
# Data-driven analysis of complex dynamical systems

**Dr. Stefan Klus**

Habilitationsschrift
Fachbereich Mathematik und Informatik

## Acknowledgements

I would like to thank the co-authors of the publications that constitute the major part of this thesis for many fruitful collaborations, in particular Christof Schütte, Michael Dellnitz, Frank Noé, Ioannis Kevrekidis, Péter Koltai, Patrick Gelß, Andreas Bittracher, Ralf Banisch, Sebastian Peitz, Feliks Nüske, and Hao Wu. This research has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114 *Scaling Cascades in Complex Systems*.

# Contents

*The goal is to transform data into information, and information into insight.*
Carly Fiorina, former CEO of Hewlett–Packard

# 1 Introduction

The main focus of this thesis is the data-driven analysis of complex dynamical systems. Although we will consider mainly molecular dynamics and fluid dynamics problems, the various methods presented in the following chapters can be applied to arbitrary dynamical systems. In fact, in order to apply these methods, no a priori knowledge about the system is required, only simulation or measurement data. Such data-driven methods got a lot of attention recently due to the availability of large data sets. Gaining insight into the characteristic properties of a system by analyzing such data sets is akin to the metaphorical search for a needle in a haystack. The goal of data-driven methods is to extract relevant information about global properties of the underlying system, whose governing equations might be unknown. Global information can be obtained by analyzing the eigenvalues and eigenfunctions of transfer operators associated with the system. Examples of such operators are the Perron–Frobenius operator and the Koopman operator, which are adjoint to each other in appropriately defined function spaces. The Perron–Frobenius operator describes the evolution of densities, whereas the Koopman operator describes the evolution of observables. For a discrete dynamical system $\Phi\colon \mathbb{X} \to \mathbb{X}$, with $\mathbb{X} \subset \mathbb{R}^d$, the Koopman operator is simply defined by

$$\mathcal{K}f = f \circ \Phi,$$

where $f : \mathbb{X} \to \mathbb{R}$ is an observable of the system. The definition can be extended in a straightforward fashion to continuous dynamical systems. That is, instead of analyzing a highly nonlinear but finite-dimensional dynamical system, we consider a linear but infinite-dimensional operator associated with the system. A pictorial representation of the relationship between a dynamical system $\Phi$ and the Koopman operator $\mathcal{K}$ is shown in Figure 1.1, see also [195]. Since we cannot handle infinite-dimensional problems numerically, the infinite-dimensional operator is typically projected onto a finite-dimensional space spanned by a given set of basis functions.

A popular data-driven method for the analysis of high-dimensional data sets is *dynamic mode decomposition* (DMD). It was shown that DMD is related to the Koopman

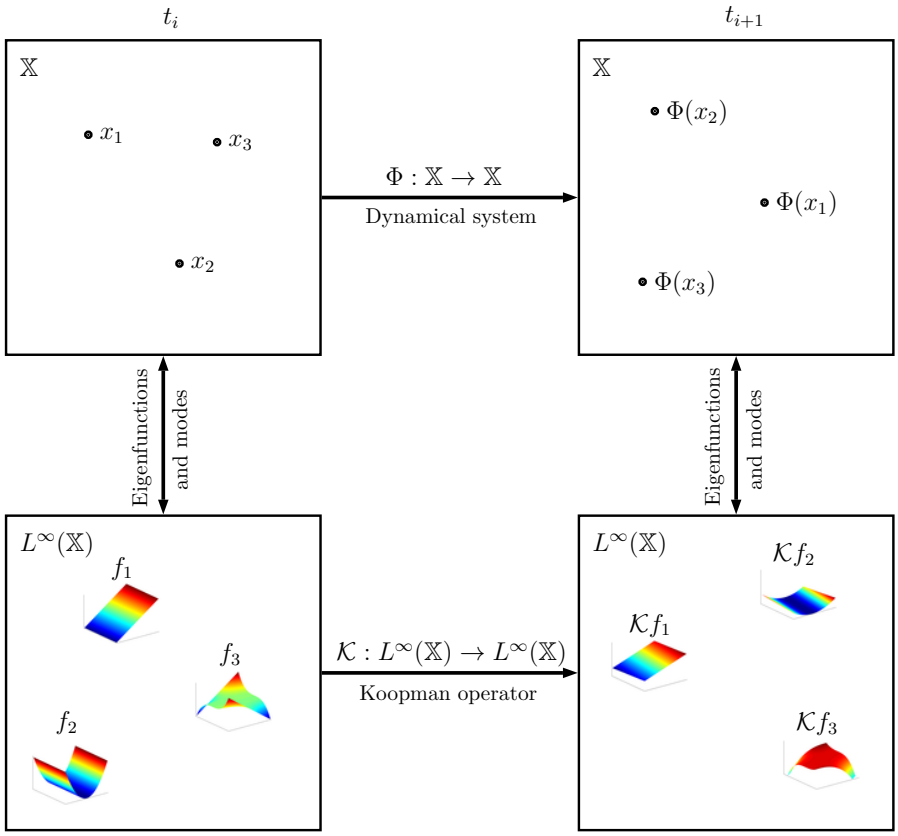



**Figure 1.1:** Pictorial representation of the Koopman operator approach based on [195]. The top path maps points $x_i \in \mathbb{X}$ to points $\Phi(x_i) \in \mathbb{X}$, the bottom path maps observables $f_i \in L^\infty(\mathbb{X})$ to functions $\mathcal{K}f_i \in L^\infty(\mathbb{X})$. The connection between states and observables is given by the spectral decomposition of the Koopman operator. The action of the dynamical system $\Phi$ can be rewritten in terms of eigenvalues, eigenmodes, and eigenfunctions.

operator, see, e.g., [159, 22]. A generalization of this method, called *extended dynamic mode decomposition* (EDMD), allows for more accurate approximations of the Koopman operator and, as shown in Chapter 2, also of the Perron–Frobenius operator. The data is in general given by two matrices

$$X = \begin{bmatrix} x_1 & \dots & x_m \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} y_1 & \dots & y_m \end{bmatrix},$$

where $y_i = \Phi(x_i)$. That is, $X, Y \in \mathbb{R}^{d \times m}$. The matrices $X$ and $Y$ can be extracted from one long trajectory. For a continuous dynamical system, $\Phi$ is then the flow map $\Phi^\tau$ for a fixed lag-time $\tau$, i.e., $x_i = x(t_i)$ and $y_i = \Phi^\tau(x_i) = x(t_i + \tau)$. In addition to the simulation or measurement data itself, a set of basis functions $\{\psi_1, \dots, \psi_k\}$ – also called *dictionary* – must be provided. This is typically written as a vector-valued function $\psi : \mathbb{X} \to \mathbb{R}^k$. We

then obtain the transformed data matrices

$$\Psi_X = \begin{bmatrix} \psi(x_1) & \dots & \psi(x_m) \end{bmatrix} \quad \text{and} \quad \Psi_Y = \begin{bmatrix} \psi(y_1) & \dots & \psi(y_m) \end{bmatrix},$$

with $\Psi_X, \Psi_Y \in \mathbb{R}^{k \times m}$. For DMD, only the data matrices $X$ and $Y$ are used, whereas EDMD is based on the transformed data matrices $\Psi_X$ and $\Psi_Y$. Similar methods have also been developed by the molecular dynamics community under the names *time-lagged independent component analysis* (TICA) and *variational approach of conformation dynamics* (VAC). The relationships between these methods will be described in detail in Chapter 3.

There are two typical scenarios: We have one long trajectory or many short trajectories for a system with a comparably low-dimensional state space, i.e., $m \gg d$ or $m \gg k$, respectively. Alternatively, we sometimes want to consider systems with a high-dimensional state space but only few snapshots, i.e., $d \gg m$, or a large set of basis functions is required in order to resolve the dynamics accurately so that $k \gg m$. The former is, for instance, the case for molecules comprising only a small number of atoms, the latter for fluid dynamics problems, where the state space is often discretized using a regular grid. Based on the use case, different algorithms have been developed to compute the eigenvalues, eigenfunctions, and eigenmodes efficiently. This is also the main reason why the tensor-based methods proposed in Chapter 4 for molecular dynamics problems and Chapter 5 for fluid dynamics problems are rather different. A detailed description and comparison of data-driven methods can be found in Chapter 3. For a more detailed introduction of transfer operators and methods to obtain finite-dimensional approximations, we refer to Chapter 2 and Chapter 3. A kernel-based version of EDMD, proposed in [196], is also briefly described in Chapter 2.

We are particularly interested in so-called *almost invariant sets* or *meta-stable sets* of dynamical systems. In the molecular dynamics setting, these sets correspond to different conformations of a molecule. A frequently used illustrating example is the $n$-butane molecule $H_3C{-}CH_2{-}CH_2{-}CH_3$, which is also considered in Chapter 2. The conformations of this molecule are well-known and can be obtained, for instance, from a long trajectory, computed with the aid of a molecular dynamics simulator such as Amber [25], using the data-driven methods presented in this thesis. It is important to note here that we reduce the dynamics of the system to the *essential* dynamics given by the dihedral angle, thus reducing the 42-dimensional state space – three coordinates for each atom – to a one-dimensional state space. In general, however, the essential coordinates are unknown a priori. A typical workflow then is to project the dynamics onto the slowest relaxation processes of the system, which can be estimated using TICA as described in Chapter 3. Alternatively, tensor-based discretization schemes could be applied as discussed in Chapter 4. The problem of obtaining the essential dynamics is also addressed in Appendix A.

One drawback of the various data-driven approaches considered here is that these methods typically suffer from the *curse of dimensionality*. Consider, for instance, the deca-alanine molecule shown in Figure 1.2. Even if we consider only the dihedral angles as our

**Table 1.1:** Sizes of eigenvectors for different dimensions $d$ and numbers of basis functions per dimension $k$.

| | k = 2 | k = 4 | k = 8 | k = 16 |
|---|---|---|---|---|
| d = 2 | 32 B | 128 B | 512 B | 2 kB |
| d = 4 | 128 B | 2 kB | 32 kB | 512 kB |
| d = 8 | 2 kB | 512 kB | 128 MB | 32 GB |
| d = 16 | 512 kB | 32 GB | 2 PB | 128 EB |

essential coordinates – as described above for the butane molecule –, the state space is still 16-dimensional (instead of 309-dimensional). Often basis functions are selected for each dimension separately and then the tensor product of these one-dimensional basis functions is defined to be the dictionary. Let us assume that we want to use the same number of basis functions for each variable of the system, say $k$. Then overall $k^d$ functions and thus also $k^d$ coefficients are required. This is illustrated in Table 1.1. For the deca-alanine molecule, we have $d = 16$ dihedral angles. Representing each angle of the system using only two basis functions is feasible, but would result in a coarse approximation. If we double the number of basis functions, the eigenvector representation requires already 32 GB and is thus challenging. Increasing the number of basis functions even further would be prohibitively expensive.

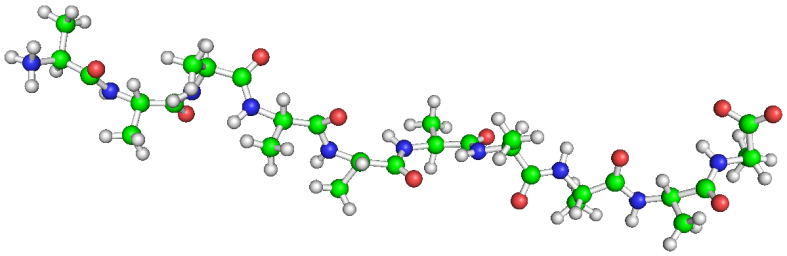


**Figure 1.2:** Deca-alanine molecule comprised of ten alanine residues.

One way to mitigate this exponential growth in the number of basis functions is to use low-rank tensor representations of the multidimensional arrays containing the coefficients for the basis functions. Tensors, in our sense, are simply multidimensional arrays. Instead of storing these arrays in the full format, only low-rank representations are computed and stored. This can be regarded as a generalization of the standard singular value decomposition of matrices. Over the last years, different tensor formats have been developed, an overview can be found in Chapter 4. We will focus mainly on the *tensor train format* or *TT format*, which is introduced and utilized in Chapter 4 and Chapter 5, where we show how data-driven methods such as DMD and EDMD can be reformulated in terms of tensors. A low-rank representation of the pseudoinverse of certain tensor unfoldings, which is required for the computation of the DMD modes of data given in the TT format, can be directly computed without converting the tensor to the full format, exploiting

properties of the TT decomposition. This is described in Chapter 5. For the tensor-based approximation of the eigenfunctions of the Koopman operator with the aid of Ulam's method or EDMD, we construct a (generalized) tensor eigenvalue problem that is then solved using simple power iteration methods, see Chapter 4. In [71], low-rank transition matrices are computed based on the reformulated version of Ulam's method and several leading eigenvalues and eigenfunctions are computed simultaneously with the *alternating linear scheme* (ALS). A tensor-based reformulation of the variational approach of conformation dynamics and numerical results for the deca-alanine molecule mentioned above are presented in [138]. Deca-alanine is also analyzed in Chapter 3 by combining TICA and Markov state models, which can be regarded as a special case of an EDMD approximation.

Another related data-driven method, which is not presented in this thesis and aims at identifying the governing equations of a dynamical system directly and not the transfer operator associated with it, is called *sparse identification of nonlinear dynamics* (SINDy), see [20]. Furthermore, a method which first approximates the Koopman operator and then, based on this operator representation, identifies the governing equations is described in [121]. This shows the close relationship between the governing equations of a system, the associated transfer operators, and their generators.

The outline of this thesis is as follows: Chapter 2 contains the publication *On the numerical approximation of the Perron–Frobenius and Koopman operator* [92]. Our publication *Data-driven model reduction and transfer operator approximation* [93] constitutes Chapter 3. In Chapter 4, we present our publication *Towards tensor-based methods for the numerical approximation of the Perron–Frobenius and Koopman operator* [95] and in Chapter 5 the publication *Tensor-based dynamic mode decomposition* [91]. Chapter 6 concludes with a summary of the main contributions of this thesis and possible future work. One additional publication related to dynamical systems and transfer operators is presented in Appendix A, where a method to compute reaction coordinates which enable the estimation of the *effective transfer operator* preserving the dominant spectrum of the full transfer operator is proposed.

# 2

# On the numerical approximation of the Perron–Frobenius and Koopman operator

Stefan Klus[1], Péter Koltai[1], and Christof Schütte[1,2]

[1] Department of Mathematics and Computer Science, Freie Universität Berlin, Germany
[2] Zuse Institute Berlin, Germany

*Journal of Computational Dynamics*, 3(1):51–79, 2016

DOI: 10.3934/jcd.2016003

**Abstract**

Information about the behavior of dynamical systems can often be obtained by analyzing the eigenvalues and corresponding eigenfunctions of linear operators associated with a dynamical system. Examples of such operators are the Perron–Frobenius and the Koopman operator. In this paper, we will review different methods that have been developed over the last decades to compute finite-dimensional approximations of these infinite-dimensional operators – in particular Ulam's method and Extended Dynamic Mode Decomposition (EDMD) – and highlight the similarities and differences between these approaches. The results will be illustrated using simple stochastic differential equations and molecular dynamics examples.

## 2.1 Introduction

The two main candidates for analyzing a dynamical system using operator-based approaches are the Perron–Frobenius and the Koopman operator. These two operators are adjoint to each other in appropriately defined function spaces and it should therefore theoretically not matter which one is used to study the system's behavior. Nevertheless, different methods have been developed for the numerical approximation of these two operators.

The Perron–Frobenius operator has been used extensively in the past to analyze the global behavior of dynamical systems stemming from a plethora of different areas such as molecular dynamics [152, 171], fluid dynamics [70, 63], meteorology and atmospheric sciences [180, 179], or engineering [188, 139]. Toolboxes for computing almost invariant sets or metastable states are available and efficiently approximate the system's behavior using adaptive box discretizations of the state space. An example of such a toolbox is GAIO [38]. This approach is, however, typically limited to low-dimensional problems.

Recently, several papers have been published focusing on data-based numerical methods to approximate the Koopman operator and to analyze the associated Koopman eigenvalues, eigenfunctions, and modes [22, 195, 196]. These methods extract the relevant global behavior of dynamical systems and can, for example, be used to find lower-dimensional approximations of a system and to split a system into fast and slow subsystems as described in [64]. In many applications, the complex behavior of a dynamical system can be replicated by a small number of modes [196].

The approximation of the Perron–Frobenius operator typically requires short simulations for a large number of different initial conditions, which, without prior knowledge about the system, grows exponentially with the number of dimensions; the approximation of the Koopman operator, on the other hand, relies on potentially fewer, but longer simulations [22]. However, we will show that this is not necessarily the case, the Perron–Frobenius operator can also be approximated using just a small number of long simulations. Thus, the latter approach might be well-suited for experimentally obtained data running just a few tests with different initial conditions for a longer time. Whether the numerically obtained operator then captures the full dynamics of the system, however, depends strongly on the initial conditions chosen.

While the Koopman operator is the adjoint of the Perron–Frobenius operator, the connections between different approaches to approximate these operators have – to our knowledge – not been fully described. In this paper, we will review different numerical methods to approximate the Perron–Frobenius operator and the Koopman operator and illustrate the similarities and differences between these approaches. We will mainly focus on simple stochastic differential equations and molecular dynamics applications.

The outline of this paper is as follows: In Section 2.2, we will introduce the Perron–Frobenius operator and the Koopman operator and give a short description of basic properties. In Section 2.3, we will describe numerical methods (more precisely, generalized Galerkin methods) to obtain finite-dimensional representations of these operators and their eigenfunctions. Section 2.4 illustrates the relationship between numerical methods developed for analyzing these operators. Section 2.5 contains examples demonstrating the efficiency and characteristic properties of these numerical methods. A conclusion and possible future work will be outlined in Section 2.6. In Appendix 2.C, we draw a connection between (i) extended dynamic mode decomposition applied to molecular dynamics simulation data and (ii) a special transfer operator used in molecular conformation analysis.

## 2.2 Transfer operators

### 2.2.1 Perron–Frobenius operator

**Deterministic systems.** Historically, transfer operators have been introduced in the field of *ergodic theory*, where the main focus is on a *measure-theoretic* characterization of the behavior of dynamical systems [97, 79, 175, 141, 107, 17]. Due to this, the starting point of the considerations is often a *measure space* $(\mathbb{X}, \mathfrak{B}, \mu)$, a three-tuple of a *state space*,

a sigma-algebra, and a (probability) measure, respectively. The evolution of the state, usually in time, is described by a dynamical system $\Phi : \mathbb{X} \to \mathbb{X}$, where $\Phi$ is a $\mu$-measurable map. When not stated explicitly otherwise, time is considered to be discrete, hence a state $x \in \mathbb{X}$ evolves as $\{x, \Phi(x), \Phi^2(x), \ldots\}$. Nevertheless, most of the concepts carry over in a straightforward fashion to continuous-time systems, which we denote by $\Phi^t$, $t \geq 0$.

In order to describe the statistical behavior of the dynamical system, we are interested in how $\Phi$ affects distributions over state space. To this end, let us think of $f \in L^1(\mathbb{X}) := L^1(\mathbb{X}, \mathfrak{B}, \mu)$, with $f \geq 0$ almost everywhere (a.e.) and $\|f\|_{L^1} = 1$, as the density of an $\mathbb{X}$-valued random variable $\boldsymbol{x}$, we write $\boldsymbol{x} \sim f$. We wish to characterize the distribution of $\Phi(\boldsymbol{x})$. It turns out that if $\Phi$ is *non-singular*[1] with respect to $\mu$, then there is a $g \in L^1(\mathbb{X})$ such that $\Phi(\boldsymbol{x}) \sim g$, and $\int_{\mathbb{A}} g \, d\mu = \int_{\Phi^{-1}(\mathbb{A})} f \, d\mu$ for all $\mathbb{A} \in \mathfrak{B}$. The mapping $f \mapsto g$ can be linearly extended to a linear operator $\mathcal{P} : L^1(\mathbb{X}) \to L^1(\mathbb{X})$,

$$\int_{\mathbb{A}} \mathcal{P} f \, d\mu = \int_{\Phi^{-1}(\mathbb{A})} f \, d\mu, \quad \mathbb{A} \in \mathfrak{B},$$

the so-called *Perron–Frobenius operator* [107, 17]. It is a linear, *positive* (i.e., $f \geq 0$ implies $\mathcal{P} f \geq 0$), *non-expansive* (i.e., $\|\mathcal{P} f\|_{L^1} \leq \|f\|_{L^1}$) operator, hence a *Markov operator*. In addition, if the underlying measure $\mu$ is *invariant*, i.e., $\mu \circ \Phi^{-1} = \mu$, then $\mathcal{P} : L^p(\mathbb{X}) \to L^p(\mathbb{X})$ is a well-defined non-expansive operator for every $p \in [1, \infty]$; see [3, 17].

The Perron–Frobenius operator $\mathcal{P}$ can be seen as a linear, infinite-dimensional representation of the nonlinear, finite-dimensional dynamical system $\Phi$. To see the connection, consider for some $x \in \mathbb{X}$ the *Dirac distribution* $\delta_x(\cdot)$ as an element of $L^1(\mathbb{X})$, with $\int_{\mathbb{A}} \delta_x(y) \, d\mu(y) = 1$ if $x \in \mathbb{A}$ and 0 otherwise. Then

$$\int_{\mathbb{A}} \mathcal{P} \delta_x \, d\mu = \int_{\Phi^{-1}(\mathbb{A})} \delta_x \, d\mu = \int_{\mathbb{A}} \delta_{\Phi(x)} \, d\mu \,,$$

such that the Perron–Frobenius operator moves the center of the Dirac distribution in accordance with the dynamics.

**Non-deterministic systems.** We define the non-deterministic dynamical system $\boldsymbol{\Phi}$ as a mapping acting on $\mathbb{X}$ such that $\boldsymbol{\Phi}(x)$ is an $\mathbb{X}$-valued random variable over some implicitly given probability space. We assume that $\boldsymbol{\Phi}$ possesses a *transition density function* $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{\geq 0}$ satisfying

$$\mathsf{P}(\boldsymbol{\Phi}(x) \in \mathbb{A}) = \int_{\mathbb{A}} k(x, y) \, d\mu(y), \quad \mathbb{A} \in \mathfrak{B} \,. \tag{2.1}$$

Here, $\mathsf{P}$ denotes the probability with respect to the underlying probability space and (2.1) essentially means that $\boldsymbol{\Phi}(x) \sim k(x, \cdot)$. The existence of a transition density function can be seen as an analogue to non-singularity in the deterministic case: it ensures that $\boldsymbol{\Phi}$ does not concentrate significant probability mass in sets of zero measure[2].

[1] $\Phi$ is (measure-theoretically) non-singular with respect to $\mu$ if $\mu \circ \Phi^{-1} \ll \mu$; i.e., $\mu \circ \Phi^{-1}$ is absolutely continuous with respect to $\mu$. This condition ensures that $\Phi$ does not map sets of positive measure to sets of zero measure, that is, it can not destroy probability measure.

[2] Such a mapping is also called in the literature "$\mu$-compatible" or "null preserving" [81, 99].

For such systems, it can be quickly seen that the Perron–Frobenius operator satisfies

$$\mathcal{P}f(y) = \int f(x)k(x,y)\,d\mu(x)\,, \tag{2.2}$$

and that the Markov operator property holds as well. If the measure $\mu$ is invariant, i.e., $\mu(\mathbb{A}) = \int_{\mathbb{A}} \int k(x,y)\,d\mu(x)\,d\mu(y)$ for every $\mathbb{A} \in \mathfrak{B}$, then $\mathcal{P} : L^p(\mathbb{X}) \to L^p(\mathbb{X})$ is a well-defined non-expansive operator for every $p \in [1,\infty]$, as in the deterministic case.

