the data set:

$$\{x^n, f(x^n),\ n \in \{1,\dots,m\}\}. \tag{94}$$

To simplify matters we directly take the values of the nonlinearity f we have chosen, at the sample points x^n. In practice, f is unknown, so one obtains approximate values of f at the sample points by observing the trajectories of the system with initial conditions x^n, and approximating the initial velocities by finite diVerences (see Section 4 and [54]).

5.2.2. *gEDMD*

Compute the following matrices from the data:

$$\begin{aligned} G^m_{1,h} &:= (\psi_i(x_j))_{\substack{i\in\{1,\dots,N\}_h \\ j\in\{1,\dots,m\}}} \\ A^m_{1,h} &:= (\dot{x}_j \cdot \nabla\psi_i(x_j))_{\substack{i\in\{1,\dots,N\}_h \\ j\in\{1,\dots,m\}}}, \end{aligned} \tag{95}$$

and define

$$R^m_{1,h} = \frac{1}{m} G^m_{1,h} (A^m_{1,h})^\top.$$

Perform the linear least squares regression:

$$K^m_{1,h} := \operatorname{argmin}_{A\in\mathbb{R}^{N\times N}} \|M^1_h A - R^m_{1,h}\|^2 = (M^m_{1,h})^{-1} R^m_{1,h}. \tag{96}$$

5.2.3. *Identifying the nonlinearity in the diVerential equation*

Now consider the identity function $g(x) = x$: $g \in \mathbb{Q}_1$ as it is linear, and

$$g(x) = \sum_{j=0}^{M} j_h \psi_j(x) = c^\top \psi(x), \quad \forall x \in [0,L]. \tag{97}$$

To recover f, we compute

$$\tilde{c}^\top := c^\top K^m_{1,h} \tag{98}$$

to get the following approximation of f on the interval $[0,L]$:

$$\tilde{f}^m_{1,h}(x) = \tilde{c}^\top \psi(x) = \sum_{j=0}^{M} \tilde{c}_j \psi_j(x), \quad \forall x \in [0,L]. \tag{99}$$

5.3. *Numerical simulations*

Figure 1 shows approximations of f (given by the thicker black curve) computed with gEDMD for a fixed dimension of the finite element space, and diVerent numbers of samples m. One can see that for a small number of samples, there are sizeable discrepancies. For a larger number of samples the curves obtained by gEDMD seem to match the curve rather well.

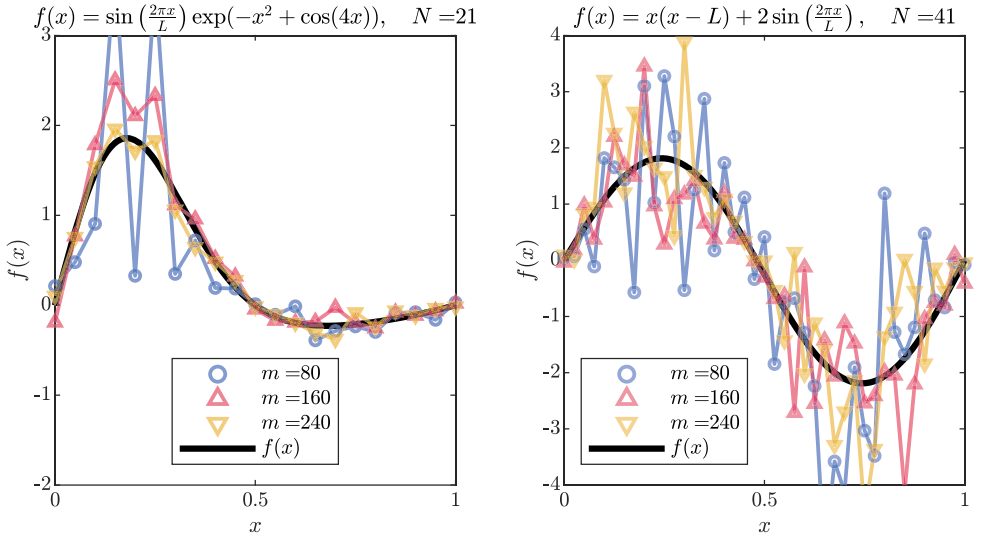

Figure 1. Identification of diVerent functions using gEDMD with linear finite elements.