*Invariant* (or *stationary*) *densities* play a special role. These are densities $f$ (i.e., positive functions with unit $L^1$ norm) which satisfy $\mathcal{P}f = f$. If such a density $f$ is unique, the system is called *ergodic*, and satisfies for any $g \in L^p(\mathbb{X})$, $p \in [1,\infty]$, that

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=0}^{n-1} g(\mathbf{\Phi}^k x) = \int gf\,d\mu \tag{2.3}$$

P-almost surely (a.s.) for $\mu$-a.e. $x \in \operatorname{supp}(f)$, where $\operatorname{supp}(f)$ is the set $\{f > 0\}$. With some additional assumptions on $k$, the convergence in (2.3) is geometric, with the rate governed by the second dominant eigenvalue of $\mathcal{P}$.

In general, eigenfunctions associated with subdominant eigenvalues correspond to the slowly converging transients of the system and yield information about *metastable* sets; sets between which a dynamical transition is a rare event. For more details, we refer to [125, 171].

### 2.2.2 Koopman operator

While the Perron–Frobenius operator describes the evolution of *densities*, the Koopman operator describes the evolution of *observables* [22]. An observable could, for instance, be a measurement or sensor probe. That is, instead of analyzing an orbit $\{x, \Phi(x), \Phi^2(x), \dots\}$ of the dynamical system, we consider the measurements $\{f(x), f(\Phi(x)), f(\Phi^2(x)), \dots\}$.

The Koopman operator $\mathcal{K} : L^\infty(\mathbb{X}) \to L^\infty(\mathbb{X})$, see e.g. [22, 195, 64], is defined by

$$\mathcal{K}f = f \circ \Phi\,. \tag{2.4}$$

The Koopman operator $\mathcal{K}$ is the adjoint of the Perron–Frobenius operator $\mathcal{P}$, i.e.

$$\langle \mathcal{P}f,\, g\rangle_\mu = \langle f,\, \mathcal{K}g\rangle_\mu\,,$$

where $\langle\cdot,\,\cdot\rangle_\mu$ is the duality pairing between $L^1$ and $L^\infty$ functions. For specific combinations of $\Phi$ and $\mu$, the Koopman operator can be defined on $L^2(\mathbb{X})$, too[3]; in what follows, we assume that this is the case.

Again, $\mathcal{K}$ is an infinite-dimensional linear operator that characterizes the finite-dimensional nonlinear system $\Phi$. To obtain the dynamics of a system defined on $\mathbb{X} \subset \mathbb{R}^d$, use the set of observables $g_i(x) = x_i$, $i = 1,\dots,d$, or in shorthand, the *vector-valued*

[3] For instance, if the measure $\mu$ is invariant under $\Phi$ [3]; or if $k \in L^\infty(\mathbb{X}\times\mathbb{X})$.

observable $g(x) = x$, where $g$ is called *full-state observable*. On vector-valued functions, the Koopman operator acts componentwise.

In order to maintain duality with the Perron–Frobenius operator, for the non-deterministic system $\mathbf{\Phi}$ with transition density function $k$, the Koopman operator is defined as

$$\mathcal{K}f(x) = \mathsf{E}\left[f(\mathbf{\Phi}(x))\right] = \int k(x,y) f(y)\, d\mu(y),$$

where $\mathsf{E}[\cdot]$ denotes the expectation value with respect to the probability measure underlying $\mathbf{\Phi}(x)$. Note that while the Koopman operator was defined here for a discrete-time dynamical system, the definition can be extended naturally to continuous-time dynamical systems as described in [22].

If $\varphi_1$ and $\varphi_2$ are eigenfunctions of the Koopman operator with eigenvalues $\lambda_1$ and $\lambda_2$, then also the product $\varphi_1\,\varphi_2$ is an eigenfunction with eigenvalue $\lambda_1\lambda_2$. The product of two functions is defined pointwise, i.e. $(\varphi_1\,\varphi_2)(x) = \varphi_1(x)\,\varphi_2(x)$. Analogously, for any eigenfunction $\varphi$ and $r \in \mathbb{R}$, $\varphi^r$ is an eigenfunction with eigenvalue $\lambda^r$ assuming that $\varphi(x) \neq 0$ for $r < 0$.

**Example 2.2.1.** Consider a linear dynamical system of the form $x_{k+1} = A\,x_k$ with $A \in \mathbb{R}^{d\times d}$, cf. [22, 195]. Let $A$ have $d$ left eigenvectors[4] $w_i$ with eigenvalues $\mu_i$, i.e. $w_i\,A = \mu_i\,w_i$ for $i = 1, \ldots, d$. Then $\varphi_i(x) = w_i\,x$ is an eigenfunction of the Koopman operator $\mathcal{K}$ with corresponding eigenvalue $\lambda_i = \mu_i$ since

$$(\mathcal{K}\varphi_i)(x) = \varphi_i(A\,x) = w_i\,A\,x = \mu_i\,w_i\,x = \mu_i\,\varphi_i(x).$$

As described above, also products of these eigenfunctions

$$\varphi_l(x) = \prod_{i=1}^{d} (w_i\,x)^{l_i}$$

are eigenfunctions with corresponding eigenvalue $\lambda_l = \prod_{i=1}^{d} \lambda_i^{l_i}$, where $l \in \mathbb{N}_0^d$ is a multi-index. For

$$A = \begin{bmatrix} 0.48 & -0.06 \\ -0.16 & 0.52 \end{bmatrix},$$

for example, the left eigenvectors are $w_1 = [0.8,\ -0.6]$ and $w_2 = \frac{1}{\sqrt{5}}[2,\ 1]$ with eigenvalues $\mu_1 = 0.6$ and $\mu_2 = 0.4$. The first eight nontrivial eigenfunctions of the Koopman operator are shown in Figure 2.1. △

Let $f : \mathbb{X} \to \mathbb{R}$ be an observable of the system that can be written as a linear combination of the linearly independent eigenfunctions $\varphi_i$, i.e.

$$f(x) = \sum_i c_i\,\varphi_i(x),$$

[4] Here and in what follows, left eigenvectors are represented as row vectors.

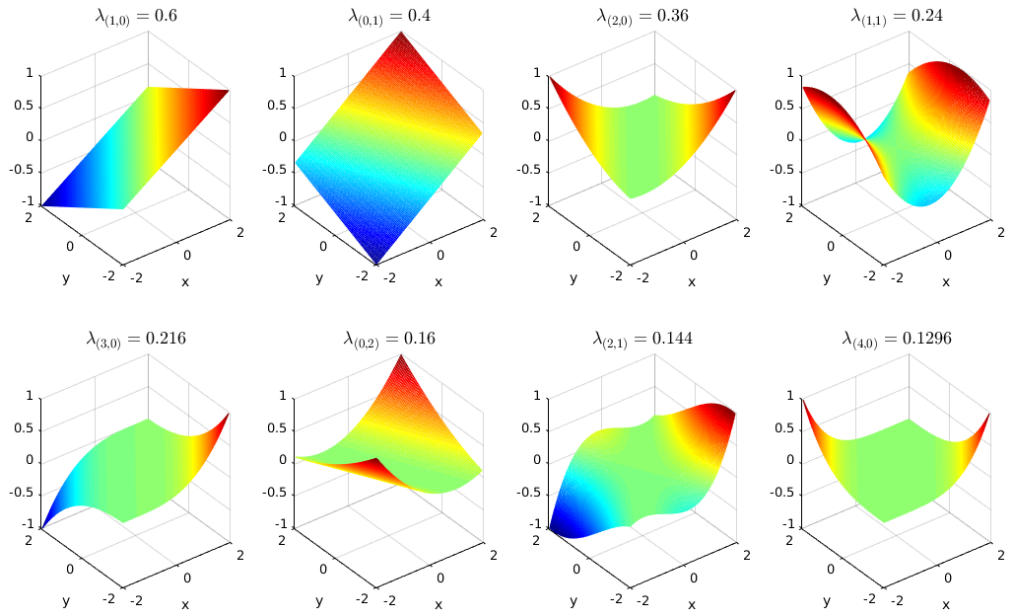



**Figure 2.1:** Eigenfunctions of the Koopman operator for the linear dynamical system described in Example 2.2.1.

with $c_i \in \mathbb{C}$. Then

$$(\mathcal{K}f)(x) = \sum_i \lambda_i \, c_i \, \varphi_i(x).$$

Analogously, for vector-valued functions $F = [f_1, \ldots, f_n]^T$, we get

$$\mathcal{K}F = \begin{bmatrix} \sum_i \lambda_i \, c_{i,1} \, \varphi_i \\ \vdots \\ \sum_i \lambda_i \, c_{i,n} \, \varphi_i \end{bmatrix} = \sum_i \lambda_i \, \varphi_i \begin{bmatrix} c_{i,1} \\ \vdots \\ c_{i,n} \end{bmatrix} = \sum_i \lambda_i \, \varphi_i \, v_i,$$

where $v_i = [c_{i,1}, \ldots, c_{i,n}]^T$. These vectors $v_i$ corresponding to the eigenfunctions $\varphi_i$ are called Koopman modes.

**Definition 2.2.2.** *Given an eigenfunction $\varphi_i$ of the Koopman operator $\mathcal{K}$ and a vector-valued observable $F$, the vector $v_i$ of coefficients of the projection of $F$ onto* $\operatorname{span}\{\varphi_i\}$ *is called Koopman mode.*

The connection between the dynamical system $\Phi$ and the Koopman operator $\mathcal{K}$ is given by the full-state observable $g(x) = x$ and the corresponding Koopman eigenvalues $\lambda_i$, eigenfunctions $\varphi_i$, and eigenmodes $v_i$ required to retrieve the full state [195]. Since $(\mathcal{K}g)(x) = (g \circ \Phi)(x) = \Phi(x)$ and, using the Koopman modes $v_i$ belonging to $g$,

$$(\mathcal{K}g)(x) = \sum_i \lambda_i \, \varphi_i(x) \, v_i,$$

we can compute $\Phi(x)$ with the aid of the Koopman operator. A pictorial representation of the relationship between states and observables as well as the evolution operator and Koopman operator can be found in [195].

## 2.3 Numerical approximation

### 2.3.1 Generalized Galerkin methods

The Galerkin discretization of an operator $\mathcal{A}$ over some Hilbert space $\mathbb{H}$ can be described as follows. Suppose we have a finite-dimensional subspace $\mathbb{V} \subset \mathbb{H}$ with basis $(\psi_1, \dots, \psi_k)$ given. The *Galerkin projection* of $\mathcal{A}$ to $\mathbb{V}$ is the unique linear operator $A : \mathbb{V} \to \mathbb{V}$ satisfying

$$\langle \psi_j,\, \mathcal{A}\psi_i \rangle = \langle \psi_j,\, A\psi_i \rangle\,, \quad \text{for all } i, j = 1, \dots, k\,. \tag{2.5}$$

If the operator $\mathcal{A}$ is not given on a Hilbert space, just a Banach space, it can be advantageous to take *basis functions* (with respect to which the projected operator is defined) and *test functions* (which serve in (2.5) to project objects not necessarily living in the same subspace) from different sets.

If $\mathcal{A} : \mathbb{Y} \to \mathbb{Y}$ is an operator on a Banach space $\mathbb{Y}$, $\mathbb{V} \subset \mathbb{Y}$ a subspace with basis $(\psi_1, \dots, \psi_k)$, $\mathbb{W} \subset \mathbb{Y}^*$ a subspace of the dual of $\mathbb{Y}$ with basis $(\psi_1^*, \dots, \psi_k^*)$, i.e. $\dim \mathbb{V} = \dim \mathbb{W}$, then the *Petrov–Galerkin* projection of $\mathcal{A}$ is the unique linear operator $A : \mathbb{V} \to \mathbb{V}$ satisfying

$$\left\langle \psi_j^*,\, \mathcal{A}\psi_i \right\rangle = \left\langle \psi_j^*,\, A\psi_i \right\rangle, \quad \text{for all } i, j = 1, \dots, k\,, \tag{2.6}$$

where $\langle \cdot, \cdot \rangle$ denotes the duality bracket.

This idea can be taken one step further, resulting in a Petrov–Galerkin-like projection even if $l := \dim \mathbb{W} > \dim \mathbb{V}$. In this case, (2.6) is over-determined and the projected operator $A$ is defined as the solution of the least-squares problem

$$\sum_{j=1}^{l} \sum_{i=1}^{k} \left\langle \psi_j^*,\, \mathcal{A}\psi_i - A\psi_i \right\rangle^2 = \min! \tag{2.7}$$

We refer to this as the *over-determined Petrov–Galerkin* method.

### 2.3.2 Ulam's method

Probably the most popular method to date for the discretization of the Perron–Frobenius operator is Ulam's method; see e.g. [187, 27, 10, 64]. Let $\{\mathbb{B}_1, \dots, \mathbb{B}_k\} \subset \mathfrak{B}$ be a covering of $\mathbb{X}$ by a finite number of disjoint measurable boxes and let $\mathbb{1}_{\mathbb{B}_i}$ be the indicator function for box $\mathbb{B}_i$, i.e.

$$\mathbb{1}_{\mathbb{B}_i}(x) = \begin{cases} 1, & \text{if } x \in \mathbb{B}_i, \\ 0, & \text{otherwise.} \end{cases}$$

Ulam's method is a Galerkin projection of the Perron–Frobenius operator to the subspace spanned by these indicator functions. More precisely, if one chooses the basis functions $\psi_i = \frac{1}{\mu(\mathbb{B}_i)} \mathbb{1}_{\mathbb{B}_i}$, then from the relationship

$$\begin{aligned} \int \mathbb{1}_{\mathbb{B}_j} \cdot \mathcal{P}\mathbb{1}_{\mathbb{B}_i} \, d\mu &= \int (\mathbb{1}_{\mathbb{B}_j} \circ \Phi) \cdot \mathbb{1}_{\mathbb{B}_i} \, d\mu = \int \mathbb{1}_{\Phi^{-1}(\mathbb{B}_j)} \cdot \mathbb{1}_{\mathbb{B}_i} \, d\mu \\ &= \mu(\Phi^{-1}(\mathbb{B}_j) \cap \mathbb{B}_i) \end{aligned} \tag{2.8}$$

we can represent the discrete operator by a matrix $P = (p_{ij}) \in \mathbb{R}^{k \times k}$ with

$$p_{ij} = \frac{\mu\left(\Phi^{-1}(\mathbb{B}_j) \cap \mathbb{B}_i\right)}{\mu(\mathbb{B}_i)}. \tag{2.9}$$

The denominator $\mu(\mathbb{B}_i)$ normalizes the entries $p_{ij}$ so that $P$ becomes a row-stochastic matrix. Thus, $P$ defines a finite Markov chain and has a left eigenvector with the corresponding eigenvalue $\lambda_1 = 1$. This eigenvector approximates the invariant measure of the Perron–Frobenius operator $\mathcal{P}$ [112, 130, 62, 40].

The entries $p_{ij}$ of the matrix $P$ can be viewed as the probabilities of being mapped from box $\mathbb{B}_i$ to box $\mathbb{B}_j$ by the dynamical system $\Phi$. These entries can be estimated by randomly choosing a large number of test points $x_i^{(l)}$, $l = 1, \ldots, n$, in each box $\mathbb{B}_i$ and counting the number of times $\Phi(x_i^{(l)})$ is contained in box $\mathbb{B}_j$, that is,

$$p_{ij} \approx \frac{1}{n} \sum_{l=1}^{n} \mathbb{1}_{\mathbb{B}_j}(\Phi(x_i^{(l)})) \,. \tag{2.10}$$

On the one hand, this is a Monte-Carlo approach to estimate the integrals in (2.8), and hence a *numerical realization* of Ulam's method. On the other hand, it is also an overdetermined Petrov–Galerkin method (2.7) with test functionals $\psi_\ell^*$ being point evaluations at the respective sample points $x_\ell$; i.e., for a piecewise continuous function $\varphi$ we have $\psi_\ell^*(\varphi) = \int \varphi \delta_{x_\ell} \, d\mu = \varphi(x_\ell)$. One can see this by noting that due to the disjoint support of the basis functions $\mathbb{1}_{\mathbb{B}_i}$ the sum in (2.7) decouples and the entries of $P$ can be readily seen to be as on the right-hand side of (2.10). The effect of Monte-Carlo sampling and the choice of the partition on the accuracy and convergence of Ulam's method has been investigated in [11, 131, 96].

**Remark 2.3.1.** We note that, given independent random test points $x_i^{(l)} \in \mathbb{B}_i$, $l = 1, \ldots, n$, expression (2.10) is a maximum-likelihood estimator for (2.9). This holds true in the non-deterministic case as well, where (2.9) reads as

$$p_{ij} = \mathsf{P}\left(\mathbf{\Phi}(x_i^{(l)}) \in \mathbb{B}_j\right),$$

and the $\Phi(x_i^{(l)})$ in (2.10) are replaced by mutually independent realizations of $\mathbf{\Phi}(x_i^{(l)})$, $l = 1, \ldots, n$.

### 2.3.3 Further discretization methods for the Perron–Frobenius operator

**Petrov–Galerkin type and higher order methods.** Ulam's method is a zeroth order method in the sense that it uses piecewise constant basis functions. We can achieve a better approximation of the operator (and its dominant spectrum, in particular) if we use higher order piecewise polynomials in the Galerkin approximation; see [44, 46].

If the eigenfunctions of the Perron–Frobenius operator are expected to have further regularity, the use of *spectral methods* can be advantageous [82, 67]. Here, *collocation* turns out to be the most efficient, in general; i.e., where basis functions are Fourier or Chebyshev polynomials [18], and test functions are Dirac distributions centered in specific domain-dependent collocation points. *Mesh-free* approaches with radial basis functions continuously gain popularity due to their flexibility with respect to state space geometry [66, 197].

A kind of regularity different from smoothness is if functions of interest do not vary simultaneously strongly in many coordinates, just in very few of them. *Sparse-grid* type Galerkin approximation schemes [23] are well suited for such objects; their combination with Ulam's method has been considered in [88].

Higher-order approximations do have, however, an unwanted disadvantage: the discretized operator is not a Markov operator (a stochastic matrix), in general [96, Section 3]. This desirable structural property can be retained if one considers specific Petrov–Galerkin methods; cf. [45], where the basis functions are piecewise first- or second-order polynomials and the test functions are piecewise constant.

**Maximum entropy optimization methods.** Let us consider a Petrov–Galerkin method for discretizing the Perron–Frobenius operator $\mathcal{P}$, such that $\psi_j^*(\varphi) = \int h_j \varphi \, d\mu$ for suitable $h_j \in L^\infty(\mathbb{X})$, $j = 1, \dots, k$. Then the image $Pf$ of $f \in \mathbb{V}$ is the unique element $g \in \mathbb{V}$ such that

$$\int (\mathcal{P}f - g) h_j \, d\mu = \int f(h_j \circ \Phi) - g h_j \, d\mu = 0, \quad j = 1, \dots, k \,. \tag{2.11}$$

One might as well alleviate the condition $g \in \mathbb{V}$, at the cost of not having a unique solution to (2.11). Then, in order to get a unique solution, one has to impose additional conditions on $g$. If one considers (2.11) as *constraints*, one could formulate an optimization problem whose solution is $g$. There is, of course, no trivial choice of objective functional for this optimization problem, however *energy-type* (i.e. $\int g^2 \, d\mu$) and *entropy-type* (i.e. $\int g \log g \, d\mu$) objective functionals turned out to be advantageous to use [43, 13, 12, 14]. The reason for this is that the available convergence analysis for Ulam's method is quite restrictive [112, 46, 62], and these optimization-based methods yield novel convergent schemes to approximate invariant densities of non-singular dynamical systems – to this end, one sets $g = f$ in (2.11). The down-side of this method is that in order to represent the approximate invariant density, one has to compute "basis functions" which arise as non-trivial combinations of the test functions $h_j$ and the dynamics $\Phi$.

### 2.3.4 Extended dynamic mode decomposition

An approximation of the Koopman operator, the Koopman eigenvalues, eigenfunctions, and eigenmodes can be computed using *Extended Dynamic Mode Decomposition* (EDMD). Note that we are using a slightly different notation than [195, 196] here to make the relationship with other methods, in particular Ulam's method and *Dynamic Mode Decomposition* (DMD, defined in Remark 2.3.6 below), more apparent. In order to obtain EDMD, we take the basis functions $\psi_i$, as above, and for the test function(al)s, we take delta distributions $\delta_{x_j}$, that is, $\langle \delta_x, \psi \rangle = \psi(x)$. EDMD requires data, i.e. a set of values $x_i$ and the corresponding $y_i = \Phi(x_i)$ values, $i = 1, \ldots, m$, written in matrix form

$$X = \begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} y_1 & \cdots & y_m \end{bmatrix}, \tag{2.12}$$

and additionally a set of basis functions or observables

$$\mathbb{D} = \{\psi_1, \psi_2, \ldots, \psi_k\}$$

called dictionary. EDMD takes ideas from collocation methods, which are, for example, used to solve PDEs, where the $x_i$ are the collocation points rather than a fixed grid [196]. Writing

$$\Psi = \begin{bmatrix} \psi_1 & \psi_2 & \cdots & \psi_k \end{bmatrix}^T$$

as a vector of functions, that is $\Psi : \mathbb{X} \to \mathbb{R}^k$, this yields

$$\Psi_Y^T = \Psi_X^T K,$$

with

$$\Psi_X = \begin{bmatrix} \Psi(x_1) & \ldots & \Psi(x_m) \end{bmatrix} \quad \text{and} \quad \Psi_Y = \begin{bmatrix} \Psi(y_1) & \ldots & \Psi(y_m) \end{bmatrix},$$

i.e. $\Psi_X, \Psi_Y \in \mathbb{R}^{k \times m}$. Here, $K \in \mathbb{R}^{k \times k}$ applied from the right to vectors in $\mathbb{R}^{1 \times k}$ represents the projection of $\mathcal{K}$ with respect to the basis $(\psi_1, \ldots, \psi_k)$. If the number of basis functions and test functions does not match, (2.6) cannot be satisfied in general and a least squares solution of the (usually overdetermined) system of equations is given by applying $\Psi_X^+$, the pseudoinverse of $\Psi_X$, giving

$$K^T = \Psi_Y \, \Psi_X^+. \tag{2.13}$$

A more detailed description can be found in Appendix 2.D. For the sake of convenience and to compare DMD and EDMD, we define $M_{\mathrm{K}} = K^T$. This approach becomes computationally expensive for large $m$ since it requires the pseudoinverse of the $k \times m$ matrix $\Psi_X$. Another possibility to compute $K$ is

$$K^T = A \, G^+,$$

where the matrices $A$, $G \in \mathbb{R}^{k \times k}$ are given by

$$\begin{aligned} A &= \frac{1}{m} \sum_{l=1}^{m} \Psi(y_l) \, \Psi(x_l)^T, \\ G &= \frac{1}{m} \sum_{l=1}^{m} \Psi(x_l) \, \Psi(x_l)^T. \end{aligned} \tag{2.14}$$

In order to obtain the second EDMD formulation from the first, the relationship $\Psi_X^+ = \Psi_X^T (\Psi_X \Psi_X^T)^+$ was used. For a detailed derivation of these results, we refer to [195, 196].

An approximation of the eigenfunction $\varphi_i$ of the Koopman operator $\mathcal{K}$ is then given by

$$\varphi_i = \xi_i \Psi,$$

where $\xi_i$ is the $i$-th left eigenvector of the matrix $M_\mathrm{K} = K^T$.

**Example 2.3.2.** Let us consider the linear system described in Example 2.2.1 again. The eigenfunctions computed using EDMD with the basis functions $\psi_l = x_1^{l_1} x_2^{l_2}$, $0 \leq l_1, l_2 \leq 5$, are in very good agreement with the theoretical results. EDMD computes exactly the eigenfunctions shown in Figure 2.1 with negligibly small numerical errors $\varepsilon < 10^{-10}$, where we computed the maximum difference between the eigenfunctions and their approximation. The first eight nontrivial eigenvalues of $M_\mathrm{K}$ are

$$\begin{aligned} &\lambda_2 = 0.6000, && \lambda_3 = 0.4000, && \lambda_4 = 0.3600, && \lambda_5 = 0.2400, \\ &\lambda_6 = 0.2160, && \lambda_7 = 0.1600, && \lambda_8 = 0.1440, && \lambda_9 = 0.1296. \end{aligned}$$ △

In order to obtain the Koopman modes for the full-state observable $g(x) = x$ introduced above, define $\boldsymbol{\varphi} = [\varphi_1, \ldots, \varphi_k]^T$ and let $B \in \mathbb{R}^{d \times k}$ be the matrix such that $g = B\,\Psi$, then $\boldsymbol{\varphi} = \Xi\,\Psi$ and

$$g = B\,\Psi = B\,\Xi^{-1}\boldsymbol{\varphi},$$

where the matrix

$$\Xi = \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_k \end{pmatrix}$$

contains all left eigenvectors of $M_\mathrm{K}$. Thus, the column vectors of the $(d \times k)$-dimensional matrix $V = B\,\Xi^{-1}$ are the Koopman modes $v_i$ and

$$g = \sum_i \varphi_i\, v_i \quad \Rightarrow \quad \mathcal{K}g = \sum_i \lambda_i\, \varphi_i\, v_i.$$

Note that since $\Xi$ is the matrix which contains all left eigenvectors of $M_\mathrm{K}$, the matrix $\Xi^{-1}$ needed for reconstructing the full-state observable $g$ contains all right eigenvectors of $M_\mathrm{K}$. That is, the Koopman eigenfunctions $\boldsymbol{\varphi} = \Xi\,\Psi$ are approximated by the left eigenvectors of $M_\mathrm{K}$ and the Koopman modes $V = B\,\Xi^{-1}$ by the right eigenvectors (cf. [195], with the difference that there the observables and eigenfunctions are written as column vectors and the data matrices $\Psi_X$ and $\Psi_Y$ are the transpose of our matrices; we chose to rewrite the EDMD formulation in order to illustrate the similarities with DMD and other methods).

**Example 2.3.3.** For Example 2.2.1 and the dictionary $\psi_l = x_1^{l_1} x_2^{l_2}$, $0 \leq l_1, l_2 \leq 5$, the matrix $B \in \mathbb{R}^{2 \times 36}$ is zero except for the two entries corresponding to the functions $\psi_{(1,0)} = x_1$ and $\psi_{(0,1)} = x_2$. Thus, only the two eigenmodes $v_{(1,0)} \approx [0.5, -1]^T$ and $v_{(0,1)} \approx \frac{\sqrt{5}}{2}[0.6, 0.8]^T$ – eigenvectors of $A$ – are required to construct the full-state observable, all the other eigenmodes are numerically zero. △

**Remark 2.3.4** (**Convergence of EDMD to a Galerkin method**). As described in [195], EDMD converges to a Galerkin approximation of the Koopman operator for large $m$ if the data points are drawn according to a distribution $\mu$. Using the Galerkin approach, we would obtain matrices $\widetilde{A}$ and $\widetilde{G}$ with entries

$$\begin{aligned}\widetilde{a}_{ij} &= \langle \mathcal{K}\psi_i,\, \psi_j \rangle_\mu\,,\\ \widetilde{g}_{ij} &= \langle \psi_i,\, \psi_j \rangle_\mu\,.\end{aligned}$$

Here, $\langle f,\, g\rangle_\mu = \int_{\mathbb{X}} f(x)\, g^*(x)\, d\mu(x)$. Then $\widetilde{K}^T = \widetilde{A}\,\widetilde{G}^{-1}$ would be the finite-dimensional approximation of the Koopman operator $\mathcal{K}$. Clearly, the entries $a_{ij}$ and $g_{ij}$ of the matrices $A$ and $G$ in (2.14) converge to $\widetilde{a}_{ij}$ and $\widetilde{g}_{ij}$ for $m \to \infty$, since

$$\begin{aligned}a_{ij} &= \frac{1}{m}\sum_{l=1}^{m} \psi_i(y_l)\, \psi_j(x_l)^* \xrightarrow[m\to\infty]{} \int_{\mathbb{X}} (\mathcal{K}\psi_i)(x)\, \psi_j(x)^*\, d\mu(x) = \langle \mathcal{K}\psi_i,\, \psi_j\rangle_\mu = \widetilde{a}_{ij},\\ g_{ij} &= \frac{1}{m}\sum_{l=1}^{m} \psi_i(x_l)\, \psi_j(x_l)^* \xrightarrow[m\to\infty]{} \int_{\mathbb{X}} \psi_i(x)\, \psi_j(x)^*\, d\mu(x) = \langle \psi_i,\, \psi_j\rangle_\mu = \widetilde{g}_{ij}.\end{aligned} \tag{2.15}$$

**Remark 2.3.5** (**Variational approach for reversible processes**). The EDMD approximation of the eigenfunctions of the Koopman operator is given by the left eigenvectors $\xi$ of the matrix $M_{\mathrm{K}} = A\,G^+$, i.e. $\xi\, M_{\mathrm{K}} = \lambda\,\xi$, and can be – provided that $G$ is regular – reformulated as a generalized eigenvalue problem of the form $\xi\, A = \lambda\,\xi\, G$. This results in a method similar to the variational approach presented in [134] for reversible processes. A tensor-based generalization of this method can be found in [138].

**Remark 2.3.6** (**DMD**). Dynamic Mode Decomposition was first introduced in [163] and is a powerful tool for analyzing the behavior of nonlinear systems which can, for instance, be used to identify low-order dynamics of a system [186]. DMD analyzes pairs of $d$-dimensional data vectors $x_i$ and $y_i = \Phi(x_i)$, $i = 1, \ldots, m$, written again in matrix form (2.12). Assuming there exists a linear operator $M_{\mathrm{L}}$ that describes the dynamics of the system such that $y_i = M_{\mathrm{L}}\, x_i$, define $M_{\mathrm{L}} = Y X^+$. The DMD modes and eigenvalues are then defined to be the eigenvectors and eigenvalues of $M_{\mathrm{L}}$. The matrix $M_{\mathrm{L}}$ minimizes the cost function $\|M_{\mathrm{L}}\, X - Y\|_F$, where $\|.\|_F$ is the Frobenius norm. There are different algorithms to compute the DMD modes and eigenvalues without explicitly computing $M_{\mathrm{L}}$ which rely on the (reduced) singular value decomposition of $X$. For a detailed description, we refer to [186].

**Remark 2.3.7** (**DMD and EDMD**). The first EDMD formulation (2.13) shows the relationship between DMD and EDMD. Let the vector of observables be given by $\Psi(x) = x$. Then $\Psi_X = X$ and $\Psi_Y = Y$, thus

$$M_{\mathrm{K}} = \Psi_Y\, \Psi_X^+ = Y\, X^+ = M_{\mathrm{L}},$$

i.e. the DMD matrix $M_{\mathrm{L}}$ is an approximation of the Koopman operator $\mathcal{K}$ using only linear basis functions. Since $B = I$, the Koopman modes are $V = \Xi^{-1}$, which are the

right eigenvectors of $M_{\mathrm{K}}$ and thus the right eigenvectors of $M_{\mathrm{L}}$, which illustrates that the Koopman modes in this case are the DMD modes. Hence, (exact) DMD can be regarded as a special case of EDMD.

**Remark 2.3.8** (**Sparsity-promoting DMD**)**.** A variant of DMD aiming at maximizing the quality of the approximation while minimizing the number of modes used to describe the data is presented in [87]. Sparsity is achieved by using an $\ell_1$-norm regularization approach. The $\ell_1$-norm can be regarded as a convexification of the cardinality function. The resulting regularized convex optimization problem is then solved with an alternating direction method. That is, the algorithm alternates between minimizing the cost function and maximizing sparsity.

In the same way, a sparsity-promoting version of EDMD could be constructed in order to minimize the number of basis functions required for the representation of the eigenfunctions.

### 2.3.5 Kernel-based extended dynamic mode decomposition

In some cases, it is possible to improve the efficiency of EDMD using the so-called *kernel trick* [196]. In fluid problems, for example, the number of measurement points $k$ is typically much larger than the number of measurements or snapshots $m$. Suppose $f(x, y) = (1 + x^T y)^2$ for $x, y \in \mathbb{R}^2$, then

$$f(x,y) = 1 + 2\,x_1\,y_1 + 2\,x_2\,y_2 + 2\,x_1\,x_2\,y_1\,y_2 + x_1^2\,y_1^2 + x_2^2\,y_2^2 = \Psi(x)^T\,\Psi(y)$$

for the vector of observables $\Psi(x) = \left[1,\, \sqrt{2}x_1,\, \sqrt{2}x_2,\, \sqrt{2}x_1\,x_2,\, x_1^2,\, x_2^2\right]^T$. The kernel function $f(x, y) = (1 + x^T y)^p$ for $x, y \in \mathbb{R}^d$ will generate a vector-valued observable that contains all monomials of order up to and including $p$. That is, instead of $\mathcal{O}(k)$, the computation of the inner product is now $\mathcal{O}(d)$ since inner products are computed implicitly by an appropriately chosen kernel function.

In [196], it is shown that any left eigenvector $v$ of $M_{\mathrm{K}}$ for an eigenvalue $\lambda \neq 0$ can be written as $v = \hat{v}\,\Psi_X^T$, with $\hat{v} \in \mathbb{R}^m$. Using the relationship $\Psi_X^+ = (\Psi_X^T\,\Psi_X)^+\Psi_X^T$, we then obtain

$$\begin{array}{l} v\,M_{\mathrm{K}} = \hat{v}\,\Psi_X^T\,M_{\mathrm{K}} = \hat{v}\,\Psi_X^T(\Psi_Y\Psi_X^+) = \hat{v}\,(\Psi_X^T\,\Psi_Y)(\Psi_X^T\,\Psi_X)^+\Psi_X^T = \hat{v}\,\hat{M}_{\mathrm{K}}\,\Psi_X^T \\ \quad\|\\ \mu\,v \;\; = \mu\,\hat{v}\,\Psi_X^T \end{array}$$

and thus a left eigenvector of $M_{\mathrm{K}}$ can be computed by a left eigenvector of $\hat{M}_{\mathrm{K}} = \hat{K}^T = \hat{A}\,\hat{G}^+$ multiplied by $\Psi_X^T$, where $\hat{A} = \Psi_X^T\,\Psi_Y \in \mathbb{R}^{m\times m}$ and $\hat{G} = \Psi_X^T\,\Psi_X \in \mathbb{R}^{m\times m}$. The entries of the matrices $\hat{A}$ and $\hat{G}$ can be computed efficiently by

$$\begin{aligned} \hat{a}_{ij} &= f(x_i, y_j), \\ \hat{g}_{ij} &= f(x_i, x_j), \end{aligned}$$

using the kernel function $f$. The computational cost for the eigenvector computation now depends on the number of snapshots $m$ rather than the number of observables $k$. For a more detailed description, we refer to [196].

## 2.4 Duality

In this section, we will show how, given the eigenfunctions of the Koopman operator, the eigenfunctions of the adjoint Perron–Frobenius operator can be computed, or vice versa. The goal here is to illustrate the similarities between the different numerical methods presented in the previous sections and to adapt methods developed for one operator to compute eigenfunctions of the other operator. We will focus in particular on Ulam's method and EDMD.

### 2.4.1 Ulam's method and EDMD

Let us consider the case where the dictionary contains the indicator functions for a given box discretization $\{\mathbb{B}_1, \ldots, \mathbb{B}_k\}$, i.e. $\mathbb{D} = \{\mathbb{1}_{\mathbb{B}_1}, \ldots, \mathbb{1}_{\mathbb{B}_k}\}$. If we now select $n$ test points $x_i^{(l)}$, $l = 1, \ldots, n$, for each box, then

$$\Psi_X = \begin{pmatrix} \mathbb{1}_n^T & & & \\ & \mathbb{1}_n^T & & \\ & & \ddots & \\ & & & \mathbb{1}_n^T \end{pmatrix} \in \mathbb{R}^{k \times k\,n},$$

where $\mathbb{1}_n \in \mathbb{R}^n$ is the vector of all ones. The pseudoinverse of this matrix is $\Psi_X^+ = \frac{1}{n}\Psi_X^T$ and the matrix $M_{\mathrm{K}} = \Psi_Y\,\Psi_X^+ \in \mathbb{R}^{k \times k}$ with entries $\underline{\mathrm{m}}_{ij}$ has the following form

$$\underline{\mathrm{m}}_{ij} = \sum_{l=1}^{k\,n} (\Psi_Y)_{il}\,(\Psi_X)_{lj}^+ = \frac{1}{n}\sum_{l=1}^{n} \psi_i(y_j^{(l)}) = \frac{1}{n}\sum_{l=1}^{n} \mathbb{1}_{\mathbb{B}_i}(\Phi(x_j^{(l)})).$$

Comparing the entries $\underline{\mathrm{m}}_{ij}$ of $M_{\mathrm{K}}$ with the entries $p_{ij}$ of $P$ in (2.10), it turns out that $M_{\mathrm{K}} = P^T$ and thus $P = K$. That is, EDMD with indicator functions for a given box discretization computes the same finite-dimensional representation of the operators as Ulam's method.

### 2.4.2 Computation of the dual basis

For the finite-dimensional approximation, let $\varphi_i$ be the eigenfunctions of $\mathcal{K}$ and $\widetilde{\varphi}_i$ the eigenfunctions of the adjoint operator $\mathcal{P}$, $i = 1, \ldots, k$. Since

$$\langle \mathcal{K}\varphi_i,\, \widetilde{\varphi}_j \rangle_\mu = \lambda_i \langle \varphi_i,\, \widetilde{\varphi}_j \rangle_\mu \quad \text{and} \quad \langle \varphi_i,\, \mathcal{P}\widetilde{\varphi}_j \rangle_\mu = \lambda_j \langle \varphi_i,\, \widetilde{\varphi}_j \rangle_\mu,$$

subtracting these two equations gives $0 = (\lambda_i - \lambda_j)\langle \varphi_i,\, \widetilde{\varphi}_j \rangle_\mu$. The left-hand side of the equation is zero due to the definition of the adjoint operator. Thus, if $\lambda_i \neq \lambda_j$, the scalar product must be zero. Furthermore, $\widetilde{\varphi}_j$ can be scaled in such a way that $\langle \varphi_i,\, \widetilde{\varphi}_i \rangle_\mu = 1$. Hence, we can assume that $\langle \varphi_i,\, \widetilde{\varphi}_j \rangle_\mu = \delta_{ij}$.

Let now $B = (b_{ij}) \in \mathbb{C}^{k\times k}$ and $C = (c_{ij}) \in \mathbb{C}^{k\times k}$. Define $b_{ij} = \langle \varphi_i, \varphi_j \rangle_\mu$ and write

$$\widetilde{\varphi}_j = \sum_{l=1}^{k} c_{jl}\, \varphi_l,$$

then

$$\langle \varphi_i, \widetilde{\varphi}_j \rangle_\mu = \left\langle \varphi_i, \sum_{l=1}^{k} c_{jl}\, \varphi_l \right\rangle_\mu = \sum_{l=1}^{k} c_{jl}^* \langle \varphi_i, \varphi_l \rangle_\mu = \sum_{l=1}^{k} b_{il}\, c_{jl}^* = \sum_{l=1}^{k} b_{il}\, c_{lj}.$$

It follows that the coefficients $c_{ij}$ have to be chosen such that $C = B^{-1}$. In order to obtain the matrix $B$, we compute

$$\begin{aligned} b_{ij} = \langle \varphi_i, \varphi_j \rangle_\mu &\approx \langle \xi_i\, \Psi, \xi_j\, \Psi \rangle_\mu = \frac{1}{m} \sum_{l=1}^{m} (\xi_i\, \Psi(x_l))\, (\xi_j\, \Psi(x_l))^* \\ &= \frac{1}{m}\, (\xi_i\, \Psi_X)\, (\xi_j\, \Psi_X)^* = \frac{1}{m}\, \xi_i\, G\, \xi_j^*, \end{aligned}$$

where again $G = \Psi_X \Psi_X^T$. That is,

$$B = \frac{1}{m} \Xi\, G\, \Xi^* \quad \Rightarrow \quad C = B^{-1} = m\, (\Xi^*)^{-1}\, G^{-1}\, \Xi^{-1}.$$

Here, we assume that the matrix $G$ is invertible. It follows that

$$\widetilde{\varphi}_j = \sum_{l=1}^{k} c_{jl}\, \xi_l\, \Psi = \widetilde{\xi}_j\, \Psi,$$

where $\widetilde{\Xi} = C\, \Xi = m\, (\Xi^*)^{-1}\, G^{-1}$. The drawback of this approach is that all the eigenvectors of the matrix $M_{\mathrm{K}}$ need to be computed, which – for a large number of basis functions – might be prohibitively time-consuming. We are often only interested in the leading eigenfunctions.

### 2.4.3 EDMD for the Perron–Frobenius operator

EDMD as presented in Section 2.3 can also directly be used to compute an approximation of the eigenfunctions of the Perron–Frobenius operator. Since

$$\widetilde{a}_{ij} = \langle \mathcal{K}\psi_i, \psi_j \rangle_\mu = \langle \psi_i, \mathcal{P}\psi_j \rangle_\mu,$$

the entries of the matrix $\widetilde{A}^T$ are given by $\langle \mathcal{P}\psi_i, \psi_j \rangle_\mu$. The matrices $A$ and $G$ are approximations of $\widetilde{A}$ and $\widetilde{G}$, respectively. Thus, the eigenfunctions of the Perron–Frobenius operator can be approximated by computing the eigenvalues and left eigenvectors of

$$M_{\mathrm{P}} = A^T\, G^+. \tag{2.16}$$

Analogously, the generalized eigenvalue problem

$$\widetilde{\xi}\, A^T = \lambda\, \widetilde{\xi}\, G$$

can be solved. We discuss an even more general way of approximating the adjoint operator in Appendix 2.A.

**Example 2.4.1.** Let us compute the dominating eigenfunction of the Perron–Frobenius operator for the linear system introduced in Example 2.2.1. Note that the origin is a fixed point and we would expect the invariant density to be the Dirac distribution $\delta$ with center $(0, 0)$. Using monomials of order up to 10 and thin plate splines of the form $\psi(r) = r^2 \ln r$, where $r$ is the distance between the point $(x, y)$ and the center, respectively, we obtain the approximations shown in Figure 2.2. This illustrates that the results strongly depend on the basis functions chosen. EDMD will return only meaningful results if the eigenfunctions can be represented by the selected basis.

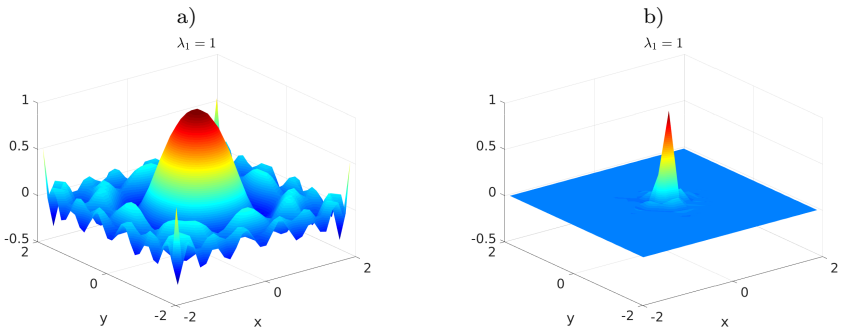



**Figure 2.2:** Approximation of the invariant density of the linear system from Example 2.2.1 using a) monomials of order up to 10 and b) thin plate splines. This example shows that EDMD captures the eigenfunctions only if they can be represented by the basis chosen.

One possibility to detect whether the chosen basis is sufficient to approximate the dominant eigenfunctions accurately is to add additional basis functions and to check whether the results remain essentially unchanged. Here, one should take into account that the condition number of the problem might deteriorate if a large number of basis functions is used. Another possibility is to compute the residual $\left\|\Psi_Y - K^T \Psi_X\right\|_F$. A large error indicates that the set of basis functions cannot represent the eigenfunctions accurately. △

**Remark 2.4.2.** The eigenfunctions computed in the previous subsection are identical to the ones computed here. The matrix $\Xi$ can be computed as the solution of the following generalized eigenvalue problem $\Xi\, A = \Lambda\, \Xi\, G$. Hence, we get $A^T = G^T\, \Xi^*\, \Lambda^*\, (\Xi^*)^{-1}$. Then $\widetilde{\Xi} = (\Xi^*)^{-1}\, G^{-1}$, neglecting the factor $m$, solves the generalized eigenvalue problem

$$\widetilde{\Xi}\, A^T = (\Xi^*)^{-1}\, G^{-1}\, G^T\, \Xi^*\, \Lambda^*\, (\Xi^*)^{-1} = \Lambda^*\, (\Xi^*)^{-1} = \Lambda^*\, \widetilde{\Xi}\, G,$$

using the fact that $G$ is symmetric.

This shows that the eigenfunctions of the Koopman operator are approximated by the left eigenvectors and the eigenfunctions of the Perron–Frobenius operator by the right eigenvectors of the generalized eigenvalue problem with the matrix pencil given by $(A, G)$. The advantage of this approach is that arbitrary basis functions can be chosen to compute eigenfunctions of the Perron–Frobenius operator. This might be beneficial if the eigenfunctions can be approximated by a small number of smooth functions – for instance monomials, Hermite polynomials, or radial basis functions – whereas using Ulam's method a large number of indicator functions would be required.

## 2.5 Numerical examples

In this section, we will illustrate the different methods described in the paper using simple stochastic differential equations and molecular dynamics examples.

### 2.5.1 Double-well problem

Consider the following stochastic differential equation

$$\begin{aligned} d\boldsymbol{x}_t &= -\nabla_x V(\boldsymbol{x}_t, \boldsymbol{y}_t)\, dt + \sigma\, d\boldsymbol{w}_{t,1}, \\ d\boldsymbol{y}_t &= -\nabla_y V(\boldsymbol{x}_t, \boldsymbol{y}_t)\, dt + \sigma\, d\boldsymbol{w}_{t,2}, \end{aligned} \tag{2.17}$$

where $\boldsymbol{w}_{t,1}$ and $\boldsymbol{w}_{t,2}$ are two independent standard Wiener processes. In this example, the potential, shown in Figure 2.3a, is given by $V(x, y) = (x^2 - 1)^2 + y^2$ and $\sigma = 0.7$.

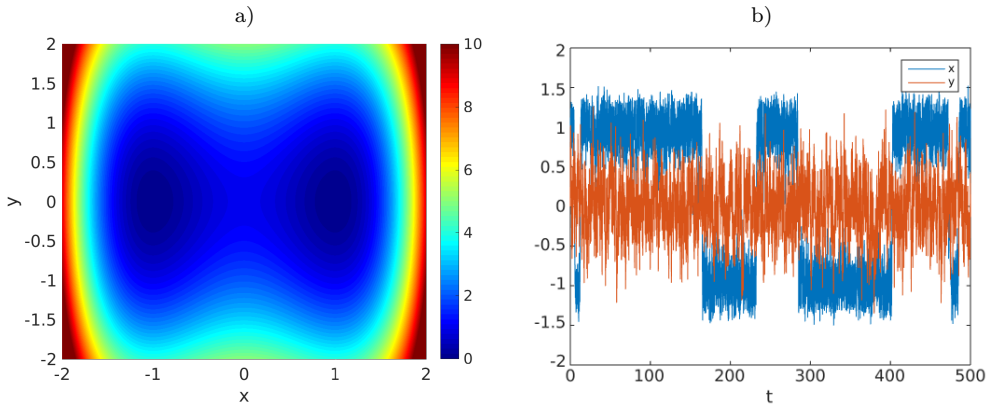



**Figure 2.3:** a) Double-well potential $V(x, y) = (x^2 - 1)^2 + y^2$. The $x$ variable typically stays for a long time close to $x = -1$ or $x = 1$ and rarely switches from one state to the other. The $y$ variable oscillates around the equilibrium $y = 0$. b) Numerical solution of the double-well SDE (2.17).