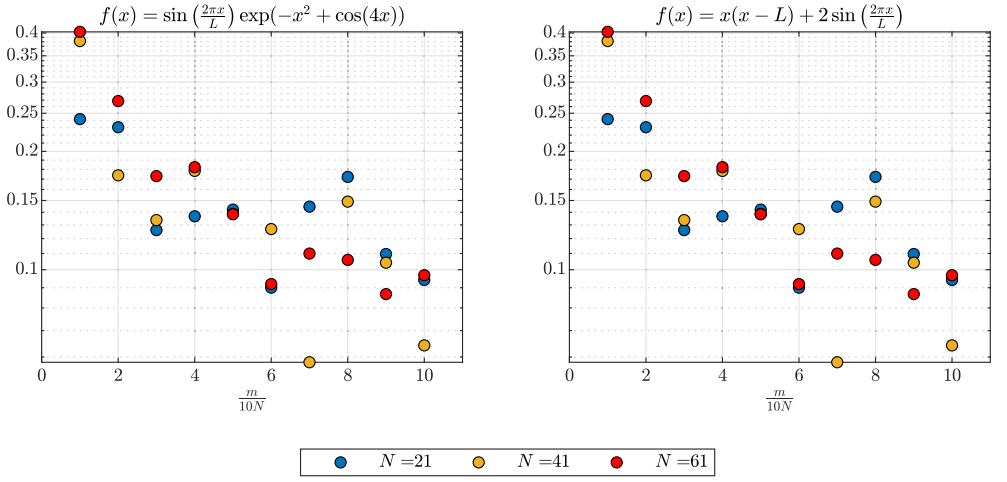

Figure 2. Relative error of gEDMD with linear elements for diVerent values of N. Dependence on the number m of samples.

In Figure 2, we plot the relative approximation error

$$\frac{\|\tilde{f}_N^m - f\|}{\|f\|}$$

as a function of the number of samples m, for several values of N: as expected, increasing the number of samples decreases the error. This decrease appears to be slower for greater values of N (see the red dots in the figure below), which is consistent with (86).

Finally, in Figure 3 we compare the relative approximation error of gEDMD with that of direct interpolation methods. We implement gEDMD for increasing values of N, with $m = 100N$ samples. We compare the results with linear interpolation and spline interpolation with m random samples. There is a clear hierarchy between the three methods, gEDMD being the least accurate by far.

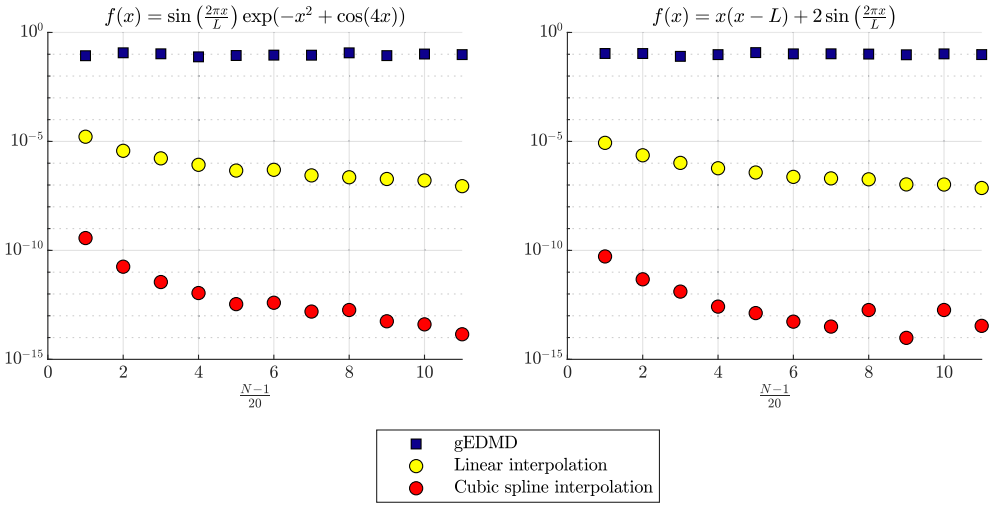

Figure 3. Comparison with direct interpolation methods. $m = 100(N-1)$.

6. Conclusion

In this article, we have presented the implementation of a *system identification* method, based on finite-dimensional, data-driven approximations $\mathscr{K}_N^m$ of the so-called *Koopman operator*, which is the transport operator associated to the unknown vector field f:

$$f(x)\cdot\nabla.$$

From trajectory data on the system

$$\dot{x} = f(x),$$

measured at randomly drawn sample points $(x^n)_{n\in\{1,\dots,m\}}$ one can compute the sample values

$$f(x^n), \quad x^n \in \Omega,\ n \in \{1,\dots,m\}.$$

From this data set, the method provides an approximation of the vector field f and its flow Φ_f^t, using the coordinate functions $g_i(x) = x_i$:

$$f_i \approx \mathscr{K}_N^m g_i, \quad (\Phi_f^t)_i \approx \mathrm{e}^{t\mathscr{K}_N^m} g_i.$$

The potential advantage of this method lies in the *linearity* of the Koopman operator: as can be seen above, the nonlinear flow Φ_f^t is approximated thanks to the linear flow $\mathrm{e}^{t\mathscr{K}_N^m}$, which is however defined on a subspace of functions rather than the state space $\mathscr{Q} \subset \mathbb{R}^d$.