Numerically, this system can be solved using the Euler–Maruyama method, which, for an SDE of the form

$$d\boldsymbol{x}_t = \mu(t, \boldsymbol{x}_t)\, dt + \sigma(t, \boldsymbol{x}_t)\, d\boldsymbol{w}_t,$$

can be written as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h\,\mu(t_k, \boldsymbol{x}_k) + \sigma(t_k, \boldsymbol{x}_k)\,\Delta\boldsymbol{w}_k,$$

where $h$ is the step size and $\Delta\boldsymbol{w}_k = \boldsymbol{w}_{k+1} - \boldsymbol{w}_k \sim \mathcal{N}(0, h)$. Here, $\mathcal{N}(0, h)$ denotes a normal distribution with mean 0 and variance $h$. A typical trajectory of system (2.17) is shown in Figure 2.3b.

In order to compare the different methods described in the preceding sections, we conputed the leading eigenfunctions with Ulam's method and EDMD. For Ulam's method, we partitioned the domain $\Omega = [-2, 2]^2$ into $50 \times 50$ boxes of the same size. For EDMD, we used monomials of order up to and including 10, i.e.

$$\mathbb{D} = \{1,\, x,\, y,\, x^2,\, x\,y,\, y^2,\, \ldots,\, x^2\,y^8,\, x\,y^9,\, y^{10}\}.$$

That is, Ulam's method requires 2500 parameters to describe the eigenfunctions while EDMD requires only 66. For each box, we generated $n = 100$ test points, i.e. 250000 test points overall, and used the same test points also for EDMD resulting in $\Psi_X, \Psi_Y \in \mathbb{R}^{66\times 250000}$. The system (2.17) is solved using the Euler–Maruyama method with a step size of $h = 10^{-3}$. One evaluation of the corresponding dynamical system $\boldsymbol{\Phi}$ corresponds to 10000 steps. That is, each initial condition is integrated from $t_0 = 0$ to $t_1 = 10$. The first two eigenfunctions of the Perron–Frobenius operator and Koopman operator are shown in Figure 2.4. Observe that the computed eigenvalues are – as expected – almost identical. The second eigenfunction computed with Ulam's method is still very coarse, increasing the number of test points per box would smoothen the approximation. Since for EDMD only smooth basis functions were chosen, the resulting eigenfunction is automatically smoothened.

The system has two metastable states and the second eigenfunction of the Perron–Frobenius operator can be used to detect these metastable states. Also the second eigenfunction of the adjoint Koopman operator contains information about a possible partitioning of the state space, it is almost constant in the $y$-direction and also almost constant in the $x$-direction except for an abrupt transition from $-1$ to 1 between the two metastable sets. The other eigenvalues of the system are numerically zero.

### 2.5.2 Triple-well problem

Consider the slightly more complex triple-well potential

$$\begin{aligned} V(x,y) = {} & 3\,e^{-x^2-(y-\frac{1}{3})^2} - 3\,e^{-x^2-(y-\frac{5}{3})^2} - 5\,e^{-(x-1)^2-y^2} - 5\,e^{-(x+1)^2-y^2} \\ & + \tfrac{2}{10}\,x^4 + \tfrac{2}{10}\left(y - \tfrac{1}{3}\right)^4 \end{aligned} \tag{2.18}$$

taken from [171]. Here, the variables $x$ and $y$ are coupled, i.e. the potential cannot be written as $V(x, y) = V_1(x) + V_2(y)$ anymore. The potential function is shown in Figure 2.5 and the first two nontrivial eigenfunctions of the Perron–Frobenius operator and the Koopman operator in Figure 2.6. Note that the eigenfunction $\varphi_2$ separates the two deep wells at $(-1, 0)$ and $(1, 0)$ and is near zero for the well at $(0, 1.5)$, the third eigenfunction $\varphi_3$

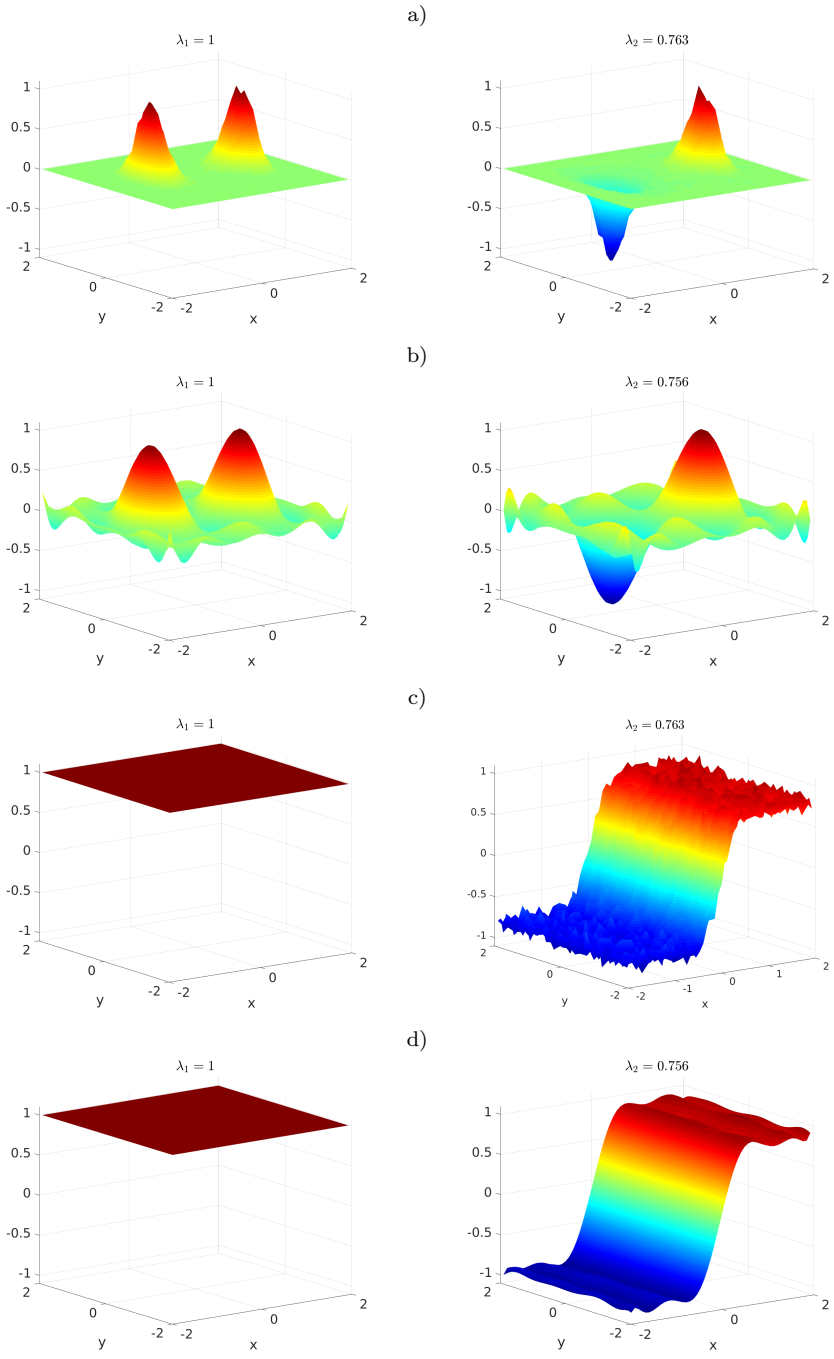



**Figure 2.4:** The first two eigenfunctions of the Perron–Frobenius operator $\mathcal{P}$ for the double-well problem computed using a) Ulam's method and b) EDMD and the first two eigenfunctions of the Koopman operator $\mathcal{K}$ computed using c) Ulam's method and d) EDMD.

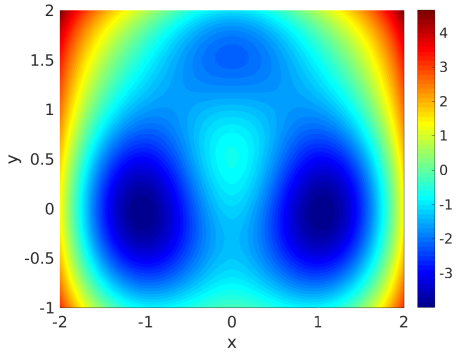



**Figure 2.5:** Triple-well potential given by (2.18).

separates the two deep wells from the shallow well. Here, the eigenfunctions of the Perron–Frobenius operator and the eigenfunctions of the Koopman operator essentially encode the same information. As before, we used EDMD with monomials of order up to and including 10 and 250000 randomly generated test points within the domain $\Omega = [-2, 2] \times [-1, 2]$. Each test point was integrated from $t_0 = 0$ to $t_1 = 0.1$ using a step size of $h = 10^{-5}$. The parameter $\sigma$ was set to 1.09.

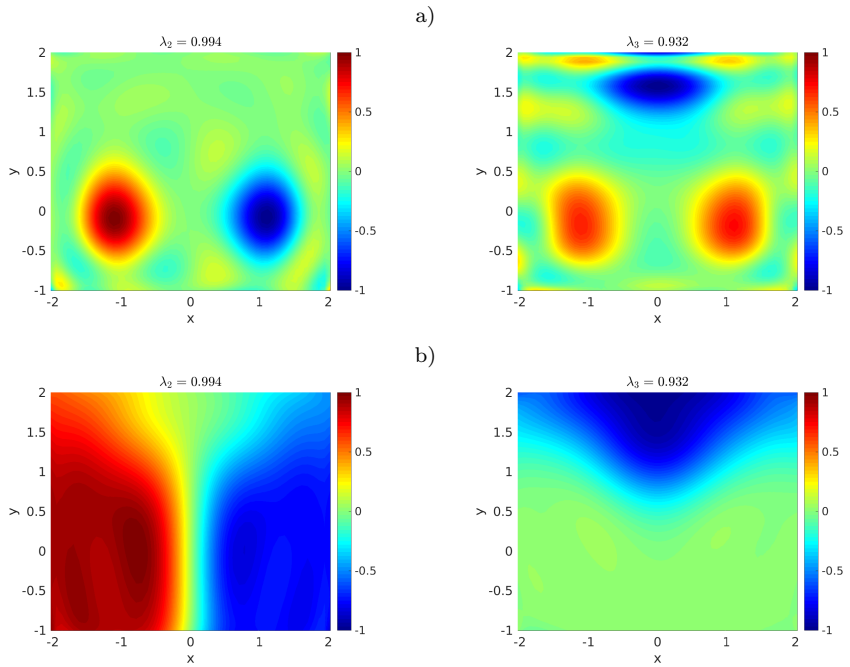



**Figure 2.6:** Second and third eigenfunction of a) the Perron–Frobenius operator $\mathcal{P}$ and b) the Koopman operator $\mathcal{K}$ for the triple-well problem.

### 2.5.3 Molecular dynamics and conformation analysis

**Classical molecular dynamics.** Classical molecular dynamics describes the motion of atoms, or groups of atoms, in terms of Hamiltonian dynamics under the influence of atomic interaction forces resulting from a potential. The *position* or *configuration* space $\mathbb{Q} \subset \mathbb{R}^d$ describes all possible positions of the atoms, while the momentum space $\mathbb{P} = \mathbb{R}^d$ contains all momenta. The potential $V : \mathbb{Q} \to \mathbb{R}$ is assumed to be a sufficiently smooth function. The *phase* space $\mathbb{X} = \mathbb{Q} \times \mathbb{P}$ of the molecule consists of all possible position-momenta pairs $x = (q, p)$. The evolution of a molecule in phase space under ideal conditions is described by Hamilton's equations of motion

$$\begin{aligned} \dot{q}_t &= M^{-1} p_t, \\ \dot{p}_t &= -\nabla V(q_t), \end{aligned} \tag{2.19}$$

where $M$ denotes the symmetric positive definite mass matrix. Since molecules do not stand alone, but are rather subject to interaction with their surrounding molecules, different models incorporating these interactions are more commonly used. One way to account for the collisions with the surrounding molecules is to include a damping and a stochastic forcing term in (2.19) to obtain the *Langevin equation*

$$\begin{aligned} d\boldsymbol{q}_t &= M^{-1} \boldsymbol{p}_t dt, \\ d\boldsymbol{p}_t &= -\nabla V(\boldsymbol{q}_t) dt - \gamma M^{-1} \boldsymbol{p}_t dt + \sigma d\boldsymbol{w}_t. \end{aligned} \tag{2.20}$$

This is an SDE giving rise to a non-deterministic evolution, hence positions and momenta are random variables. Here, $\boldsymbol{w}_t$ is a standard Wiener process in $\mathbb{R}^d$. Further, $\gamma$ and $\sigma$ satisfy the fluctuation-dissipation relation $2\gamma = \beta\sigma\sigma^T$, where $0 < \beta$ is called the inverse temperature. This is due to the fact that $\beta = (k_{\mathrm{B}} T)^{-1}$, where $T$ is the macroscopic temperature of the system, and $k_{\mathrm{B}}$ is the Boltzmann constant. The fluctuation-dissipation relation ensures that the energy of the system is conserved in expectation.

It can also be shown (cf. [119, 171]) that the Langevin process, governed by (2.20), has a unique invariant density with respect to which it is ergodic. This density is also called the *canonical* density, and has the explicit form $f_{\mathrm{can}}(q, p) = Z^{-1} \exp\left(-\beta(\frac{1}{2} p^T M p + V(q))\right)$, where $Z$ is a normalizing constant.

**Spatial transfer operator.** One of the main features of molecules we are interested in is that it has several important geometric forms, called *conformations*, between which it "switches". Hereby it spends "long" times (measured on the time scales of its internal motion) in one conformation, and passes quickly to another. Due to this time scale separation the conformations are called *metastable*. The identification of metastable conformations is of major interest, and it is connected to the sub-dominant eigenfunctions of a special transfer operator which is adapted to the problem at hand [167]: although the more appreciated models describe the dynamics of a molecule in the complete phase space including positions and momenta, metastability is observed (and described) in the positional coordinate only.

This problem-adapted transfer operator is called the *spatial transfer operator* (cf. (2.21) below), and describes positional density fluctuations in a molecule which is in thermal equilibrium. More precisely, if an ensemble of molecules with positional coordinates distributed according to the density $w : \mathbb{Q} \to \mathbb{R}$ with respect to the canonical distribution is given, then its image under the spatial transfer operator with lag time $t$ describes the density of the positional coordinate of the ensemble after time $t$, again with respect to the canonical distribution:

$$\mathcal{S}^t w(q) = \frac{1}{f_{\mathbb{Q}}(q)} \int_{\mathbb{P}} \left(\mathcal{P}^t_{\mathrm{Lan}} w f_{\mathrm{can}}\right)(q,p)\, dp, \tag{2.21}$$

where $f_{\mathbb{Q}}$ is the positional marginal of the canonical density, i.e.

$$f_{\mathbb{Q}}(q) = \int_{\mathbb{P}} f_{\mathrm{can}}(q,p)\, dp,$$

and $\mathcal{P}^t_{\mathrm{Lan}}$ is the transfer operator of the Langevin process governed by (2.20).

The operator $\mathcal{S}^t : L^2(\mathbb{Q}, \mu_{\mathbb{Q}}) \to L^2(\mathbb{Q}, \mu_{\mathbb{Q}})$, where $d\mu_{\mathbb{Q}}(q) = f_{\mathbb{Q}}(q)dq$, is self-adjoint (i.e. has pure real point spectrum), and due to the ergodicity of the Langevin process it possesses the isolated and simple eigenvalue 1 with corresponding eigenfunction $\mathbb{1}_{\mathbb{Q}}$ [9].

With the right chemical intuition at hand the range of positional coordinates possibly interesting for conformation analysis can be drastically reduced to just a handful of *essential coordinates*; as it is shown in Section 2.5.4. The spatial transfer operator can be adapted to this situation, as we describe in Appendix 2.C. There we also show that if we carry out the EDMD procedure in the space of these reduced observables, we actually approximate a Galerkin projection of the corresponding *reduced* spatial transfer operator. A similar technique has been developed in [134, 137]. Chekroun et al [26] also approximate a reduced transfer operator from observable time series from climate models, but only for the case where the basis functions are characteristic functions, as in Ulam's method.

### 2.5.4 $n$-butane

Let us now consider the $n$-butane molecule $H_3C{-}CH_2{-}CH_2{-}CH_3$ shown in Figure 2.7 (drawn with PyMOL [166]). We want to analyze this molecule since the energy landscape and conformations are well-known. The four configurations illustrated in Figure 2.7 can be obtained by rotating around the bond between the second and third carbon atom. The potential energy of a molecule depends on the structure. The higher the potential energy of a conformation, the lower the probability the system will remain in that state. Thus, we would expect a high probability for the anti configuration, a slightly lower probability for the gauche configuration, and low probabilities for the other configurations. Indeed, the anti and gauche configurations are metastable conformations.

Molecular dynamics simulators are standard tools to analyze the conformations and conformational dynamics of biological molecules such as proteins and the extraction of this essential information from molecular dynamics simulations is still an active field of research [138]. We simulated the $n$-butane molecule for an interval of 10 ns with a

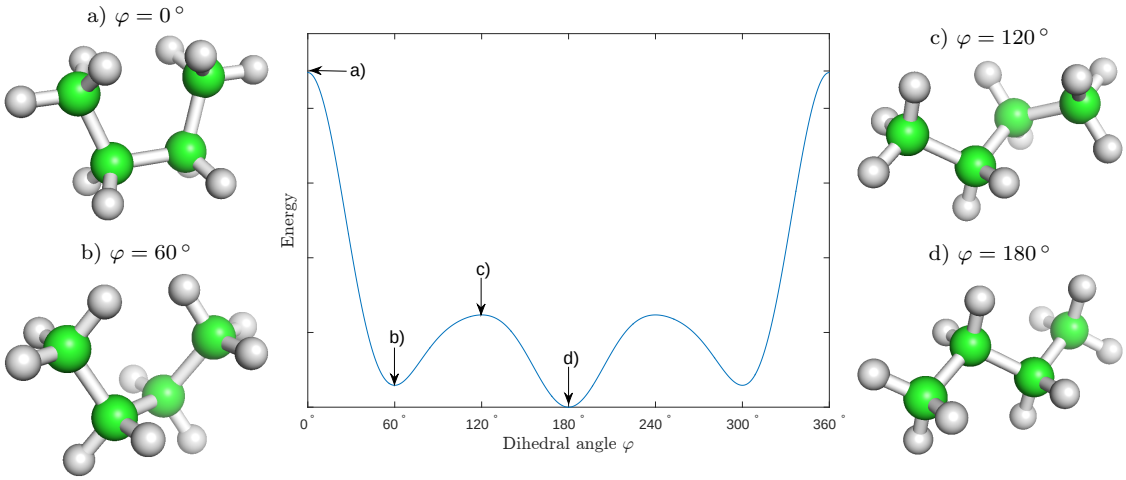



**Figure 2.7:** Potential of the $n$-butane molecule as a function of the dihedral angle $\varphi$ and different conformations. a) Fully eclipsed. b) Gauche. c) Partly eclipsed. d) Anti.

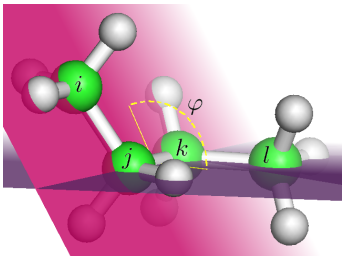



**Figure 2.8:** Definition of the dihedral angle $\varphi$ for the butane molecule.

step size of 2 fs using AmberTools15 [25] and, downsampling by a factor of 100, created one trajectory containing 50,000 data points. From this 42-dimensional trajectory – 3 coordinates for each of the 14 atoms –, we extracted the dihedral angle $\varphi$ shown in Figure 2.8 as

$$\cos\varphi = \frac{n_1 \cdot n_2}{\|n_1\| \, \|n_2\|}, \tag{2.22}$$

where $n_1 = v_{ij} \times v_{jk}$ and $n_2 = v_{lk} \times v_{jk}$ are the vectors perpendicular to the planes spanned by the carbon atoms $i, j, k$ and $j, k, l$, respectively, and $v_{ij}$ is the bond between $i$ and $j$.

In order to compute the dominant eigenfunctions of the spatial transfer operator for this one essential coordinate, we used 41 basis functions $\{1, \cos(i\,x), \sin(i\,x)\}$, $i = 1, \ldots, 20$, for the interval $[0, 2\,\pi]$. The resulting leading eigenfunctions are shown in Figure 2.9. As expected, the first eigenfunction predicts high probabilites for the gauche and anti configurations and low probabilites for the other configurations. The (sign) structure of the second and third eigenfunctions contain information about the metastable sets.

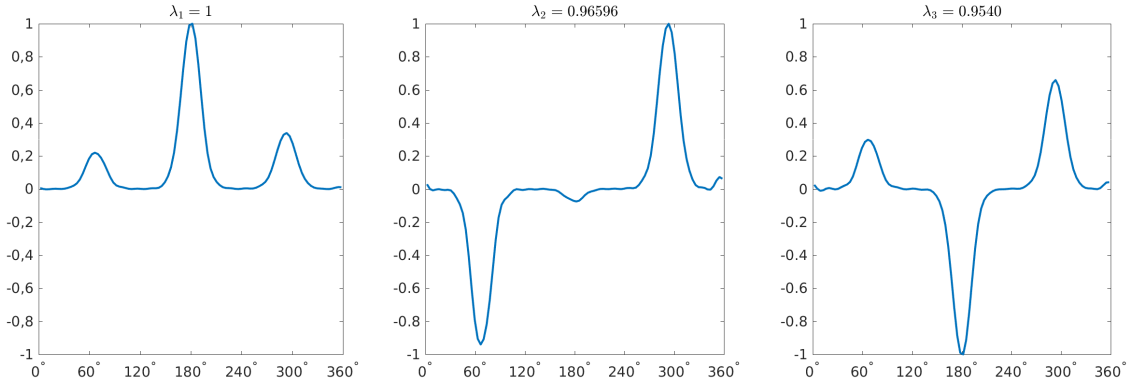



**Figure 2.9:** First three eigenfunctions of the Perron–Frobenius operator obtained from a simulation of the $n$-butane molecule computed with EDMD.

## 2.6 Conclusion

The global behavior of dynamical systems can be analyzed using operator-based approaches. We reviewed and described different, projection-based numerical methods such as Ulam's method and EDMD to compute finite-dimensional approximations of the Perron–Frobenius operator and the Koopman operator. Furthermore, we highlighted the similarities and differences between these methods and showed that methods developed for the approximation of the Koopman operator can be used for the Perron–Frobenius operator, and vice versa. We demonstrated the performance of different methods with the aid of several examples. If the eigenfunctions of the Perron–Frobenius operator or Koopman operator are smooth, EDMD enables an accurate approximation with a small number of basis functions. Thus, this approach is well suited also for higher-dimensional problems.

The next step could be to investigate the possibility of extending the methods reviewed within this paper using tensors as described in [138] for reversible processes. Currently, not all numerical methods required for generalizing these methods to tensor-based methods are available. Nevertheless, developing tensor-based algorithms for these eigenvalue problems might enable the analysis of high-dimensional systems.

## Acknowledgements

This research has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114 and by the Einstein Center for Mathematics Berlin (ECMath), project grant CH7.

## 2.A Adjoint EDMD

### Notation

We will use the same notation as in the main text. More precisely, let us use the set of (linearly independent), piecewise continuous basis functions (dictionary)

$$\mathbb{D} = \{\psi_1, \dots, \psi_k\}, \qquad \psi_i : \mathbb{R}^d \to \mathbb{R},\ i = 1, \dots, k,$$

and let us denote $\mathbb{V} = \operatorname{span}(\mathbb{D})$. For every $f \in \mathbb{R}^k$, we define $\bar{f} = \sum_{i=1}^k f_i \psi_i \in \mathbb{V}$. We also identify a linear operator $A : \mathbb{V} \to \mathbb{V}$ with its matrix representation $A \in \mathbb{R}^{k \times k}$ with respect to the basis $\mathbb{D}$. Here we mean multiplication from the left, i.e. $Kf$ is identified with $K\bar{f}$. Further, let

$$\Psi = \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_k \end{pmatrix},$$

a vector-valued function, and for sets of points (collected column-wise into a $d \times m$ matrix)

$$X = [x_1\ x_2\ \dots\ x_m], \qquad Y = [y_1\ y_2\ \dots\ y_m]$$

define

$$\Psi_X = [\Psi(x_1)\ \Psi(x_2)\ \dots\ \Psi(x_m)], \qquad \Psi_Y = [\Psi(y_1)\ \Psi(y_2)\ \dots\ \Psi(y_m)]\,.$$

### Scalar products

Given $f, g \in \mathbb{R}^k$ and some positive measure $\mu$, such that $|\int \psi_i \psi_j \, d\mu| < \infty$ for all $i, j = 1, \dots, k$, we wish to express the $\mu$-weighted $L^2$ scalar products of elements of $\mathbb{V}$. To this end, we compute

$$\langle \bar{f}, \bar{g} \rangle_\mu = \int \bar{f}\bar{g} \, d\mu = \sum_{i,j=1}^k f_i g_j \int \psi_i \psi_j \, d\mu = f^T S g\,,$$

where $S \in \mathbb{R}^{k \times k}$ with $S_{ij} = \int \psi_i \psi_j \, d\mu$. Since $\mu$ is positive, $S$ is symmetric positive definite, hence invertible.

### Adjoint operator

With this, we are ready to express the adjoint $A^*$ of any (linear) operator $A : \mathbb{V} \to \mathbb{V}$ with respect to the scalar product $\langle \cdot, \cdot \rangle_\mu$. By successive reformulations of the defining equation

for the adjoint, we obtain

$$\begin{aligned}
\langle A\bar{f}, \bar{g}\rangle_\mu &= \langle \bar{f}, A^*\bar{g}\rangle_\mu && \forall \bar{f}, \bar{g} \in \mathbb{V}, \\
&\Updownarrow \\
\langle \sum_{i=1}^{k} (Af)_i \psi_i, \sum_{i=1}^{k} g_i \psi_i\rangle_\mu &= \langle \sum_{i=1}^{k} f_i \psi_i, \sum_{i=1}^{k} (A^*g)_i \psi_i\rangle_\mu && \forall f, g \in \mathbb{R}^k, \\
&\Updownarrow \\
f^T A^T S g &= f^T S A^* g && \forall f, g \in \mathbb{R}^k.
\end{aligned}$$

Thus,

$$A^* = S^{-1} A^T S\,. \tag{2.23}$$

**Remark 2.A.1.** From (2.23) we can see that $A^T$ represents the adjoint of $A$ if $S$ is a multiple of the identity matrix, implying that the basis functions are orthogonal with respect to $\langle\cdot,\cdot\rangle_\mu$. This is the case for Ulam's method, given the boxes have all the same measures.

### The Perron–Frobenius operator

Let $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ be some dynamical system. The following properties hold also, if $\Phi$, such as the basis functions and the measure $\mu$ are restricted to some set $\mathbb{X}$.

Recall equations (2.1) and (2.4), stating that the *Perron–Frobenius operator* $\mathcal{P}_\mu : L^1 \to L^1$ with respect to the measure $\mu$ is (uniquely) defined by

$$\int_{\mathbb{A}} \mathcal{P}_\mu h \, d\mu = \int_{\Phi^{-1}(\mathbb{A})} h \, d\mu, \quad \text{for all measurable } \mathbb{A}\,,$$

and the *Koopman operator* $\mathcal{K} : L^\infty \to L^\infty$ is defined by

$$\mathcal{K}h = h \circ \Phi\,,$$

respectively. They satisfy the duality relation

$$\langle \mathcal{P}_\mu h_1, h_2\rangle_\mu = \langle h_1, \mathcal{K}h_2\rangle_\mu \quad \forall h_1 \in L^1,\, h_2 \in L^\infty\,.$$

We have seen in section 2.3.4, that if the data points satisfy $y_i = \Phi(x_i)$, $i = 1, \ldots, m$, then $K$, with $K^T = \Psi_Y \Psi_X^+$, is a data-based approximation of the Koopman operator. More precisely, in the infinite-data limit $m \to \infty$, $x_i \sim \mu$, the operator $K$ converges to a Galerkin approximation of $\mathcal{K}$ on $\mathbb{V}$ with respect to $\langle\cdot,\cdot\rangle_\mu$. Using (2.15), we can also conclude that

$$\frac{1}{m} \Psi_X \Psi_X^T \to S \text{ as } m \to \infty\,,$$

where $S$ is the symmetric positive definite weight matrix from above. This suggests, using (2.23), that if there is a sufficient amount of data points at hand, then we can approximate the Galerkin projection of the Perron–Frobenius operator $\mathcal{P}_\mu$ to $\mathbb{V}$ by

$$P_\mu = S^{-1} K^T S = (\Psi_X \Psi_X^T)^{-1} \Psi_Y \Psi_X^+ (\Psi_X \Psi_X^T) = (\Psi_X \Psi_X^T)^{-1} \Psi_Y \Psi_X^T\,. \tag{2.24}$$

The same matrix representation has been obtained in equation (2.16), by a different consideration. Note also, that if one can compute the matrix $S$ with $S_{ij} = \int \psi_i \psi_j \, d\rho$ with respect to a different measure $\rho$, the Perron–Frobenius operator with respect to $\rho$ can be approximated as well, one is not restricted to use the empirical distribution $\mu$ of the data points.

**Remark 2.A.2.** All these considerations can be extended to the case where the dynamics $\Phi$ is non-deterministic.

## 2.B On the ergodic behavior of one-step pairs

We will need the result of this section, equation (2.25), in the following section.

Let the non-deterministic dynamical system $\mathbf{\Phi}$ be given with transition density function $k$, that is,

$$\mathsf{P}\left(\mathbf{\Phi}(x) \in \mathbb{A}\right) = \int_{\mathbb{A}} k(x,y)\, d\mu(y), \quad \mathbb{A} \in \mathfrak{B},$$

for a.e. $y \in \mathbb{X}$. Further, let $f$ denote the unique invariant density of $\mathbf{\Phi}$,

$$\int f(x)k(x,y)\, d\mu(x) = f(y) \quad \text{for a.e. } y \in \mathbb{X},$$

with respect to which $\mathbf{\Phi}$ is geometrically ergodic. Geometric ergodicity of the Langevin process (2.20) has been established in [119].

For $\phi, \psi \in L^2(\mathbb{X})$ we wish to determine the ergodic limit

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \phi\big(\mathbf{\Phi}^n(x)\big)\psi\big(\mathbf{\Phi}^{n+1}(x)\big)\,.$$

To this end, we consider the non-deterministic dynamical system $\mathbf{\Psi} : \mathbb{X}\times\mathbb{X} \to \mathbb{X}\times\mathbb{X}$ with

$$\mathbf{\Psi} : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} y \\ \mathbf{\Phi}(y) \end{pmatrix}.$$

In order to find the transition density function of $\mathbf{\Psi}$, note that

$$\mathsf{P}\big(\mathbf{\Psi}(x,y) \in \mathbb{A}\times\mathbb{B}\big) = \mathbb{1}_{\mathbb{A}}(y) \int_{\mathbb{B}} k(y,z)\, d\mu(z) = \int_{\mathbb{A}\times\mathbb{B}} \delta_y(u)k(u,z)\, d\mu(u)\, d\mu(z)\,,$$

yielding $k_{\mathbf{\Psi}}((x,y),(u,z)) = \delta_y(u)k(u,z)$ as the transition density function of $\mathbf{\Psi}$. From this we immediately find its invariant density.

**Lemma 2.B.1.** *The density $f(x)k(x,y)$ is invariant under $\mathbf{\Psi}$.*

*Proof.* Direct computation shows

$$\iint f(x)k(x,y)k_{\mathbf{\Psi}}((x,y),(u,z))\, d\mu(x)\, d\mu(y)$$

$$
\begin{aligned}
&= \iint f(x)k(x,y)\delta_y(u)k(u,z)\,d\mu(x)\,d\mu(y) \\
&= k(u,z)\int f(x)\int k(x,y)\delta_y(u)\,d\mu(y)\,d\mu(x) \\
&= k(u,z)\int f(x)k(x,u)\,d\mu(x) \\
&= f(u)k(u,z),
\end{aligned}
$$

the last equality following from the invariance of $f$ under $\mathbf{\Phi}$. □

Geometric ergodicity of $\mathbf{\Phi}$ with respect to $f$ implies ergodicity of $\mathbf{\Psi}$ with respect to $f(x)k(x,y)$. Thus, for $\zeta \in L^2(\mathbb{X}\times\mathbb{X})$ we have

$$
\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}\zeta\big(\mathbf{\Psi}^n(x)\big) = \iint \zeta(x,y)f(x)k(x,y)\,d\mu(x)\,d\mu(y)\,.
$$

With $\zeta(x,y)=\phi(x)\psi(y)$ this implies

$$
\begin{aligned}
&\lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}\phi\big(\mathbf{\Phi}^n(x)\big)\psi\big(\mathbf{\Phi}^{n+1}(x)\big) \\
&\qquad = \lim_{N\to\infty}\frac{1}{N}\sum_{n=0}^{N-1}\zeta\big(\mathbf{\Psi}^n(x)\big) \\
&\qquad = \iint \zeta(x,y)f(x)k(x,y)\,d\mu(x)\,d\mu(y) \\
&\qquad = \iint \phi(x)f(x)\psi(y)k(x,y)\,d\mu(x)\,d\mu(y) \\
&\qquad = \int \psi(y)\int (\phi(x)f(x))k(x,y)\,d\mu(x)\,d\mu(y) \\
&\qquad = \int \psi\,\mathcal{P}(\phi f)\,d\mu\,,
\end{aligned}
\tag{2.25}
$$

where the last equality follows from (2.2), the definition of the Perron–Frobenius operator.

## 2.C EDMD for the reduced spatial transfer operator

We shall first discuss the restriction of the spatial transfer operator, introduced in (2.21), to a collection of coordinates which we assume to be sufficient to describe the metastable behavior of the system. Let $\xi : \mathbb{Q} \to \mathbb{U} \subset \mathbb{R}^r$ be a smooth, possibly nonlinear mapping of the configuration variable $q$ to these so-called *essential* (or *reduced*) coordinates. For instance, in case of $n$-butane in Section 2.5.4 we have $r = 1$ and $\xi$ describes the mapping $q \mapsto \varphi$ given implicitly by (2.22). Let $\xi$ have the property that for every regular value $z \in \mathbb{U}$ of $\xi$,

$$
\mathbb{M}_z := \{q \in \mathbb{Q} \,|\, \xi(q) = z\} \subset \mathbb{Q}
$$

is a smooth, codimension $r$ manifold. We suppose that $\xi$ is a physically relevant observable of the dynamics, e.g. a reaction coordinate.

To define the spatial transfer operator for the essential coordinates, we need a non-linear variant of Fubini's theorem, the so-called *coarea formula* [58, Section 3.2]. For an integrable function $h : \mathbb{Q} \to \mathbb{R}$ it holds

$$\int_{\mathbb{Q}} h(q)\, dq = \int_{\mathbb{U}} \left( \int_{\mathbb{M}_z} hG\, d\sigma_z \right) dz, \tag{2.26}$$

where $G(q) = \left|\det \nabla\xi^T \nabla\xi\right|^{-1/2}$ is the Gramian, and $d\sigma_z$ denotes the Riemannian volume element on $\mathbb{M}_z$. It follows that the (marginal) canonical density for the observable $\xi$ is

$$f_{\mathbb{U}}(z) = \iint_{\mathbb{M}_z \times \mathbb{P}} f_{\text{can}} G\, d\sigma_z\, dp\,.$$

Thus, the spatial transfer operator for the essential coordinates given by $\xi$ reads as

$$\mathcal{S}^t_{\text{ess}} w(z) = \frac{1}{f_{\mathbb{U}}(z)} \iint_{\mathbb{M}_z \times \mathbb{P}} \mathcal{P}^t_{\text{Lan}}(f_{\text{can}} \cdot w \circ \xi) G\, d\sigma_z\, dp\,, \tag{2.27}$$

for $w \in L^2(\mathbb{U}, \mu_{\mathbb{U}})$, with $d\mu_{\mathbb{U}}(z) = f_{\mathbb{U}}(z)dz$.

To see what EDMD does with the molecular trajectory data, we have to consider the limit

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \phi(\xi(q_n))\psi(\xi(q_{n+1})),$$

where $q_0, q_1, q_2, \ldots$ are the positional coordinates of the time-$t$-sampled simulation, and $\phi, \psi : \mathbb{U} \to \mathbb{R}$ are basis functions. We know that the Langevin dynamics is ergodic with respect to the canonical density [119], hence (2.25) yields

$$\begin{aligned}
\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \phi(\xi(q_n))\psi(\xi(q_{n+1})) &= \iint_{\mathbb{Q}\times\mathbb{P}} \psi(\xi(q)) \mathcal{P}^t_{\text{Lan}} \left(f_{\text{can}} \cdot \phi \circ \xi\right)(q,p)\, dq\, dp \\
&= \int_{\mathbb{Q}} \psi(\xi(q)) \int_{\mathbb{P}} \mathcal{P}^t_{\text{Lan}} \left(f_{\text{can}} \cdot \phi \circ \xi\right)(q,p)\, dq\, dp \\
&\overset{(2.26)}{=} \int_{\mathbb{U}} \psi(z) \iint_{\mathbb{M}_z\times\mathbb{P}} \mathcal{P}^t_{\text{Lan}} \left(f_{\text{can}} \cdot \phi \circ \xi\right)(q,p) G(q)\, dp\, d\sigma_z dz \\
&\overset{(2.27)}{=} \int_{\mathbb{U}} \psi(z) f_{\mathbb{U}}(z) S^t_{\text{ess}} \phi(z)\, dz \\
&= \left\langle \psi,\, S^t_{\text{ess}} \phi \right\rangle_{\mu_{\mathbb{U}}}.
\end{aligned}$$

Due to ergodicity it follows also

$$\lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \phi(\xi(q_n))\psi(\xi(q_n)) = \iint_{\mathbb{Q}\times\mathbb{P}} \psi(\xi(q))\phi(\xi(q)) f_{\text{can}}(q,p)\, dq\, dp$$

$$\overset{(2.26)}{=} \int_{\mathbb{U}} \psi(z)\phi(z) \iint_{\mathbb{M}_z \times \mathbb{P}} f_{\mathrm{can}}(q,p) G(q)\, dp\, d\sigma_z\, dz$$
$$= \langle \psi,\, \phi \rangle_{\mu_{\mathbb{U}}}\,.$$

Comparing with (2.15), we thus see that EDMD converges in the infinite-data limit to a Galerkin projection in $L^2(\mathbb{U}, \mu_{\mathbb{U}})$ of the spatial transfer operator for the essential coordinates given by $\xi$.

## 2.D Derivation of the EDMD-discretized Koopman operator

Let the finite dictionary $\mathbb{D} = \{\psi_1, \ldots, \psi_k\}$ of piecewise continuous functions be given, and define $\mathbb{V}$ to be the linear space spanned by $\mathbb{D}$. We will give a step-by-step derivation of the matrix representation of the EDMD-discretized Koopman operator $K : \mathbb{V} \to \mathbb{V}$ with respect to the basis $\mathbb{D}$. Let us denote also with $K \in \mathbb{R}^{k\times k}$ this matrix representation, and note that the matrix $K$ acts by multiplication from the left, i.e. if the vector $c \in \mathbb{R}^k$ represents the function $\sum_i c_i \psi_i$, then $K\,c$ represents its image under the discrete Koopman operator. Recall that $\psi : \mathbb{X} \to \mathbb{R}^k$ denotes the column-vector valued function with $[\psi(x)]_i = \psi_i(x)$.

Now, EDMD is an over-determined Petrov–Galerkin method (2.7),

$$\sum_{\ell=1}^{l} \sum_{i=1}^{k} \langle \delta_{x_\ell},\, \mathcal{K}\psi_i - K\psi_i \rangle^2 = \min!\,, \tag{2.28}$$

where $x_1 \ldots, x_l$ are the initial data points and $y_1, \ldots, y_l$ denote their images under the dynamics. If there was just one single data point $x_\ell$, we would like to find a matrix $K$ satisfying the equation

$$\left\langle \delta_{x_\ell},\, \mathcal{K}\Big(\sum_i c_i\, \psi_i\Big) \right\rangle = \left\langle \delta_{x_\ell},\, \sum_i \Big(\sum_j K_{ij}\, c_j\Big)\psi_i \right\rangle$$

for every $c \in \mathbb{R}^k$. Rearranging the terms and using $\mathcal{K}\psi_i(x_\ell) = \psi_i(y_\ell)$ yields

$$\sum_i c_i\, \psi_i(y_\ell) = \sum_j c_j \sum_i K_{ij}\, \psi_i(x_\ell)\,,$$

or, in vectorial notation, $c^T\, \psi(y_\ell) = c^T\, K^T \psi(x_\ell)$. Since this has to hold true for every $c \in \mathbb{R}^k$, we have $\psi(y_\ell) = K^T \psi(x_\ell)$. From this it follows by putting the column vectors $\psi(x_\ell)$ and $\psi(y_\ell)$ side-by-side for multiple data points $x_\ell$ to form the matrices $\Psi_X$ and $\Psi_Y$, respectively, that (2.28) is equivalent with

$$\big\|\Psi_Y - K^T \Psi_X\big\|_F^2 = \min!,$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Thus, EDMD can be viewed as a DMD of the transformed data $\Psi_X$ and $\Psi_Y$. The solution of the minimization problem is given by

$$K^T = \Psi_Y \Psi_X^+,$$

where $\Psi_X^+$ is the pseudoinverse of $\Psi_X$.

# 3

# Data-driven model reduction and transfer operator approximation

Stefan Klus[1], Feliks Nüske[1], Péter Koltai[1], Hao Wu[1], Ioannis Kevrekidis[2,3], Christof Schütte[1,3], and Frank Noé[1]

[1] Department of Mathematics and Computer Science, Freie Universität Berlin, Germany
[2] Department of Chemical and Biological Engineering & Program in Applied and Computational Mathematics, Princeton University, USA
[3] Zuse Institute Berlin, Germany

*Journal of Nonlinear Science*, 2018

DOI: 10.1007/s00332-017-9437-7

**Abstract**

In this review paper, we will present different data-driven dimension reduction techniques for dynamical systems that are based on transfer operator theory as well as methods to approximate transfer operators and their eigenvalues, eigenfunctions, and eigenmodes. The goal is to point out similarities and differences between methods developed independently by the dynamical systems, fluid dynamics, and molecular dynamics communities such as *time-lagged independent component analysis* (TICA), *dynamic mode decomposition* (DMD), and their respective generalizations. As a result, extensions and best practices developed for one particular method can be carried over to other related methods.

## 3.1 Introduction

The numerical solution of complex systems of differential equations plays an important role in many areas such as molecular dynamics, fluid dynamics, mechanical as well as electrical engineering, and physics. These systems often exhibit multi-scale behavior which can be due to the coupling of subsystems with different time scales – for instance, fast electrical and slow mechanical components – or due to the intrinsic properties of the system itself – for instance, the fast vibrations and slow conformational changes of molecules. Analyzing such problems using transfer operator based methods is often infeasible or prohibitively expensive from a computational point of view due to the so-called *curse of dimensionality*. One possibility to avoid this is to project the dynamics of the

high-dimensional system onto a lower-dimensional space and to then analyze the reduced system representing, for instance, only the relevant slow dynamics, see, e.g., [150, 64].

In this paper, we will introduce different methods such as *time-lagged independent component analysis* (TICA) [128, 150, 172] and *dynamic mode decomposition* (DMD) [164, 28, 186, 104] to identify the dominant dynamics using only simulation data or experimental data. It was shown that these methods are related to Koopman operator approximation techniques [97, 126, 159, 22]. Extensions of the aforementioned methods called the *variational approach of conformation dynamics* (VAC) [134, 137, 138] developed mainly for reversible molecular dynamics problems and *extended dynamic mode decomposition* (EDMD) [195, 196, 92] can be used to compute eigenfunctions, eigenvalues, and eigenmodes of the Koopman operator (and its adjoint, the Perron–Frobenius operator). Interestingly, although the underlying ideas, derivations, and intended applications of these methods differ, the resulting algorithms share a lot of similarities. The goal of this paper is to show the equivalence of different data-driven methods which have been widely used by the dynamical systems, fluid dynamics, and molecular dynamics communities, but under different names. Hence, extensions, generalizations, and algorithms developed for one method can be carried over to its counterparts, resulting in a unified theory and set of tools. An alternative approach to data-driven model reduction – also related to transfer operators and their generators – would be to use *diffusion maps* [132, 32, 59, 74]. Manifold learning methods, however, are beyond the scope of this paper.

The outline of the paper is as follows: Section 3.2 briefly introduces transfer operators and the concept of reversibility. In Section 3.3, different data-driven methods for the approximation of the eigenvalues, eigenfunctions, and eigenmodes of transfer operators will be described. The theoretical background and the derivation of these methods will be outlined in Section 3.4. Section 3.5 addresses open problems and lists possible future work.

## 3.2 Transfer operators and reversibility

In the literature, the term *transfer operator* is sometimes used in different contexts. In this section, we will briefly introduce the Perron–Frobenius operator, the Perron–Frobenius operator with respect to the equilibrium density, and the Koopman operator. All these three operators are, according to our definition, transfer operators.

### 3.2.1 Guiding example

Our paper will deal with data-driven methods to analyze both stochastic and deterministic dynamical systems. To illustrate the concepts of transfer operators and their spectral components, we first introduce a simple stochastic dynamical system that will be revisited throughout the paper.

**Example 3.2.1.** Consider the following one-dimensional Ornstein–Uhlenbeck process,

given by an Itô stochastic differential equation[1] of the form:

$$\mathrm{d}\boldsymbol{X}_t = -\alpha D\, \boldsymbol{X}_t\, \mathrm{d}t + \sqrt{2D}\, \mathrm{d}\boldsymbol{W}_t.$$

Here, $\{W_t\}_{t\geq 0}$ is a one-dimensional standard Wiener process (Brownian motion), the parameter $\alpha$ is the friction coefficient, and $D = \beta^{-1}$ is the diffusion coefficient. The stochastic forcing usually models physical effects, most often thermal fluctuations and it is customary to call $\beta$ the inverse temperature.

The transition density of the Ornstein–Uhlenbeck process, i.e., the conditional probability density to find the process near $y$ a time $\tau$ after it had been at $x$, is given by

$$p_\tau(x,y) = \frac{1}{\sqrt{2\,\pi\,\sigma^2(\tau)}} \exp\left(-\frac{\left(y - x\,e^{-\alpha D\tau}\right)^2}{2\,\sigma^2(\tau)}\right), \tag{3.1}$$

where $\sigma^2(\tau) = \alpha^{-1}\left(1 - e^{-2\alpha D\tau}\right)$. Figure 3.1a shows the transition densities for different values of $\tau$. More details can be found in [147]. For complex dynamical systems, the transition density is not known explicitly, but must be estimated from simulation or measurement data.