We have focused on the case of Galerkin projections of the Koopman operator on finite element spaces. They are generic approximation spaces, widely used in numerical analysis to approximate PDEs. In this case, the approximating properties of these spaces guarantee that the corresponding Galerkin projection of the Koopman operator is a reliable approximation, without having to rely on hypothetical spectral properties. Indeed the Koopman operator (which is essentially a transport operator) does not have nice spectral properties in general.

For one-dimensional problems, we have compared this method with *direct interpolation* of the vector field f from the sample values

$$f(x^n), \quad x^n \in [0,L],\ n \in \{1,\dots,m\}.$$

Interpolation yields an approximation $\tilde{f}$ of f. Predictions can then be made by integrating the nonlinear diVerential equation

$$\dot{x} = \tilde{f}(x).$$

We have found that this Koopman operator-based system identification method is generally not advantageous in practice, due to two drawbacks.

(1) **Dimension of the Galerkin projections**. With Galerkin projections on classical approximation spaces, high accuracy requires high-dimensional spaces of functions. This drawback, due to the well-known *curse of dimensionality*, is a recurring feature of approximation spaces. On these spaces, the approximation of the linear transport equation associated to the Koopman operator then comes at a high computational cost.

(2) **Data-driven approximation of the Galerkin projections**. Galerkin projection on finite element spaces provide accurate, albeit high-dimensional, approximations of the Koopman operator for system identification and predictions. However, for the system identification problem we have formulated, these projections must be recovered from a finite amount of data. To that eVect, we have brought a novel modification to the usual gEDMD algorithm. As we are working with classical finite element spaces, the mass matrix can be computed o Z ine. Then, recovering the Galerkin projection of the Koopman operator reduces to approximating the stiVness matrix from data. To approximate the integrals in the stiVness matrix, given the nature of the data set, the natural path is to use Monte-Carlo integration (as is the case for regular EDMD in the literature).

    In addition to the inherent slowness of Monte-Carlo integration, we furthermore observe that the rate of convergence of the data-driven approximation has multiple dependencies in the parameters h (mesh size) and k (degree) of the finite element space. As a consequence, the more accurate the Galerkin projection we take, the harder it becomes to approximate from data by the gEDMD method.

    We have quantified this precisely in 1D, which allows for a clear comparison between recovery of the vector field f by the gEDMD method and direct interpolation (linear or cubic splines) of f.

We have based our analysis on a specific choice of approximation space, but our estimates illustrate what is probably a general phenomenon: the Galerkin projection of the Koopman operator on generic approximation spaces is prohibitively slow to approximate from data.

Let us conclude with some prospects that could be explored to overcome these drawbacks:

6.1. *Quasi-Monte-Carlo integration and quadrature rules*

As mentioned in Section 4, we have chosen to focus our quantitative analysis on the classical implementation of EDMD, which uses Monte-Carlo integration to approximate integrals. We have listed quadrature rules as a possible improvement in cases where the sampling points can be chosen freely (at least in low dimension, see [36]). In higher dimension, another possibility would be to use Quasi-Monte-Carlo integration (see [58]), which also requires to have some degree of control on how the sampling points are chosen.

Note however, that the eVect of taking large finite element spaces on the convergence of the integrals (whether computed by quadrature, Monte-Carlo or Quasi-Monte-Carlo), which we have noted in Section 4.5, would still have to be taken into account.

6.2. *Learning dictionaries to find better Galerkin projections*

Given our remarks on the curse of dimensionality, another major point of improvement in Galerkin approximations of the Koopman operator is to find function spaces that are optimal for a given system. In other words, a promising perspective would be to be able to learn appropriate bases of functions (dictionaries) (ψ_j) from data. In [36], a reproducing kernel method is used

to handle higher-dimensional applications. This allows the authors to compute an appropriate dictionary (ψ_j) from data. On the space spanned by this dictionary, the Galerkin projection of $\mathscr{K}$ can be computed using gEDMD. Although this is designed for the spectral study of $\mathscr{K}$, it appears as a promising improvement of gEDMD for system identification as well. It would be of interest to gain a deep understanding of how the kernel should be chosen, but one must keep in mind that this restricts the method to Galerkin projections of $\mathscr{K}$ on spaces that are indeed Reproducing Kernel Hilbert Spaces. These specific spaces might not always be relevant for applications.

6.3. *Galerkin projections of low dimension*

In addition to finding relevant dictionaries for Galerkin methods, it would be of interest to find lower-dimensional Galerkin projections, which still allow for reliable estimates as in Proposition 3.2 with finite element spaces. Considering that classical interpolation methods (or SINDy with sparsity constraints) provide a very good approximation of the vector field, the goal for the gEDMD method is to provide a numerically interesting approximation of the flow Φ_f^t in order to make predictions. This will be the case if the Galerkin projection is of low dimension.

Finding such spaces would also reduce the impact of the choice of the Galerkin projection on the Monte-Carlo convergence, which in addition to being inherently slow, is further slowed down when taking higher-dimensional finite element spaces.

It seems very likely that there are many situations where this is not possible (chaotic systems are a frequently cited example [62,67]). Between the universal solution of classical approximation spaces, and the ideal situation of subspaces of functions that are invariant under the Koopman operator (see [26]), the complete picture is still missing: what characterizes the existence of these relevant low-dimensional Galerkin projections? How "rare" is it? Is there a data-driven manner to determine whether they exist and how to find them?

6.4. *Use of deep neural networks*

Finally, let us point out that there have been many recent attempts to find such low-dimensional Galerkin projections, using deep neural networks [30, 43, 44, 46, 68]. The output of such networks is a so-called *dictionary*, that is, a basis of functions $\psi_1, \dots, \psi_N$, where N is fixed beforehand. These functions are taken from a large class of functions (essentially given by the architecture of the network) and the network is trained to minimize the prediction error made when performing EDMD with the subspace $V_N = \mathrm{Span}\{\psi_1, \dots, \psi_N\}$, while additionally penalizing situations where V_N does not contain the g_i, for the purpose of system identification (see Section 4.5). Although numerical implementations yield promising results, there is still much work to be done to interpret the outputs of these deep neural networks. Moreover, as is mentioned in [30], there is no guarantee that the output is not a local minimizer.

Conflicts of interest

Authors have no conflict of interest to declare.

Acknowledgements

We would like to thank Professor Lars Grüne for fruitful discussions on this topic.

We are greatly indebted to Jesus Oroya Villalta, for his help in the numerical experiments and for his astute comments.

This work has received funding from the Alexander von Humboldt-Professorship program. The work of EZ is partially funded by the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 765579-ConFlex, the Grant MTM2017-92996-C2-1-R COS-NET of MINECO (Spain), the Air Force O Y ce of Scientific Research (AFOSR) under Award No. FA9550-18-1-0242 and the Transregio 154 Project "Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks" of the German DFG.

Appendix A. Linear 1D finite elements

We fix a mesh size

$$h := \frac{1}{M}, \quad M \in \mathbb{N}^*, \tag{A 1}$$

we define the mesh

$$\mathcal{T}_h = \{jh,\ j \in \{0,\dots,M\}\}, \tag{A 2}$$

which is a subdivision of $[0,1]$. We define the finite elements of degree 1:

$$V_h^1 := \{v \in C^0([0,1]),\ v_{|[jh,(j+1)h]} \text{ is a Y ne},\ j \in \{0,\dots,M-1\}\}. \tag{A 3}$$

We define

$$N := \dim V_h^1 = M+1 = \frac{1}{h}+1. \tag{A 4}$$

We denote by ψ_j, $j \in \{0,\dots,M\}$ the node functions, which satisfy

$$\psi_j(kh) = \delta_{jk}, \quad j,k\{0,\dots,M\}, \tag{A 5}$$

they form a basis of V_h^1.

Explicitly, for a given node jh, ψ_j is defined by (see also Figure 4):

$$j \in \{1,\dots,M-1\}, \quad \psi_j(x) = \begin{cases} 0 & \text{on } [-L,(j-1)h] \cup [(j+1)h,L] \\ \dfrac{x-(j-1)h}{h} & \text{on } [(j-1)h,jh] \\ \dfrac{(j+1)h-x}{h} & \text{on } [jh,(j+1)h] \end{cases} \tag{A 6}$$

$$\psi_0(x) = \begin{cases} \dfrac{h-x}{h} & \text{on } [0,h] \\ 0 & \text{on } [h,L] \end{cases} \tag{A 7}$$

$$\psi_M(x) = \begin{cases} 0 & \text{on } [-L,L-h] \\ \dfrac{x-L+h}{h} & \text{on } [L-h,L] \end{cases} \tag{A 8}$$