In this work we will describe the dynamics of a system in terms of dynamical operators such as the propagator $\mathcal{P}_\tau$, which is defined by the transition density $p_\tau(x,y)$ and propagates a probability density of Brownian walkers in time by

$$p_{t+\tau}(x) = \mathcal{P}_\tau\, p_t(x).$$

See Figure 3.1b for the time evolution of the Ornstein–Uhlenbeck process initiated from a localized starting condition. It can be seen that the distribution spreads out and converges towards a Gaussian distribution, which is then invariant in time. For this simple dynamical system we can give the equation for the invariant density explicitly:

$$\pi(x) = \frac{1}{\sqrt{2\,\pi\,\alpha^{-1}}} \exp\left(-\frac{x^2}{2\,\alpha^{-1}}\right), \tag{3.2}$$

which is a Gaussian whose variance is decreasing with increasing friction and decreasing temperature. △

### 3.2.2 Transfer operators

Let $\{\mathbf{X}_t\}_{t\geq 0}$ be a time-homogeneous[2] stochastic process defined on the bounded state space $\mathbb{X} \subset \mathbb{R}^d$. It can be genuinely stochastic or it might as well be deterministic, such

[1] A general time-homogeneous Itô stochastic differential equation is given by $\mathrm{d}\mathbf{X}_t = -\alpha(\mathbf{X}_t)\,\mathbf{X}_t\,\mathrm{d}t + \sigma(\mathbf{X}_t)\,\mathrm{d}\mathbf{W}_t$, where $\alpha : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ are coefficient functions, and $\{\mathbf{W}_t\}_{t\geq 0}$ is a $d$-dimensional standard Wiener process.

[2] We call a stochastic process $\{\mathbf{X}_t\}_{t\geq 0}$ *time-homogeneous*, or *autonomous*, if it holds for every $t \geq s \geq 0$ that the distribution of $\mathbf{X}_t$ conditional to $\mathbf{X}_s = x$ only depends on $x$ and $(t-s)$. It is the stochastic analogue of the flow of an autonomous (time-independent) ordinary differential equation.

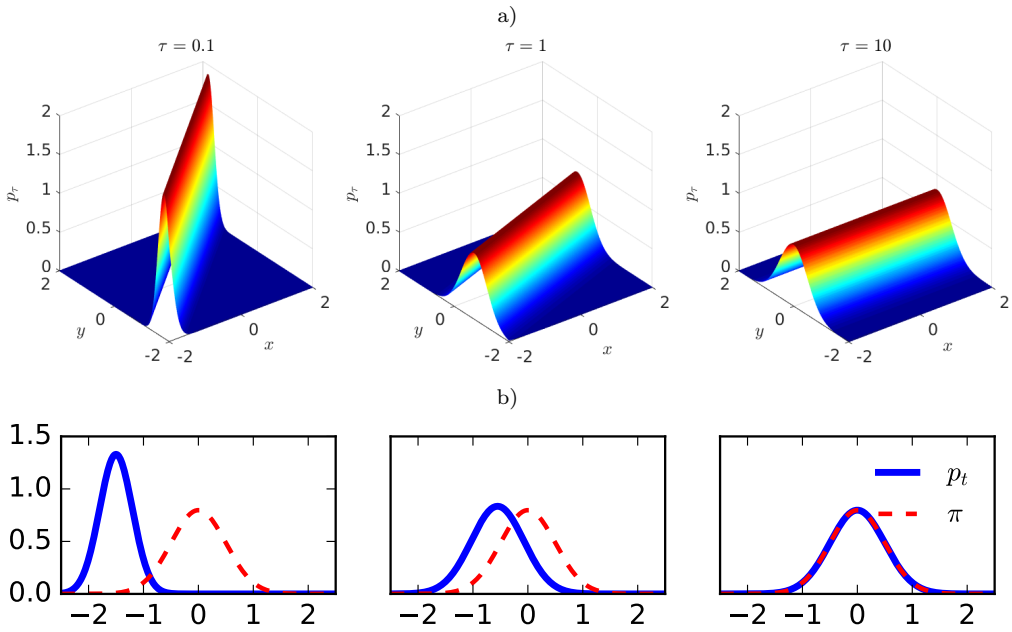



**Figure 3.1:** a) Transition density function of the Ornstein–Uhlenbeck process for different values of $\tau$. If $\tau$ is small, values starting in $x$ will stay close to it. For larger values of $\tau$, the influence of the starting point $x$ is negligible. Densities converge to the equilibrium density, denoted by $\pi$. Here, $\alpha = 4$ and $D = 0.25$. b) Evolution of the probability to find the dynamical system at any point $x$ over time $t$, after starting with a peaked distribution at $t = 0$. We show the resulting distributions at times $t = 0.1$, and $t = 1$, and $t = 10$. The system relaxes towards the stationary density $\pi(x)$.

that there is a flow map $\Phi_\tau : \mathbb{X} \to \mathbb{X}$ with $\Phi_\tau(\mathbf{X}_t) = \mathbf{X}_{t+\tau}$ for $\tau \geq 0$. Let the measure[3] $\mathbb{P}$ denote the law of the process $\{\mathbf{X}_t\}_{t\geq 0}$ that we will study in terms of its statistical transition properties. To this end, under some mild regularity assumptions[4] which are satisfied by Itô diffusions with smooth coefficients [81, 99], we can give the following definition.

**Definition 3.2.2.** *The transition density function* $p_\tau : \mathbb{X} \times \mathbb{X} \to [0, \infty]$ *of a process* $\{\mathbf{X}_t\}_{t\geq 0}$ *is defined by*

$$\mathbb{P}[\mathbf{X}_{t+\tau} \in \mathbb{A} \,|\, \mathbf{X}_t = x] = \int_{\mathbb{A}} p_\tau(x, y)\, \mathrm{d}y,$$

*for every measurable set* $\mathbb{A}$. *Here and in what follows,* $\mathbb{P}[\,\cdot \,|\, \mathfrak{E}]$ *denotes probabilities conditioned on the event* $\mathfrak{E}$. *That is,* $p_\tau(x, y)$ *is the conditional probability density of* $\mathbf{X}_{t+\tau} = y$

[3] For a measure-theoretic discussion of this construction, please refer to [92]. For our purposes, it is sufficient to equip $\mathbb{X}$ with the standard Lebesgue measure. In particular, if not stated otherwise, measurability of a set $\mathbb{A} \subset \mathbb{X}$ is meant with respect to the Borel $\sigma$-algebra.

[4] These conditions are called interchangeably *absolute continuity*, *$\mu$-compatibility*, or *null preservingness*.

*given that* $\mathbf{X}_t = x$.

For $1 \leq r \leq \infty$, the spaces $L^r(\mathbb{X})$ denote the usual spaces of $r$-Lebesgue integrable functions, which is a Banach space with the corresponding norm $\|\cdot\|_{L^r}$.

**Definition 3.2.3.** *Let $p_t \in L^1(\mathbb{X})$ be the probability density and $f_t \in L^\infty(\mathbb{X})$ an observable of the system. For a given lag time $\tau$:*

*a) The* Perron–Frobenius operator *or* propagator $\mathcal{P}_\tau : L^1(\mathbb{X}) \to L^1(\mathbb{X})$ *is defined by*

$$\mathcal{P}_\tau p_t(x) = \int_{\mathbb{X}} p_\tau(y, x)\, p_t(y)\, \mathrm{d}y.$$

*b) The* Koopman operator $\mathcal{K}_\tau : L^\infty(\mathbb{X}) \to L^\infty(\mathbb{X})$ *is defined by*

$$\mathcal{K}_\tau f_t(x) = \int_{\mathbb{X}} p_\tau(x, y)\, f_t(y)\, \mathrm{d}y = \mathbb{E}[f_t(\mathbf{X}_{t+\tau}) \,|\, \mathbf{X}_t = x].$$

Both $\mathcal{P}_\tau$ and $\mathcal{K}_\tau$ are linear but infinite-dimensional operators which are adjoint to each other with respect to the standard duality pairing $\langle \cdot, \cdot \rangle$, given by $\langle f, g \rangle = \int_{\mathbb{X}} f(x)\, g(x)\, dx$. The homogeneity of the stochastic process $\{\mathbf{X}_t\}_{t \geq 0}$ implies the so-called *semigroup property* of the operators, i.e., $\mathcal{P}_{\tau+\sigma} = \mathcal{P}_\tau \mathcal{P}_\sigma$ and $\mathcal{K}_{\tau+\sigma} = \mathcal{K}_\tau \mathcal{K}_\sigma$ for $\tau, \sigma \geq 0$. In other words, these operators describe time-stationary Markovian dynamics. While the Perron–Frobenius operator describes the evolution of densities, the Koopman operator describes the evolution of observables. For the analysis of the long-term behavior of dynamical systems, densities that remain unchanged by the dynamics play an important role (one can think of the concept of *ergodicity*).

**Definition 3.2.4.** *A density $\pi$ is called an* invariant density *or* equilibrium density *if $\mathcal{P}_\tau\, \pi = \pi$. That is, the equilibrium density $\pi$ is an eigenfunction of the Perron–Frobenius operator $\mathcal{P}_\tau$ with corresponding eigenvalue* 1.

In what follows, $L^r_\pi(\mathbb{X})$ denotes the weighted $L^r$-space of functions $f$ such that $\|f\|_{L^r_\pi} := \int_{\mathbb{X}} |f(x)|^r \pi(x)\, dx < \infty$. While one can consider the evolution of densities with respect to any density, we are particularly interested in the evolution with respect to the equilibrium density. From this point on, we assume there is a unique invariant density. This assumption is typically satisfied for molecular dynamics applications, where the invariant density is given by the Boltzmann distribution.

**Definition 3.2.5.** *Let $L^1_\pi(\mathbb{X}) \ni u_t(x) = \pi(x)^{-1}\, p_t(x)$ be a probability density with respect to the equilibrium density $\pi$. Then the* Perron–Frobenius operator (propagator) with respect to the equilibrium density*, denoted by $\mathcal{T}_\tau$, is defined by*

$$\mathcal{T}_\tau u_t(x) = \int_{\mathbb{X}} \frac{\pi(y)}{\pi(x)}\, p_\tau(y, x)\, u_t(y)\, \mathrm{d}y.$$

The operators $\mathcal{P}_\tau$ and $\mathcal{K}_\tau$ can be defined on other spaces $L^r$ and $L^{r'}$, with $r \neq 1$ and $r' \neq \infty$, see [3, 92] for more details. By defining the weighted duality pairing $\langle f, g \rangle_\pi = \int_{\mathbb{X}} f(x)\, g(x)\, \pi(x)\, dx$ for $f \in L^r_\pi(\mathbb{X})$ and $g \in L^{r'}_\pi(\mathbb{X})$, where $\frac{1}{r} + \frac{1}{r'} = 1$, $\mathcal{T}_\tau$ defined on $L^{r'}_\pi(\mathbb{X})$ is the adjoint of $\mathcal{K}_\tau$ defined on $L^r_\pi(\mathbb{X})$ with respect to $\langle \cdot, \cdot \rangle_\pi$:

$$\langle \mathcal{K}_\tau f,\, g \rangle_\pi = \langle f,\, \mathcal{T}_\tau g \rangle_\pi.$$

For more details, see [107, 168, 67, 134, 137, 92, 198]. The two operators $\mathcal{P}_\tau$ and $\mathcal{T}_\tau$ are often referred to as *forward operators*, whereas $\mathcal{K}_\tau$ is also called *backward operator*, as they are the solution operators of the forward (Fokker–Planck) and backward Kolmogorov equations [107, Section 11], respectively.

### 3.2.3 Spectral decomposition of transfer operators

In what follows, let $\mathcal{A}_\tau$ denote one of the transfer operators defined above, i.e., $\mathcal{P}_\tau$, $\mathcal{T}_\tau$, or $\mathcal{K}_\tau$. We are particularly interested in computing eigenvalues $\lambda_\ell(\tau) \in \mathbb{C}$ and eigenfunctions $\varphi_\ell : \mathbb{X} \to \mathbb{C}$ of transfer operators, i.e.:

$$\mathcal{A}_\tau \varphi_\ell = \lambda_\ell(\tau)\, \varphi_\ell.$$

Note that the eigenvalues depend on the lag time $\tau$. For the sake of simplicity, we will often omit this dependency. The eigenvalues and eigenfunctions of transfer operators contain important information about the global properties of the system such as metastable sets or fast and slow processes and can also be used as reduced coordinates, see [40, 168, 126, 171, 67, 64] and references therein.

**Example 3.2.6.** The eigenvalues $\lambda_\ell$ and eigenfunctions $\varphi_\ell$ of $\mathcal{K}_\tau$ associated with the Ornstein–Uhlenbeck process introduced in Example 3.2.1 are given by

$$\lambda_\ell(\tau) = e^{-\alpha D\,(\ell-1)\,\tau}, \quad \varphi_\ell(x) = \frac{1}{\sqrt{(\ell-1)!}}\, H_{\ell-1}\left(\sqrt{\alpha}\, x\right), \quad \ell = 1, 2, \ldots,$$

where $H_\ell$ denotes the $\ell$th probabilists' Hermite polynomial [147]. The eigenfunctions of $\mathcal{P}_\tau$ are given by the eigenfunctions of $\mathcal{K}_\tau$ multiplied by the equilibrium density $\pi$, see also Figure 3.2. △

In addition to the eigenvalues and eigenfunctions, an essential part of the Koopman operator analysis is the set of *Koopman modes* for the so-called *full-state observable* $g(x) = x$. The Koopman modes are vectors that, together with the eigenvalues and eigenfunctions, allow us to reconstruct and to propagate the system's state [195]. More precisely, assume that each component $g_i$ of the full-state observable, i.e., $g_i(x) = x_i$ for $i = 1, \ldots, d$, can be written in terms of the eigenfunctions as $g_i(x) = \sum_\ell \varphi_\ell(x)\, \eta_{i\ell}$. Defining the Koopman modes by $\eta_\ell = [\eta_{1\ell}, \ldots, \eta_{d\ell}]^T$, we obtain $g(x) = x = \sum_\ell \varphi_\ell(x)\, \eta_\ell$ and thus

$$\mathcal{K}_\tau g(x) = \mathbb{E}[g(\mathbf{X}_\tau) \,|\, \mathbf{X}_0 = x] = \mathbb{E}[\mathbf{X}_\tau \,|\, \mathbf{X}_0 = x] = \sum_\ell \lambda_\ell(\tau)\, \varphi_\ell(x)\, \eta_\ell. \tag{3.3}$$

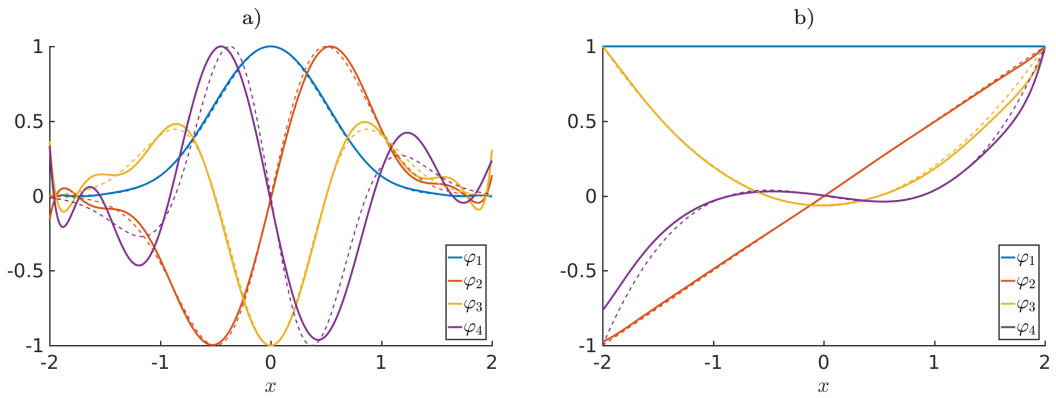



**Figure 3.2:** Dominant eigenfunctions of the Ornstein–Uhlenbeck process computed analytically (dotted lines) and using VAC/EDMD (solid lines). a) Eigenfunctions of the Perron–Frobenius operator $\mathcal{P}_\tau$. b) Eigenfunctions of the Koopman operator $\mathcal{K}_\tau$.

For vector-valued functions, the Koopman operator is defined to act componentwise. In order to be able to compute eigenvalues, eigenfunctions, and eigenmodes numerically, we project the infinite-dimensional operators onto finite-dimensional spaces spanned by a given set of basis functions. This will be described in detail in Section 3.4.

### 3.2.4 Reversibility

We briefly recapitulate the properties of reversible systems. For many applications, including commonly used molecular dynamics models, the dynamics in full phase space are known to be reversible.

**Definition 3.2.7.** *A system is said to be reversible if the so-called detailed balance condition is fulfilled, i.e., it holds for all $x, y \in \mathbb{X}$ that*

$$\pi(x)\, p_\tau(x, y) = \pi(y)\, p_\tau(y, x). \tag{3.4}$$

**Example 3.2.8.** The Ornstein–Uhlenbeck process is reversible. It is straightforward to verify that (3.4) is fulfilled by the transition density (3.1) and the stationary density (3.2) for all values of $x$, $y$ and $\tau$. Also general Smoluchowski equations of a $d$-dimensional system of the form

$$\mathrm{d}\mathbf{X}_t = -D\nabla V(\mathbf{X}_t)\,\mathrm{d}t + \sqrt{2dD}\,\mathrm{d}\mathbf{W}_t$$

with dimensionless potential $V(x)$ are reversible. The stationary density is then given by $\pi \propto \exp(-V(x))$ [110]. △

As a result of the detailed balance condition, the Koopman operator $\mathcal{K}_\tau$ and the Perron–Frobenius operator with respect to the equilibrium density, $\mathcal{T}_\tau$, are identical (hence also self-adjoint):

$$\mathcal{K}_\tau f = \int_{\mathbb{X}} p_\tau(x, y)\, f(y)\,\mathrm{d}y = \int_{\mathbb{X}} \frac{\pi(y)}{\pi(x)}\, p_\tau(y, x)\, f(y)\,\mathrm{d}y = \mathcal{T}_\tau f.$$

Moreover, both $\mathcal{K}_\tau$ and $\mathcal{P}_\tau$ become self-adjoint with respect to the stationary density, i.e.

$$\langle \mathcal{P}_\tau f,\, g\rangle_{\pi^{-1}} = \langle f,\, \mathcal{P}_\tau g\rangle_{\pi^{-1}},$$
$$\langle \mathcal{K}_\tau f,\, g\rangle_{\pi} = \langle f,\, \mathcal{K}_\tau g\rangle_{\pi}.$$

Hence, the eigenvalues $\lambda_\ell$ are real and the eigenfunctions $\varphi_\ell$ of $\mathcal{K}_\tau$ form an orthogonal basis with respect to $\langle\cdot,\,\cdot\rangle_\pi$. That is, the eigenfunctions can be scaled so that $\langle \varphi_\ell,\, \varphi_{\ell'}\rangle_\pi = \delta_{\ell\ell'}$. Furthermore, the leading eigenvalue $\lambda_1$ is the only eigenvalue with absolute value 1 and we obtain

$$1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \dots,$$

see, e.g., [137]. We can then expand a function $f \in L^2_\pi(\mathbb{X})$ in terms of the eigenfunctions as $f = \sum_{\ell=1}^{\infty} \langle f,\, \varphi_\ell\rangle_\pi\, \varphi_\ell$ such that

$$\mathcal{K}_\tau f = \sum_{\ell=1}^{\infty} \lambda_\ell(\tau)\, \langle f,\, \varphi_\ell\rangle_\pi\, \varphi_\ell. \tag{3.5}$$

Furthermore, the eigenvalues decay exponentially with $\lambda_\ell(\tau) = \exp(-\kappa_\ell \tau)$ with relaxation rate $\kappa_\ell$ and relaxation timescale $t_\ell^{-1}$. Thus, for a sufficiently large lag time $\tau$, the fast relaxation processes have decayed and (3.5) can be approximated by finitely many terms. The propagator $\mathcal{P}_\tau$ has the same eigenvalues and the eigenfunctions $\tilde{\varphi}_\ell$ are given by $\tilde{\varphi}_\ell(x) = \pi(x)\, \varphi_\ell(x)$.

## 3.3 Data-driven approximation of transfer operators

In this section, we will describe different data-driven methods to identify the dominant dynamics of dynamical systems and to compute eigenfunctions of transfer operators associated with the system, namely TICA and DMD as well as VAC and EDMD. A formal derivation of methods to compute finite-dimensional approximations of transfer operators – resulting in the aforementioned methods – will be given in Section 3.4. Although TICA can be regarded as a special case of VAC, and DMD as a special case of EDMD, these methods are often used in different settings. With the aid of TICA, for instance, it is possible to identify the main slow coordinates and to project the dynamics onto the resulting reduced space, which can then be discretized using conventional Markov state models (a special case of VAC or EDMD, respectively, see Subsection 3.3.5). We will introduce the original methods – TICA and DMD – first and then extend these methods to the more general case. Since in many publications a different notation is used, we will first start with the required basic definitions.

In what follows, let $x_i, y_i \in \mathbb{R}^d$, $i = 1, \dots, m$, be a set of pairs of $d$-dimensional data vectors, where $x_i = \mathbf{X}_{t_i}$ and $y_i = \mathbf{X}_{t_i+\tau}$. Here, the underlying dynamical system is not necessarily known, the vectors $x_i$ and $y_i$ can simply be measurement data or data from a black-box simulation. In matrix form, this can be written as

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}, \tag{3.6}$$

with $X, Y \in \mathbb{R}^{d\times m}$. If one long trajectory $\{z_0, z_1, z_2, \dots\}$ of a dynamical system is given, i.e., $z_i = \mathbf{X}_{t_0+h\,i}$, where $h$ is the step size and $\tau = n_\tau\, h$ the lag time, we obtain

$$X = \begin{bmatrix} z_0 & z_1 & \cdots & z_{m-1} \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} z_{n_\tau} & z_{n_\tau+1} & \cdots & z_{n_\tau+m-1} \end{bmatrix}.$$

That is, in this case $Y$ is simply $X$ shifted by the lag time $\tau$. Naturally, if more than one trajectory is given, the data matrices $X$ and $Y$ can be concatenated.

In addition to the data, VAC and EDMD require a set of uniformly bounded basis functions or observables, given by $\{\psi_1, \psi_2, \dots, \psi_k\} \subset L^\infty(\mathbb{X})$. Since $\mathbb{X}$ is assumed to be bounded, we have $\psi_i \in L^r(\mathbb{X})$ for all $i = 1, \dots, k$ and $1 \leq r \leq \infty$. The basis functions could, for instance, be monomials, indicator functions, radial basis functions, or trigonometric functions. The optimal choice of basis functions remains an open problem and depends strongly on the system. If the set of basis functions is not sufficient to represent the eigenfunctions, the results will be inaccurate. A too large set of basis functions, on the other hand, might lead to ill-conditioned matrices and overfitting. Cross-validation strategies have been developed to detect overfitting [123].

For a basis $\psi_i$, $i = 1, \dots, k$, define $\psi : \mathbb{R}^d \to \mathbb{R}^k$ to be the vector-valued function given by

$$\psi(x) = [\psi_1(x), \psi_2(x), \dots, \psi_k(x)]^T. \tag{3.7}$$

The goal then is to find the best approximation of a given transfer operator in the space spanned by these basis functions. This will be explained in detail in Section 3.4. In addition to the data matrices $X$ and $Y$, we will need the transformed data matrices

$$\Psi_X = \begin{bmatrix} \psi(x_1) & \psi(x_2) & \dots & \psi(x_m) \end{bmatrix} \quad \text{and} \quad \Psi_Y = \begin{bmatrix} \psi(y_1) & \psi(y_2) & \dots & \psi(y_m) \end{bmatrix}. \tag{3.8}$$

### 3.3.1 Time-lagged independent component analysis

*Time-lagged independent component analysis* (TICA) has been introduced in [128] as a solution to the *blind source separation* problem, where the correlation matrix and the time-delayed correlation matrix are used to separate superimposed signals. The term TICA has been introduced later [85]. TDSEP [201], an extension of TICA, is popular in the machine learning community. It was shown only recently that TICA is a special case of the VAC by computing the optimal linear projection for approximating the slowest relaxation processes, and as such provides an approximation of the leading eigenvalues and eigenfunctions of transfer operators [150]. TICA is now a popular dimension reduction technique in the field of molecular dynamics [150, 172]. That is, TICA is used as a preprocessing step to reduce the size of the state space by projecting the dynamics onto the main coordinates. The time-lagged independent components are required **(a)** to be uncorrelated and **(b)** to maximize the autocovariances at lag time $\tau$, see [85, 150] for more details. Assuming that the system is reversible, the TICA coordinates are the eigenfunctions of $\mathcal{T}_\tau$ or $\mathcal{K}_\tau$, respectively, projected onto the space spanned by linear basis functions, i.e., $\psi(x) = x$.