Classically, the mass matrix is given by:

$$M_h^1 = h \begin{pmatrix} 1/3 & 1/6 & 0 & \cdots & \cdots & \cdots & 0 \\ 1/6 & 2/3 & 1/6 & 0 & \ddots & \cdots & \vdots \\ 0 & 1/6 & 2/3 & 1/6 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1/6 & 2/3 & 1/6 \\ 0 & \cdots & \cdots & \cdots & 0 & 1/6 & 1/3 \end{pmatrix} \tag{A 9}$$

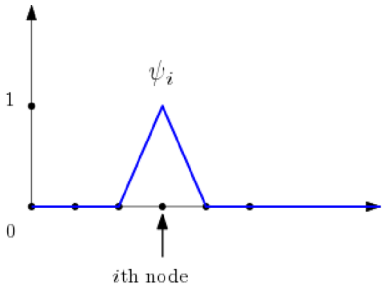

Figure 4. The shape function ψ_j.

and its eigenvalues are given by (after a slight modification of [69, Exercise 7.4.1]):

$$\lambda_j = \frac{h}{3}\left(2+\cos\left(\frac{jh\pi}{1+2h}\right)\right), \quad j \in \{1,\dots,N\}. \tag{A 10}$$

Consequently,

$$\rho^+_{1,h} = \frac{h}{3}\left(2+\cos\left(\frac{\pi h}{1+2h}\right)\right), \tag{A 11}$$

$$\rho^-_{1,h} = \frac{h}{3}\left(2+\cos\left(\frac{N\pi h}{1+2h}\right)\right) \underset{h\to 0}{\sim} 0, \tag{A 12}$$

and

$$c^1_h = \frac{\rho^+_{1,h}}{\rho^-_{1,h}} = \frac{2+\cos\left(\frac{\pi h}{1+2h}\right)}{2+\cos\left(\frac{\pi(h+1)}{1+2h}\right)} \xrightarrow[h\to 0]{} 3. \tag{A 13}$$

Let us now give estimates on the variances of the coefficients of the rigidity matrix, which is involved in the convergence of the Monte-Carlo method in Section 4.3. First, an obvious remark is that

$$\sigma^{k,h}_{ij} = 0, \quad |i-j|>1. \tag{A 14}$$

On the other hand, assuming that $f \in C^2(\mathcal{Q})$: for $i \in \{0,\dots,M-1\}$,

$$\begin{aligned}\int_\Omega (\psi_i(x) f(x) \psi'_{i+1}(x))^2\,dx - \frac{f(x_i)^2}{h^2}\int_{x_i}^{x_{i+1}} \psi_i(x)^2\,dx &= \frac{1}{h^2}\int_{x_i}^{x_{i+1}} \psi_i(x)^2 (f(x)^2 - f(x_i)^2)\,dx \\ &\le \|f'\|_\infty \frac{C_1}{h},\end{aligned} \tag{A 15}$$

for some constant $C_1>0$. Similarly,

$$\begin{aligned}\int_\Omega (\psi_{i+1}(x) f(x) \psi'_i(x))^2\,dx - \frac{f(x_i)^2}{h^2}\int_{x_i}^{x_{i+1}} \psi_{i+1}(x)^2\,dx &= \frac{1}{h^2}\int_{x_i}^{x_{i+1}} \psi_{i+1}(x)^2 (f(x)^2 - f(x_i)^2)\,dx \\ &\le \|f'\|_\infty \frac{C_2}{h},\end{aligned} \tag{A 16}$$

and

$$\left(\int_\Omega \psi_i(x) f(x) \psi'_{i+1}(x)\,dx\right)^2 - \frac{f(x_i)^2}{h^2}\left(\int_{x_i}^{x_{i+1}} \psi_i(x)\,dx\right)^2 \le \|f'\|_\infty \frac{C_3}{h}, \tag{A 17}$$

$$\left(\int_\Omega \psi_{i+1}(x) f(x) \psi'_i(x)\,dx\right)^2 - \frac{f(x_i)^2}{h^2}\left(\int_{x_i}^{x_{i+1}} \psi_{i+1}(x)\,dx\right)^2 \le \|f'\|_\infty \frac{C_4}{h}, \tag{A 18}$$

for some constants $C_2, C_3, C_4 > 0$.