Let $C(\tau)$ be the time-lagged covariance matrix defined by

$$C_{ij}(\tau) = \langle \mathbf{X}_{t,i}\, \mathbf{X}_{t+\tau,j} \rangle_t = \mathbb{E}_\pi \left[ \mathbf{X}_{t,i}\, \mathbf{X}_{t+\tau,j} \right].$$

Given data $X$ and $Y$ as defined above, estimators $C_0$ and $C_\tau$ for the covariance matrices $C(0)$ and $C(\tau)$ can be computed as

$$\begin{aligned} C_0 &= \tfrac{1}{m-1} \sum_{k=1}^{m} x_k \, x_k^T = \tfrac{1}{m-1} X X^T, \\ C_\tau &= \tfrac{1}{m-1} \sum_{k=1}^{m} x_k \, y_k^T = \tfrac{1}{m-1} X Y^T. \end{aligned} \tag{3.9}$$

The time-lagged independent components are defined to be solutions of the eigenvalue problem

$$C_\tau \, \xi_\ell = \lambda_\ell \, C_0 \, \xi_\ell \quad \text{or} \quad C_0^+ C_\tau \, \xi_\ell = \lambda_\ell \, \xi_\ell, \tag{3.10}$$

respectively. In what follows, let $M_{\mathrm{TICA}} = C_0^+ C_\tau$, where $C_0^+$ denotes the Moore–Penrose pseudo-inverse of $C_0$.

In applications, often the symmetrized estimators

$$C_0 = \tfrac{1}{2m-2}(X X^T + Y Y^T) \quad \text{and} \quad C_\tau = \tfrac{1}{2m-2}(X Y^T + Y X^T)$$

are used so that the resulting TICA coordinates become real-valued. This corresponds to averaging over the trajectory and the time-reversed trajectory. Note that this symmetrization can introduce a large estimator bias that affects the dominant spectrum of (3.10), if the process is non-stationary, or the distribution of the data is far from the equilibrium of the process. In the latter case, a reweighting procedure can be applied to obtain weighted versions of the estimators (3.9), to reduce that bias [198].

**Example 3.3.1.** Let us illustrate the idea behind TICA with a simple example. Consider the data shown in Figure 3.3, which was generated by a stochastic process which will typically spend a long time in one of the two clusters before it jumps to the other. We are interested in finding these metastable sets. Standard *principal component analysis* (PCA) leads to the coordinate shown in red, whereas TICA – shown in black – takes time-information into account and is thus able to identify the slow direction of the system correctly. Projecting the system onto the $x$-coordinate will preserve the slow process while eliminating the fast stochastic noise. △

**Algorithm 3.1** AMUSE algorithm to compute TICA.

1. Compute a reduced SVD of $X$, i.e., $X = U \, \Sigma \, V^T$.
2. Whiten data: $\tilde{X} = \Sigma^{-1} U^T X$ and $\tilde{Y} = \Sigma^{-1} U^T Y$.
3. Compute $\overline{M}_{\mathrm{TICA}} = \tilde{X} \tilde{Y}^T = \Sigma^{-1} U^T X Y^T U \Sigma^{-1}$.
4. Solve the eigenvalue problem $\overline{M}_{\mathrm{TICA}} \, w_\ell = \lambda_\ell \, w_\ell$.
5. The TICA coordinates are then given by $\xi_\ell = U \Sigma^{-1} w_\ell$.

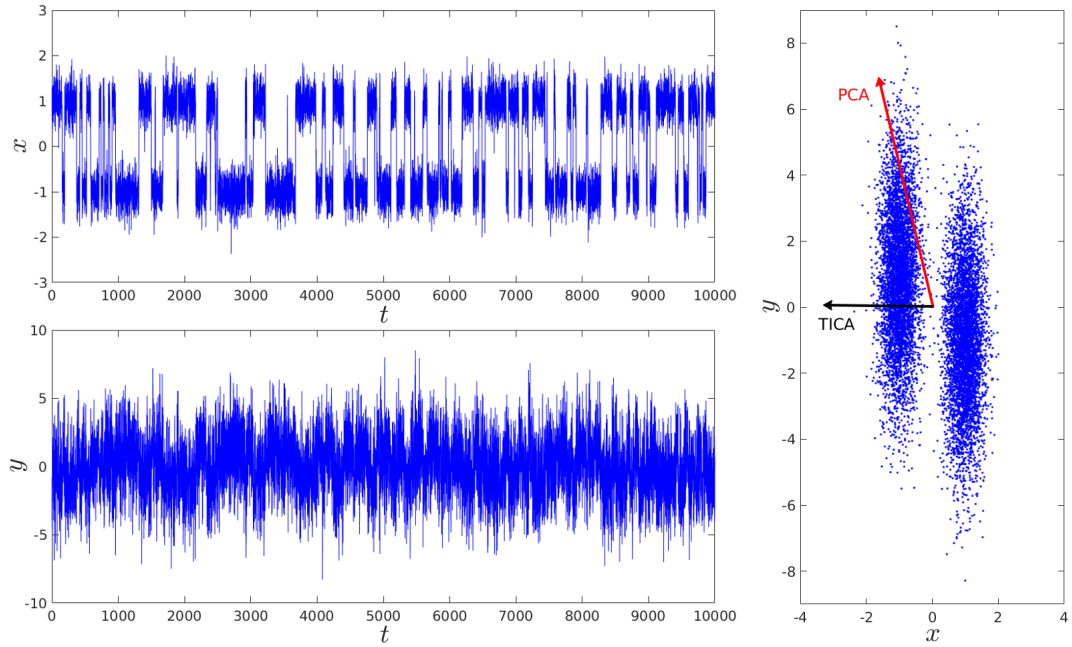



**Figure 3.3:** The difference between PCA and TICA. The top and bottom plot show the $x$- and $y$-component of the system, respectively, the plot on the right the resulting main principal component vector and the main TICA coordinate.

The TICA coordinates can be computed using AMUSE[5] [182] as shown in Algorithm 3.1. Instead of computing a singular value decomposition of the data matrix $X$ in step 1, an eigenvalue decomposition of the covariance matrix $XX^T$ could be computed, which is more efficient if $m \gg d$, but less accurate. The vectors $\xi_\ell$ computed by AMUSE are solutions of the eigenvalue problem (3.10), since

$$\begin{aligned} M_{\mathrm{TICA}}\, \xi_\ell &= \left(XX^T\right)^+ XY^T U\Sigma^{-1} w_\ell \\ &= U\Sigma^{-1}\tilde{X}\tilde{Y}^T w_\ell \\ &= \lambda_\ell\, U\Sigma^{-1} w_\ell \\ &= \lambda_\ell\, \xi_\ell. \end{aligned}$$

In the second line, we used the fact that $(XX^T)^+ = U\Sigma^{-2}U^T$ and in the third that $w_\ell$ is an eigenvector of $\overline{M}_{\mathrm{TICA}} = \tilde{X}\tilde{Y}^T$.

### 3.3.2 Dynamic mode decomposition

*Dynamic mode decomposition* (DMD) was developed by the fluid dynamics community as a tool to identify coherent structures in fluid flows [164]. Since its introduction, several

[5] Algorithm for Multiple Unknown Signals Extraction.

variants and extensions have been proposed, see [28, 186, 87, 21, 91]. A review of the applications of Koopman operator theory in fluid mechanics can be found in [127]. DMD can be viewed as a combination of a PCA in the spatial domain and a Fourier analysis in the frequency domain [19]. It can be shown that the DMD modes are the Koopman modes for the set of basis functions defined by $\psi(x) = x$. Given again data $X$ and $Y$ as above, the idea behind DMD is to assume that there exists a linear operator $M_{\mathrm{DMD}}$ such that $y_i = M_{\mathrm{DMD}}\, x_i$. Since the underlying dynamical system is in general nonlinear, this equation cannot be fulfilled exactly and we want to compute the matrix $M_{\mathrm{DMD}}$ in such a way that the Frobenius norm of the deviation is minimized, i.e.,

$$\min \|Y - M_{\mathrm{DMD}} X\|_F \,. \tag{3.11}$$

The solution of this minimization problem is given by

$$M_{\mathrm{DMD}} = YX^+ = \left(YX^T\right)\left(XX^T\right)^+ = C_\tau^T C_0^+ = M_{\mathrm{TICA}}^T. \tag{3.12}$$

The eigenvalues and eigenvectors of $M_{\mathrm{DMD}}$ are called DMD eigenvalues and modes, respectively. That is, we are solving

$$M_{\mathrm{DMD}}\, \xi_\ell = \lambda_\ell\, \xi_\ell.$$

The above equations already illustrate the close relationship with TICA, cf. (3.10). The DMD modes are the right eigenvectors of $M_{\mathrm{DMD}}$, whereas the TICA coordinates are defined to be the right eigenvectors of the transposed matrix $M_{\mathrm{TICA}}$. Hence, the TICA coordinates are the left eigenvectors of the DMD matrix and the DMD modes the left eigenvectors of the TICA matrix. This is consistent with the results that will be presented in the VAC and EDMD subsections below: The TICA coordinates represent the *Koopman eigenfunctions* projected onto the space spanned by linear basis functions, i.e., $\psi(x) = x$, while the DMD modes are the corresponding *Koopman modes*.

Similar to AMUSE, the DMD modes and eigenvalues can be obtained without explicitly computing $M_{\mathrm{DMD}}$ by using a reduced singular value decomposition of $X$. Standard and exact DMD are presented in Algorithm 3.2. The standard DMD modes are simply the exact DMD modes projected onto the range of the matrix $X$, see [186].

**Remark 3.3.2.** TICA and standard DMD are closely related. When comparing with the AMUSE formulation, we obtain

$$\overline{M}_{\mathrm{TICA}} = \tilde{X}\tilde{Y}^T = \Sigma^{-1}U^T XY^T U\Sigma^{-1} = \Sigma\, U^T M_{\mathrm{TICA}}\, U\Sigma^{-1} =: W\Lambda W^{-1}$$

and

$$\overline{M}_{\mathrm{DMD}} = U^T YV\Sigma^{-1} = U^T M_{\mathrm{DMD}}\, U = U^T M_{\mathrm{TICA}}^T U =: \tilde{W}\tilde{\Lambda}\tilde{W}^{-1}.$$

The TICA coordinates are given by $\Xi = U\Sigma^{-1}W$ and the standard DMD modes by $\tilde{\Xi} = U\tilde{W}$ so that – except for the scaling $\Sigma^{-1}$ – AMUSE and standard DMD use the same projection, the main difference is that the former computes the eigenvectors of $M_{\mathrm{TICA}}$ and the latter the eigenvectors of the transposed matrix $M_{\mathrm{TICA}}^T$. As a result, AMUSE could be rewritten to compute the DMD modes if we define $\overline{M}'_{\mathrm{DMD}} = \tilde{Y}\tilde{X}^T = \Sigma^{-1}U^T YX^T U\Sigma^{-1}$ in step 3 of the algorithm and $\xi_\ell = U\Sigma\, w_\ell$ in step 5, where $w_\ell$ now denotes the eigenvectors of $\overline{M}'_{\mathrm{DMD}}$.

**Algorithm 3.2** Standard and exact DMD.

1. Compute compact SVD of $X$, given by $X = U\,\Sigma\,V^T$.
2. Define $\overline{M}_{\text{DMD}} = U^T Y V \Sigma^{-1}$.
3. Compute eigenvalues and eigenvectors of $\overline{M}_{\text{DMD}}$, i.e., $\overline{M}_{\text{DMD}}\, w_\ell = \lambda_\ell\, w_\ell$.
4. The DMD mode corresponding to the eigenvalue $\lambda_\ell$ is defined as

$$\text{a) } \xi_\ell = U w_\ell. \qquad \text{(Standard DMD)}$$
$$\text{b) } \xi_\ell = \frac{1}{\lambda} Y V \Sigma^{-1} w_\ell. \qquad \text{(Exact DMD)}$$

### 3.3.3 Variational approach of conformation dynamics

The *variational approach of conformation dynamics* (VAC) [134, 137, 138] is a generalization of the frequently used Markov state modeling framework that allows arbitrary basis functions and is similar to the variational approach in quantum mechanics [137]. As described above, VAC and EDMD (see below) require – in addition to the data – a set of basis functions (also called *dictionary*), given by $\psi$. The variational approach is defined only for reversible systems – EDMD does not require this restriction – and computes eigenfunctions of $\mathcal{T}_\tau$ or $\mathcal{K}_\tau$, respectively. Using the data matrices $\Psi_X$ and $\Psi_Y$ defined in (3.8), $C_0$ and $C_\tau$ defined in (3.9) for the transformed data can be estimated as

$$C_0 = \tfrac{1}{m-1} \sum_{k=1}^{m} \psi(x_k)\, \psi(x_k)^T = \tfrac{1}{m-1} \Psi_X \Psi_X^T,$$
$$C_\tau = \tfrac{1}{m-1} \sum_{k=1}^{m} \psi(x_k)\, \psi(y_k)^T = \tfrac{1}{m-1} \Psi_X \Psi_Y^T.$$

In what follows, let $M_{\text{VAC}} = C_0^+ C_\tau$ for the transformed data matrices $\Psi_X$ and $\Psi_Y$. The matrix $M_{\text{VAC}}$ can be regarded as a finite-dimensional approximation of $\mathcal{K}_\tau$ (or $\mathcal{T}_\tau$, since the system is assumed to be reversible; the derivation is shown in Section 3.4), respectively. Eigenfunctions of the operator can then be approximated by the eigenvectors of the matrix $M_{\text{VAC}}$. Let $\xi_\ell$ be an eigenvector of $M_{\text{VAC}}$, i.e.,

$$M_{\text{VAC}}\, \xi_\ell = \lambda_\ell\, \xi_\ell,$$

and $\varphi_\ell(x) = \xi_\ell^* \psi(x)$, where $^*$ denotes the conjugate transpose. Since

$$\mathcal{K}_\tau \varphi_\ell(x) \approx (M_{\text{VAC}}\, \xi_\ell)^*\, \psi(x) = \lambda_\ell\, \xi_\ell^*\, \psi(x) = \lambda_\ell\, \varphi_\ell(x),$$

we obtain an approximation of the eigenfunctions of $\mathcal{K}_\tau$. The derivation will be described in detail in Section 3.4.

### 3.3.4 Extended dynamic mode decomposition

*Extended dynamic mode decomposition* (EDMD), a generalization of DMD, can be used to compute finite-dimensional approximations of the Koopman operator, its eigenvalues, eigenfunctions, and eigenmodes [195, 196]. It was shown in [92] that EDMD can be extended to approximate also eigenfunction of the Perron–Frobenius operator with respect to the density underlying the data points. With the notation introduced above, the minimization problem (3.11) for the transformed data matrices $\Psi_X$ and $\Psi_Y$ can be written as

$$\min \|\Psi_Y - M_{\mathrm{EDMD}} \Psi_X\|_F \,. \tag{3.13}$$

The solution – see also (3.12) – is given by

$$M_{\mathrm{EDMD}} = \Psi_Y \Psi_X^+ = \big(\Psi_Y \Psi_X^T\big)\big(\Psi_X \Psi_X^T\big)^+ = C_\tau^T \, C_0^+ = M_{\mathrm{VAC}}^T.$$

That is, instead of assuming a linear relationship between the data matrices $X$ and $Y$, EDMD aims at finding a linear relationship between the transformed data matrices $\Psi_X$ and $\Psi_Y$. Eigenfunctions of the Koopman operator are then given by the left eigenvectors $\xi_\ell$ of $M_{\mathrm{EDMD}}$, i.e.,

$$\varphi_\ell(x) = \xi_\ell^* \, \psi(x).$$

The derivation of EDMD can be found in Section 3.4. Since the left eigenvectors of $M_{\mathrm{EDMD}}$ are the right eigenvectors of $M_{\mathrm{VAC}}$, VAC and EDMD are equivalent as they compute exactly the same eigenvalue and eigenfunction approximations for a data and basis set.

As shown in [92], EDMD can also be used to approximate the Perron–Frobenius operator as follows:

$$\tilde{M}_{\mathrm{EDMD}} = \big(\Psi_X \Psi_Y^T\big)\big(\Psi_X \Psi_X^T\big)^+ = C_\tau \, C_0^+.$$

It is important to note that the Perron–Frobenius operator is computed with respect to the density underlying the data matrices. That is, if $X$ is sampled from a uniform distribution, we obtain the eigenfunctions of the Perron–Frobenius operator $\mathcal{P}_\tau$. If we, on the other hand, use one long trajectory, the underlying density converges to the equilibrium density $\pi$ and we obtain the eigenfunctions of the Perron–Frobenius operator with respect to the equilibrium density, denoted by $\mathcal{T}_\tau$. An approach to compute the equilibrium density from off-equilibrium data is proposed in [198].

**Example 3.3.3.** Let us consider the Ornstein–Uhlenbeck process introduced in Example 3.2.1. Here, $\alpha = 4$ and $D = 0.25$. The lag time is defined to be $\tau = 1$. We generated $10^5$ uniformly distributed test points in $[-2, 2]$ and used a basis comprising monomials of order up to 10. With the aid of EDMD, we computed the dominant eigenfunctions of the Perron–Frobenius operator $\mathcal{P}_\tau$ and the Koopman operator $\mathcal{K}_\tau$ (which is identical to $\mathcal{T}_\tau$ here due to reversibility). The results are shown in Figure 3.2. The corresponding eigenvalues are given by

$$\lambda_1(\tau) = 1.00, \quad \lambda_2(\tau) = 0.37, \quad \lambda_3(\tau) = 0.13, \quad \lambda_4(\tau) = 0.049,$$

which is a good approximation of the analytically computed eigenvalues (Example 3.2.6). △

In order to approximate the Koopman modes, let $\varphi(x) = [\varphi_1(x), \ldots, \varphi_k(x)]^T$ be the vector of eigenfunctions and

$$\Xi = \begin{bmatrix} \xi_1 & \xi_2 & \ldots & \xi_k \end{bmatrix}$$

the matrix that contains all left eigenvectors of $M_{\text{EDMD}}$. Furthermore, define $B \in \mathbb{R}^{d \times k}$ such that $g(x) = B\,\psi(x)$. That is, the full-state observable is written in terms of the basis functions[6]. Since $\varphi(x) = \Xi^*\,\psi(x)$, this leads to $g(x) = B\,\psi(x) = B\,(\Xi^*)^{-1}\varphi(x)$. Thus, the $\ell$th column vector of the matrix $\boldsymbol{\eta} = B\,(\Xi^*)^{-1}$ represents the Koopman mode $\eta_\ell$ required for the reconstruction of the dynamical system, see (3.3).

### 3.3.5 Relationships with other methods

For particular choices of basis functions, VAC and EDMD are equivalent to other methods (see also [138, 92]):

1. If we choose $\psi(x) = x$, we obtain TICA and DMD, respectively. That is, the TICA coordinates are the eigenfunctions of the Koopman operator projected onto the space spanned by linear basis functions and the DMD modes are the corresponding Koopman modes. (Note that in this case $B = I$ and the matrix $\boldsymbol{\eta} = (\Xi^*)^{-1}$ contains the right eigenvectors of $M_{\text{EDMD}}$.) In many applications of TICA, the basis functions are modified to have zero mean. For reversible processes, this eliminates the stationary eigenvalue $\lambda_1 = 1$ and its eigenfunction $\varphi_1 \equiv 1$. The largest eigenpair then approximates the slowest dynamical eigenvalue and eigenfunction, respectively.

2. If the set of basis functions comprises indicator functions $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ for a given decomposition of the state space into disjoint sets $A_1, \ldots, A_k$, VAC and EDMD result in Ulam's method [187] and thus a Markov state model (MSM).

These relationships are shown in Figure 3.4. Detailed examples illustrating the use of VAC and EDMD can be found in [137, 195, 92].

### 3.3.6 Examples

#### Double gyre

Let us consider the autonomous double gyre, which was introduced in [174], given by the SDE

$$\begin{aligned} d\mathbf{X}_t &= -\pi\, A\, \sin(\pi\, \mathbf{X}_t)\, \cos(\pi\, \mathbf{Y}_t) + \varepsilon\, d\mathbf{W}_{t,1}, \\ d\mathbf{Y}_t &= \quad \pi\, A\, \cos(\pi\, \mathbf{X}_t)\, \sin(\pi\, \mathbf{Y}_t) + \varepsilon\, d\mathbf{W}_{t,2} \end{aligned}$$

on the domain $\mathbb{X} = [0, 2] \times [0, 1]$ with reflecting boundary. For $\varepsilon = 0$, there is no transport between the left half and the right half of the domain and both subdomains are invariant sets with measure $\frac{1}{2}$, cf. [68, 69]. For $\varepsilon > 0$, there is a small amount of transport due

[6] The easiest way to accomplish this is by adding the observables $x_i$, $i = 1, \ldots, d$, to the set of basis functions.

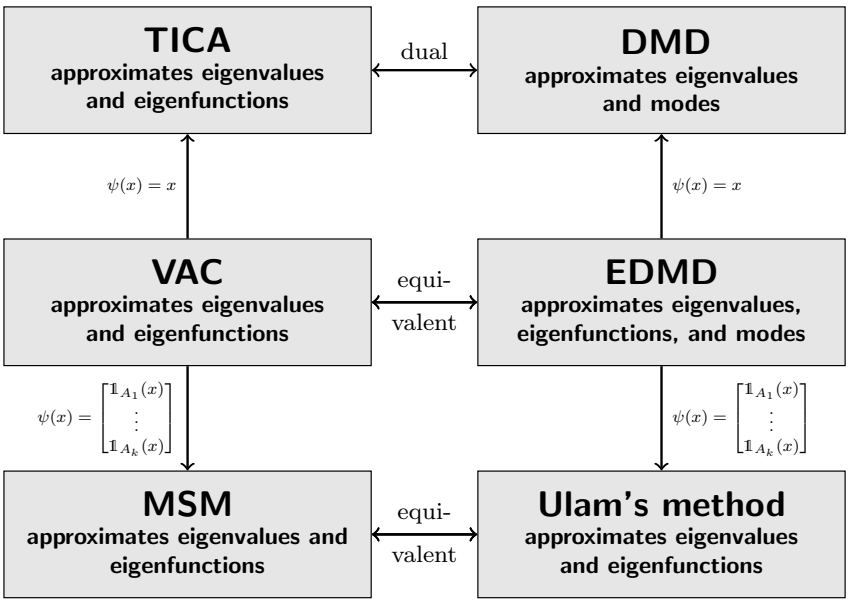



**Figure 3.4:** Relationships between data-driven methods. While VAC was derived for reversible dynamical systems, the derivation of EDMD covers non-reversible dynamics as well.

to diffusion and the subdomains are almost invariant. For the Koopman operator $\mathcal{K}_\tau$, this means that for $\varepsilon = 0$ the characteristic functions $\tilde{\varphi}_1 = \mathbb{1}_{[0,1]\times[0,1]}$ and $\tilde{\varphi}_2 = \mathbb{1}_{[1,2]\times[0,1]}$ are both eigenfunctions with corresponding eigenvalue 1. If, on the other hand, $\varepsilon > 0$, then the two-dimensional eigenspace subdivides into two one-dimensional eigenspaces with eigenvalues $\lambda_1 = 1$ and $\lambda_2 = 1 - \mathcal{O}(\varepsilon)$ and eigenfunctions $\varphi_1 = \mathbb{1}_{[0,2]\times[0,1]}$ and $\varphi_2 \approx \tilde{\varphi}_1 - \tilde{\varphi}_2$. Typical trajectories of the system are shown in Figure 3.5a. Using the parameters $A = 0.25$ and $\varepsilon = 0.05$, we integrated $10^5$ randomly generated test points using the Euler–Maruyama scheme with step size $h = 10^{-3}$.