Putting (A 15) and (A 17) together, we get, for $i \in \{0,\dots,M-1\}$,

$$(\sigma^{k,h}_{i,i+1})^2 = \frac{1}{3h^2} f(x_i)^2 + \mathcal{O}\left(\frac{1}{h}\right), \tag{A 19}$$

and similarly with (A 16) and (A 18),

$$(\sigma^{k,h}_{i+1,i})^2 = \frac{1}{3h^2} f(x_i)^2 + \mathcal{O}\left(\frac{1}{h}\right), \tag{A 20}$$

and, for $i \in \{0,\dots,M\}$,

$$(\sigma^{k,h}_{ii})^2 = \frac{2}{3h^2} f(x_i)^2 + \mathcal{O}\left(\frac{1}{h}\right). \tag{A 21}$$

Recalling (67), this yields

$$C^1_h(f,\alpha) = \frac{2}{3h^2} \| f \|^2_\infty + \mathcal{O}\left(\frac{1}{h}\right). \tag{A 22}$$

References

[1] L. Lennart, *System Identification: Theory for the User*, PTR Prentice Hall, Upper Saddle River, NJ, 1999, 1-14 pages.

[2] B. Bamieh, L. Giarré, "Identification of linear parameter varying models", *Int. J. Robust Nonlinear Control* **12** (2002), no. 9, p. 841-853.

[3] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*, Springer Science & Business Media, 2013.

[4] A. P. Calderón, "On an inverse boundary value problem", *Comput. Appl. Math.* **25** (2006), no. 2–3, p. 133-138.

[5] V. Isakov, *Inverse Problems for Partial DiVerential Equations*, vol. 127, Springer, 2006.

[6] M. Yamamoto, "Carleman estimates and an inverse heat source problem for the thermoelasticity system", *Inverse Probl.* **27** (2010), no. 1, article no. 015006.

[7] A. M. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes*, F. Didot, 1805.

[8] P. Binev, A. Cohen, W. Dahmen, R. DeVore, V. Temlyakov, "Universal algorithms for learning theory part I: Piecewise constant functions", *J. Mach. Learn. Res.* **6** (2005), p. 1297-1321.

[9] P. Binev, A. Cohen, W. Dahmen, R. DeVore, "Universal algorithms for learning theory. Part II: Piecewise polynomial functions", *Constr. Approx.* **26** (2007), no. 2, p. 127-152.

[10] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*, MIT Press, 2016, http://www.deeplearningbook.org.

[11] A. Quarteroni, A. Valli, *Numerical Approximation of Partial DiVerential Equations*, Springer Series in Computational Mathematics, vol. 23, Springer-Verlag, Berlin, 1994.

[12] S. L. Brunton, J. L. Proctor, J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems", *Proc. Natl. Acad. Sci. USA* **113** (2016), no. 15, p. 3932-3937.

[13] K. Champion, B. Lusch, J. Nathan Kutz, S. L. Brunton, "Data-driven discovery of coordinates and governing equations", *Proc. Natl. Acad. Sci. USA* **116** (2019), no. 45, p. 22445-22451.

[14] D. A. Messenger, D. M. Bortz, "Weak SINDy: Galerkin-based data-driven model selection", *Multiscale Model. Simul.* **19** (2021), no. 3, p. 1474-1497.

[15] R. J. DiPerna, P. L. Lions, "Ordinary diVerential equations, transport theory and Sobolev spaces", *Invent. Math.* **98** (1989), no. 3, p. 511-547.

[16] M. Hauray, C. Le Bris, P.-L. Lions, "Deux remarques sur les flots généralisés d'équations diVérentielles ordinaires", *C. R. Math.* **344** (2007), no. 12, p. 759-764.

[17] B. O. Koopman, "Hamiltonian systems and transformation in Hilbert space", *Proc. Natl. Acad. Sci. USA* **17** (1931), no. 5, p. 315-318.

[18] J. v. Neumann, "Proof of the quasi-ergodic hypothesis", *Proc. Natl. Acad. Sci. USA* **18** (1932), no. 1, p. 70-82.

[19] B. O. Koopman, J. V. Neumann, "Dynamical systems of continuous spectra", *Proc. Natl. Acad. Sci. USA* **18** (1932), no. 3, p. 255-263.

[20] I. Mezić, "Spectral properties of dynamical systems, model reduction and decompositions", *Nonlinear Dyn.* **41** (2005), no. 1, p. 309-325.

[21] I. Mezić, "Analysis of fluid flows via spectral properties of the Koopman operator", *Annu. Rev. Fluid Mech.* **45** (2013), no. 1, p. 357-378.

[22] Y. Lan, I. Mezić, "Linearization in the large of nonlinear systems and Koopman operator spectrum", *Physica D* **242** (2013), no. 1, p. 42-53.

[23] A. Mauroy, I. Mezic, "Global stability analysis using the eigenfunctions of the Koopman operator", *IEEE Trans. Automat. Contr.* **61** (2016), no. 11, p. 3356-3369.

[24] I. Mezić, "Spectrum of the Koopman operator, spectral expansions in functional spaces, and state-space geometry", *J. Nonlinear Sci.* **30** (2019), p. 2091-2145.

[25] M. D. Kvalheim, S. Revzen, "Existence and uniqueness of global Koopman eigenfunctions for stable fixed points and periodic orbits", *Physica D* **425** (2021), article no. 132959.

[26] S. L. Brunton, B. W. Brunton, J. L. Proctor, J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control", *PLoS One* **11** (2016), no. 2, p. 1-19.
[27] M. Korda, I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control", *Automatica* **93** (2018), p. 149-160.
[28] S. Klus, F. Nüske, S. Peitz, J.-H. Niemann, C. Clementi, C. Schütte, "Data-driven approximation of the Koopman generator: Model reduction, system identification, and control", *Physica D* **406** (2020), article no. 132416.
[29] T. Carleman, "Application de la théorie des équations intégrales linéaires aux systèmes d'équations diVérentielles non linéaires", *Acta Math.* **59** (1932), p. 63-87.
[30] N. Takeishi, Y. Kawahara, T. Yairi, "Learning Koopman invariant subspaces for dynamic mode decomposition", in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett, eds.), Curran Associates, Inc., 2017, p. 1130-1140.
[31] J. Slipantschuk, O. F. Bandtlow, W. Just, "Dynamic mode decomposition for analytic maps", *Commun. Nonlinear Sci. Numer. Simul.* **84** (2020), article no. 105179.
[32] E. Kaiser, J. N. Kutz, S. L. Brunton, "Data-driven discovery of Koopman eigenfunctions for control", *Mach. Learn.: Sci. Technol.* **2** (2021), no. 3, article no. 035023.
[33] M. Korda, I. Mezić, "Optimal construction of Koopman eigenfunctions for prediction and control", *IEEE Trans. Automat. Contr.* **65** (2020), no. 12, p. 5114-5129.
[34] N. Govindarajan, R. Mohr, S. Chandrasekaran, I. Mezic, "On the approximation of Koopman spectra for measure preserving transformations", *SIAM J. Appl. Dyn. Syst.* **18** (2019), no. 3, p. 1454-1497.
[35] M. Korda, M. Putinar, I. Mezić, "Data-driven spectral analysis of the Koopman operator", *Appl. Comput. Harmon. Anal.* **48** (2020), no. 2, p. 599-629.
[36] M. J. Colbrook, A. Townsend, "Rigorous data-driven computation of spectral properties of Koopman operators for dynamical systems", 2021, preprint, https://arxiv.org/abs/2111.14889.
[37] M. O. Williams, I. G. Kevrekidis, C. W. Rowley, "A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition", *J. Nonlinear Sci.* **25** (2015), no. 6, p. 1307-1346.
[38] M. Korda, I. Mezić, "On convergence of extended dynamic mode decomposition to the Koopman operator", *J. Nonlinear Sci.* **28** (2018), no. 2, p. 687-710.
[39] B. Lusch, J. Nathan Kutz, S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics", *Nat. Commun.* **9** (2018), no. 1, p. 1-10.
[40] S. Klus, F. Nüske, B. Hamzi, "Kernel-based approximation of the Koopman generator and Schrödinger operator", *Entropy* **22** (2020), no. 7, article no. 722.
[41] S. Kmiecik, D. Gront, M. Kolinski, L. Wieteska, A. E. Dawid, A. Kolinski, "Coarse-grained protein models and their applications", *Chem. Rev.* **116** (2016), no. 14, p. 7898-7936.
[42] H. Arbabi, M. Korda, I. Mezić, "A data-driven Koopman model predictive control framework for nonlinear partial diVerential equations", in *2018 IEEE Conference on Decision and Control (CDC)*, 2018, p. 6409-6414.
[43] Z. Ping, Z. Yin, X. Li, Y. Liu, T. Yang, "Deep Koopman model predictive control for enhancing transient stability in power grids", *Int. J. Robust Nonlinear Control* **31** (2021), no. 6, p. 1964-1978.
[44] M. Korda, Y. Susuki, I. Mezić, "Power grid transient stabilization using Koopman model predictive control", *IFAC-PapersOnLine* **51** (2018), no. 28, p. 297-302, 10th IFAC Symposium on Control of Power and Energy Systems CPES 2018.
[45] S. Otto, C. Rowley, "Linearly recurrent autoencoder networks for learning dynamics", *SIAM J. Appl. Dyn. Syst.* **18** (2019), no. 1, p. 558-593.
[46] E. Yeung, S. Kundu, N. Hodas, "Learning deep neural network representations for Koopman operators of nonlinear dynamical systems", in *2019 American Control Conference (ACC)*, 2019, p. 4832-4839.
[47] A. Mauroy, J. Goncalves, "Koopman-based lifting techniques for nonlinear systems identification", *IEEE Trans. Automat. Contr.* **65** (2020), no. 6, p. 2550-2565.
[48] R. Bellman, *Dynamic Programming*, 1st ed., Princeton University Press, Princeton, NJ, USA, 1957.
[49] R. A. DeVore, "Nonlinear approximation", *Acta Numer.* **7** (1998), p. 51-150.
[50] R. J. Kunsch, "Breaking the curse for uniform approximation in Hilbert spaces via Monte Carlo methods", *J. Complex.* **48** (2018), p. 15-35.
[51] E. Novak, K. Ritter, "The curse of dimension and a universal method for numerical integration", in *Multivariate Approximation and Splines*, Springer, 1997, p. 177-187.
[52] H.-J. Bungartz, M. Griebel, "Sparse grids", *Acta Numer.* **13** (2004), p. 147-269.
[53] A. Chkifa, A. Cohen, C. Schwab, "Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs", *J. Math. Pures Appl.* **103** (2015), no. 2, p. 400-428.
[54] R. Chartrand, "Numerical diVerentiation of noisy, nonsmooth, multidimensional data", in *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2017, p. 244-248.
[55] M. O. Williams, M. S. Hemati, S. T. M. Dawson, I. G. Kevrekidis, C. W. Rowley, "Extending data-driven Koopman analysis to actuated systems", *IFAC-PapersOnLine* **49** (2016), no. 18, p. 704-709.

[56] S. Klus, P. Koltai, C. Schütte, “On the numerical approximation of the Perron–Frobenius and Koopman operator”, *J. Comput. Dyn.* **3** (2016), no. 1, p. 51-79.
[57] P.-N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, 1st ed., Addison-Wesley Longman Publishing Co., Inc., USA, 2005.
[58] R. E. Caflisch, “Monte Carlo and quasi-Monte Carlo methods”, *Acta Numer.* **7** (1998), p. 1-49.
[59] R. Penrose, “A generalized inverse for matrices”, *Math. Proc. Cambridge Philos. Soc.* **51** (1955), no. 3, p. 406-413.
[60] A. Tonge, “Equivalence constants for matrix norms: a problem of Goldberg”, *Linear Algebra Appl.* **306** (2000), no. 1, p. 1-13.
[61] D. Giannakis, “Data-driven spectral decomposition and forecasting of ergodic dynamical systems”, *Appl. Comput. Harmon. Anal.* **47** (2019), no. 2, p. 338-396.
[62] H. Arbabi, I. Mezić, “Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator”, *SIAM J. Appl. Dyn. Syst.* **16** (2017), no. 4, p. 2096-2126.
[63] A. G. Kachurovskii, “The rate of convergence in ergodic theorems”, *Russ. Math. Surv.* **51** (1996), no. 4, p. 653-703.
[64] C. A. Hall, W. Weston Meyer, “Optimal error bounds for cubic spline interpolation”, *J. Approx. Theory* **16** (1976), no. 2, p. 105-122.
[65] R. Pyke, “Spacings (with discussion)”, *J. Roy. Statist. Soc. Ser. B* **27** (1965), p. 395-449, https://www.jstor.org/stable/2345793.
[66] K. Toraichi, K. Katagishi, I. Sekita, R. Mori, “Computational complexity of spline interpolation”, *Int. J. Syst. Sci.* **18** (1987), no. 5, p. 945-954.
[67] H. Arbabi, T. Sapsis, “Generative stochastic modeling of strongly nonlinear flows with non-Gaussian statistics”, 2020, preprint, https://arxiv.org/abs/1908.08941.
[68] Y. Xiao, X. Zhang, X. Xu, X. Liu, J. Liu, “A deep learning framework based on Koopman operator for data-driven modeling of vehicle dynamics”, 2020, preprint, https://arxiv.org/abs/2007.02219.
[69] G. Allaire, *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation*, Oxford University Press, 2007.