For the computation of the eigenfunctions, we choose a set of radial basis functions whose centers were given by the midpoints of an equidistant $50 \times 25$ box discretization, and a lag time $\tau = 3$. The resulting nontrivial leading eigenfunctions of the Koopman operator computed with EDMD are shown in Figure 3.5b. The two almost invariant sets are clearly visible. The eigenfunctions of the Perron–Frobenius operator exhibit similar patterns (but "rotating" in the opposite direction).

### Deca-alanine

As a second example, we illustrate what has become a typical workflow for the application of VAC/EDMD in molecular dynamics, using deca-alanine as a model system. Deca-alanine is a small peptide comprised of ten alanine residues, it has been used as a test system many times before. Here, we analyze equilibrium simulations of $3\,\mu$s total

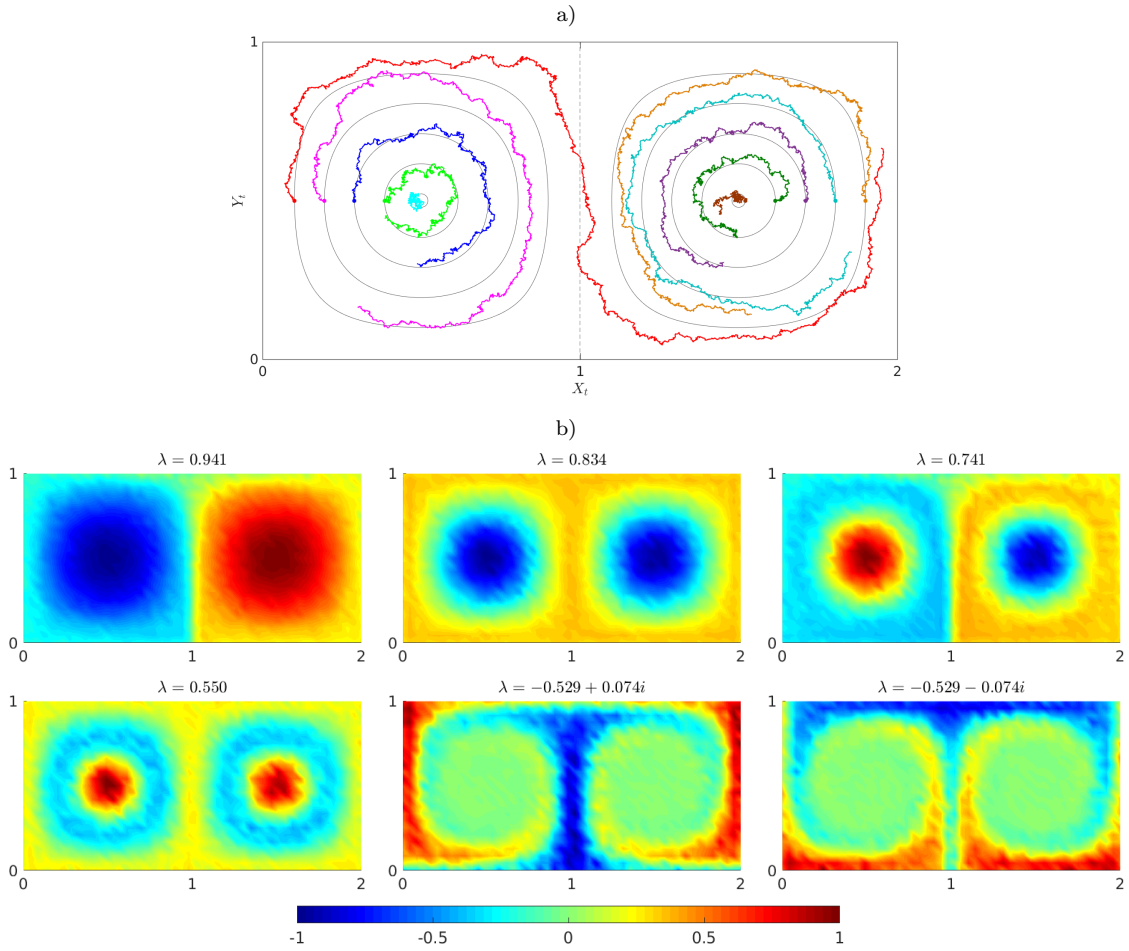



**Figure 3.5:** a) Typical trajectories (of different lengths) of the double gyre system for $\varepsilon = 0.05$. The initial states are marked by dots. Due to the diffusion term, particles can cross the separatrix (dashed line). The gray lines show the trajectories with the same initial conditions for $\varepsilon = 0$. b) Leading eigenfunctions of the Koopman operator associated with the double gyre system computed using EDMD.

simulation time using the Amber03 force field, see [137, 138] for the detailed simulation setup. A set of important quantities for our analysis are the leading *implied timescales*

$$t_m = -\frac{\tau}{\log|\lambda_m|}, \tag{3.14}$$

for $m = 2, 3, \ldots$. Implied timescales are independent of the lag time [148, Theorem 2.2.4]. However, if they are estimated using (3.14) and an approximation to the eigenvalues $\lambda_m$ obtained from VAC/EDMD, the timescales will be underestimated (see Section 3.4.2) and the error will decrease as a function of the lag time [47]. Approximate convergence of implied timescales with increasing lag time has become a standard model validation criterion in molecular dynamics [153].

In the first step, a set of internal molecular coordinates is extracted from the simulation data, to which TICA is applied. In our example, we select all 16 backbone dihedral angles as internal molecular coordinates. Figure 3.6a shows the first five implied timescales estimated by TICA as a function of the lag time $\tau$.

Next, a first dimension reduction is performed, where the data is projected onto the leading $M$ TICA eigenvectors. The number $M$ is selected by the criterion of total kinetic variance, that is, $M$ is the smallest number such that the cumulative sum of the first $M$ squared eigenvalues exceeds 95 per cent of the total sum of squared eigenvalues [133]. Figure 3.6c shows the resulting dimension $M$ as a function of the lag time.

As a third step, the reduced data set is discretized by application of a clustering method. In our case, we use $k$-means clustering to assign the data to 50 discrete states. A Markov state model (MSM, equivalent to Ulam's method, see above) is estimated from the discretized time series. We show the first five implied timescales from the MSM in Figure 3.6c and observe that estimates improve compared to the TICA approximations. Also, timescale estimates converge for lag times $\tau \geq 4\,\text{ns}$.

Finally, we use the converged model at lag time $\tau = 4\,\text{ns}$ for further analysis. As the slowest implied timescale $t_2$ dominates all others, and as it is the only one which is larger than the lag time used for analysis (indicated by the gray line in Figure 3.6c), we attempt to extract a two-state model that captures the essential dynamics. We employ the PCCA+ algorithm [42, 157] to coarse grain all MSM states into two macrostates. Inspection of randomly selected trajectory frames belonging to each macrostate reveals that the slow dynamical process in the data corresponds to the formation of a helix, see FigureR 3.6d. It should be noted that this coarse graining works well for visualization purposes, but some details need to be taken into account. In fact, PCCA performs a fuzzy assignment of MSM states to macrostates, where each MSM state belongs to each macrostate with a certain *membership* in $[0, 1]$. We simply assign each MSM state to the macrostate with maximal membership here. Alternatively, we could also use a *hidden Markov model* (HMM) to perform the coarse graining [136].

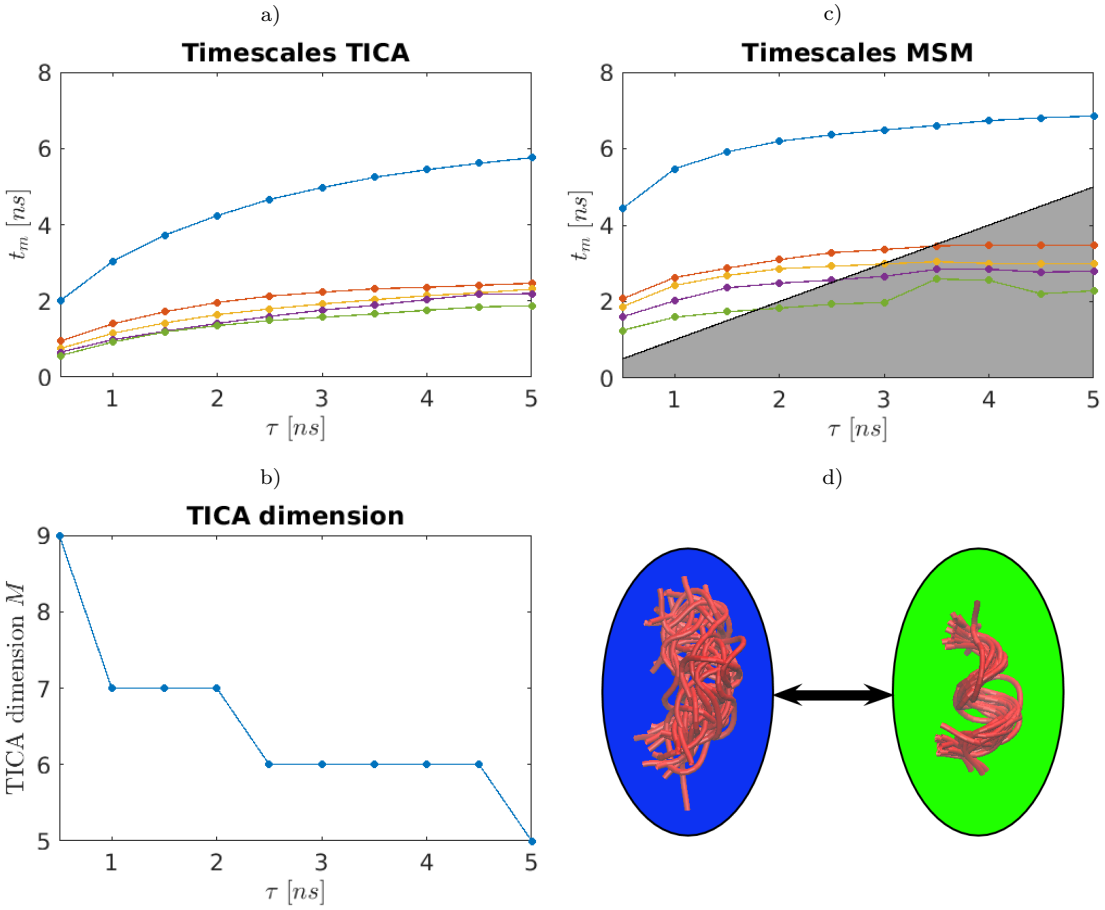



**Figure 3.6:** Illustration of standard EDMD workflow in molecular dynamics using the deca-alanine model system. a) Leading implied timescales $t_m$ (in nanoseconds) as estimated by TICA as a function of the lag time.b) Effective dimension $M$ selected by applying the criterion of total kinetic variance to the TICA eigenvalues. c) Leading implied timescales $t_m$ estimated by a Markov state model after projecting the data onto the first $M$ TICA eigenvectors and discretizing this data set into 50 states using $k$-means. d) Simple visualization of effective coarse grained dynamics. All MSM states are assigned to two macrostates using the PCCA algorithm. An overlay of representative structures from both macrostates shows that the dynamics between them corresponds to helix formation. Macrostates are drawn proportionally to their stationary probability.

## 3.4 Derivations

In this section, we will show how VAC and EDMD as well as their respective special cases TICA and DMD can be derived and how these methods are related to eigenfunctions and eigenmodes of transfer operators.

### 3.4.1 General dynamical systems

Let us begin with general, not necessarily reversible dynamical systems. In order to be able to compute eigenfunctions of transfer operators numerically, the infinite-dimensional operators are projected onto a finite-dimensional space. We will briefly outline how the EDMD minimization problem (3.13) leads to an approximation of the Koopman operator.

**Theorem 3.4.1.** *Let the process $\{\mathbf{X}_t\}_{t\geq 0}$ be Feller-continuous[7]. Let $\psi_i$, $i = 1, \ldots, k$, be the set of at least piecewise continuous basis functions of the finite-dimensional linear space $\mathbb{V}$. Let the empirical distribution of the data points $x_1, x_2, \ldots$ converge weakly to the density $\rho$. Then the minimization problem*

$$\min_{K\in\mathbb{R}^{k\times k}} \frac{1}{m}\sum_{j=1}^{m} \left\|\psi(y_j) - K^T\psi(x_j)\right\|_2^2$$

*converges, as $m \to \infty$, almost surely to $\min_{\hat{K}} \sum_{i=1}^{k} \|\mathcal{K}_\tau\psi_i - \hat{K}\psi_i\|_\rho^2$, where the minimization is over all linear mappings $\hat{K} : \mathbb{V} \to \mathbb{V}$.*

*Proof.* Let $f = \sum_{i=1}^{k} a_i\,\psi_i = a^T\psi \in \mathbb{V}$ be an arbitrary function, where $a = [a_1, \ldots, a_k]^T$. For a single data point $x_j$, we have for a linear mapping $\hat{K} : \mathbb{V} \to \mathbb{V}$ with matrix representation $K \in \mathbb{R}^{k\times k}$ that

$$\hat{K}f(x_j) = \sum_{i=1}^{k}(K\,a)_i\,\psi_i(x_j) = a^T K^T\psi(x_j).$$

Here, the $i$th column of the matrix $K$ corresponds to $\hat{K}\psi_i$. Thus, we obtain

$$\begin{aligned}\frac{1}{m}\sum_{j=1}^{m}\left\|\psi(y_j) - K^T\psi(x_j)\right\|_2^2 \quad &= \quad \sum_{i=1}^{k}\frac{1}{m}\sum_{j=1}^{m}(\psi_i(y_j) - \hat{K}\psi_i(x_j))^2 \\ &\xrightarrow{m\to\infty} \quad \sum_{i=1}^{k}\int_{\mathbb{X}}(\mathbb{E}[\psi_i(\mathbf{X}_\tau)\,|\,\mathbf{X}_0 = x] - \hat{K}\psi_i(x))^2\rho(x)\,dx\end{aligned}$$

[7] A process $\{\mathbf{X}_t\}_{t\geq 0}$ is called Feller-continuous if the mapping $x \mapsto \mathbb{E}[g(\mathbf{X}_t)|\mathbf{X}_0 = x]$ is continuous for any fixed continuous function $g$. This implies, that the Koopman operator of a Feller-continuous process has a well defined restriction from $L^\infty(\mathbb{X})$ to the set of continuous functions. Any stochastic process generated by an Itô stochastic differential equation with Lipschitz-continuous coefficients is Feller-continuous [140, Lemma 8.1.4].

$$= \sum_{i=1}^{k} \|\mathcal{K}_\tau \psi_i - \hat{K}\psi_i\|_\rho^2,$$

where the convergence for $m \to \infty$ is almost sure. From the first line to the second we used that the $y_j$ are realizations of the random variables $\mathbf{X}_\tau$ given $\mathbf{X}_0 = x_j$, that $\mathbf{X}_\tau$ is a Feller-continuous process, that the $\psi_i$ are (piecewise) continuous functions, and that the sampling process of $x_j$ is independent of the noise process that decides over $\mathbf{X}_\tau$ given $\mathbf{X}_0 = x_j$. □

With the aid of the data matrices $\Psi_X$ and $\Psi_Y$ defined in (3.8), this minimization problem can be written as

$$\min \left\| \Psi_Y - K^T \Psi_X \right\|_F^2,$$

which is identical to (3.13), where now $K^T = M_{\text{EDMD}}$. Thus, the transposed EDMD matrix $M_{\text{EDMD}}$ is an approximation of the Koopman operator. A similar setup allows for the approximation of the Perron–Frobenius operator with respect to the data point density $\rho$. For details, we refer to [92, Appendix A]. Note, however, that although the Perron–Frobenius and Koopman operators are adjoint, the matrix representation of the discrete Perron–Frobenius operator will in general not just be the transposed of the matrix $K$, unless the ansatz functions $\psi_i$ are orthonormal with respect to $\langle \cdot, \cdot \rangle_\rho$.

If the dynamical system is deterministic, we can already interpret the minimization (3.13) for finite values of $m$. As shown, e.g., in [92, 98], the solution of (3.13) is a Petrov–Galerkin projection of the Koopman operator on the ansatz space $\mathbb{V}$.

### 3.4.2 Reversible dynamical systems

Let us now assume that the system is reversible. That is, it holds that $\pi(x)\, p_\tau(x, y) = \pi(y)\, p_\tau(y, x)$ for all $x$ and $y$.

#### Variational principle for the Rayleigh trace

We can also derive a variational formulation for the first $M$ eigenvalues of the Koopman operator $\mathcal{K}_\tau$ in the reversible setting. It is a standard result for self-adjoint operators on a Hilbert space with bounded eigenvalue spectrum, see, e.g., [2]:

**Proposition 3.4.2.** *Assume that $1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_M$ are the dominant eigenvalues of the Koopman operator $\mathcal{K}_\tau$ on $L^2_\pi$. Then*

$$\begin{aligned} \sum_{\ell=1}^{M} \lambda_\ell &= \sup \sum_{\ell=1}^{M} \langle \mathcal{K}_\tau v_\ell, v_\ell \rangle_\pi, \\ \langle v_\ell, v_{\ell'} \rangle_\pi &= \delta_{\ell\ell'} \end{aligned} \tag{3.15}$$

*The sum of the first $M$ eigenvalues maximizes the Rayleigh trace, which is the sum on the right-hand side of (3.15) over all selections of $M$ orthonormal functions $v_\ell$. The maximum is attained for the first $M$ eigenfunctions $\varphi_1, \ldots, \varphi_M$.*

*Proof.* The $M$-dimensional space $\mathbb{V}$ spanned by the functions $v_\ell$ must contain an element $u_M$ which is orthonormal to the first $M-1$ eigenfunctions $\varphi_\ell$, i.e., $\langle u_M, \varphi_\ell \rangle_\pi = 0$, $\ell = 1, \ldots, M-1$, and $\|u_M\|_\pi = 1$. By the standard Rayleigh principle for self-adjoint operators

$$\langle K_\tau u_M, u_M \rangle_\pi \leq \lambda_M.$$

Next, determine a normalized element $u_{M-1}$ of the orthogonal complement of $u_M$ in $\mathbb{V}$ with $\langle u_{M-1}, \varphi_\ell \rangle_\pi = 0$, $\ell = 1, \ldots, M-2$. Again, we can invoke the Rayleigh principle to find

$$\langle \mathcal{K}_\tau u_{M-1}, u_{M-1} \rangle_\pi \leq \lambda_{M-1}.$$

Repeating this argument another $M-2$ times provides an orthonormal basis $u_1, \ldots, u_M$ of the space $\mathbb{V}$ such that

$$\sum_{\ell=1}^{M} \langle \mathcal{K}_\tau u_\ell, u_\ell \rangle_\pi \leq \sum_{\ell=1}^{M} \lambda_\ell.$$

As the Rayleigh trace is independent of the choice of orthonormal basis for the subspace $\mathbb{V}$, and the space itself was arbitrary, this proves (3.15). Clearly, the maximum is attained for the first $M$ eigenfunctions. □

Proposition 3.4.2 motivates the variational approach developed in [134, 137] to maximize the Rayleigh trace restricted to some fixed space of ansatz functions:

**Proposition 3.4.3.** *Let $\mathbb{V}$ be a space of $k$ linearly independent ansatz functions $\psi_i$ given by a dictionary as above. The set of $M \leq k$ mutually orthonormal functions $f^\ell = \sum_{i=1}^{k} a_i^\ell \psi_i$ which maximize the Rayleigh trace of the Koopman operator restricted to $\mathbb{V}$ is given by the first $M$ eigenvectors of the generalized eigenvalue problem*

$$C_\tau \, a^\ell = \hat{\lambda}_\ell \, C_0 \, a^\ell, \tag{3.16}$$

*where $a^\ell = \left(a_i^\ell\right)_{i=1}^k$, and the matrices $C_\tau$, $C_0$ are given by*

$$(C_\tau)_{ij} = \langle \mathcal{K}_\tau \psi_i, \, \psi_j \rangle_\pi,$$
$$(C_0)_{ij} = \langle \psi_i, \, \psi_j \rangle_\pi.$$

*Proof.* First, note that for any functions $f = \sum_{i=1}^{k} a_i \, \psi_i$ and $g = \sum_{i=1}^{k} b_i \, \psi_i$, we have that

$$\langle \mathcal{K}_\tau f, g \rangle_\pi = a^T C_\tau \, b,$$
$$\langle f, g \rangle_\pi = a^T C_0 \, b.$$

Let us assume that the ansatz functions are mutually orthonormal, i.e., $C_0 = I$. Then, maximization of the Rayleigh trace is equivalent to finding $M$ vectors $a^\ell$, such that $\left(a^\ell\right)^T a^{\ell'} = \delta_{\ell\ell'}$ and

$$\sum_{\ell=1}^{M} \left(a^\ell\right)^T C_\tau \, a^\ell = \sum_{\ell=1}^{M} \langle C_\tau \, a^\ell, a^\ell \rangle$$

is maximal. By Proposition 3.4.2 applied to the operator $C_\tau$ on $\mathbb{R}^N$, the vectors $a^\ell$ are given by the first $M$ eigenvectors of $C_\tau$. In the general case, transform the basis functions into a set of mutually orthonormal functions $\tilde{\psi}_i$ via $\tilde{\psi}_i = \sum_{j=1}^{k} C_0^{-1/2}(j,i)\,\psi_j$. For the transformed basis, we need to compute the eigenvectors $\tilde{a}^\ell$ of

$$C_0^{-1/2} C_\tau C_0^{-1/2}\, \tilde{a}^\ell = \hat{\lambda}_\ell\, \tilde{a}^\ell.$$

This is equivalent to the generalized eigenvalue problem (3.16), the relation between the eigenvectors is given by

$$a^\ell = C_0^{-1/2}\, \tilde{a}^\ell. \qquad \square$$

## 3.5 Conclusion

In this review paper, we established connections between different data-driven model reduction and transfer operator approximation methods developed independently by the dynamical systems, fluid dynamics, machine learning, and molecular dynamics communities. Although the derivations of these methods differ, we have shown that the resulting algorithms share many similarities.

DMD, TICA and MSMs are popular methods to approximate the dynamics of high-dimensional systems. Due to their simple basis functions, they conduct relatively rough approximations, but when only a few spectral components are required, the approximation error can be controlled by choosing sufficiently large lag times $\tau$ [160]. The more general methods VAC and EDMD are better suited to obtain accurate approximations of eigenfunctions. However, to ensure such an accurate approximation, one would have to deploy multiple basis functions in all coordinates and their combinations, which is unfeasible for high-dimensional systems, and would also lead to overfitting when estimating the eigenfunctions of the Koopman operator from a finite data set [123].

A natural approach to mitigate these problems is to construct an iterative or "deep" approach in which the dynamical systems subspace in which a high resolution of basis functions is required is found by multiple successive analysis steps. A common approach is to first reduce the dimension by an inexpensive method such as TICA, in order to have a relatively low-dimensional space in which the eigenfunctions are approximated with a higher-resolution method. Another possibility is to exploit low-rank tensor approximations of transfer operators and their eigenfunctions. Tensor- and sparse-grid based reformulations of some of the methods described in this paper can be found in [138, 95, 91], and in [88], respectively. The efficiency of these tensor decomposition approaches depends strongly on the coupling structure; strong coupling between different variables typically leads to high ranks. Furthermore, some tensor formats also depend on the ordering of variables and a permutation of the variable's indices would lead to different tensor decompositions. Yet another approach might be to exploit sparsity-promoting methods using $L_1$-regularization techniques. Basis functions that are not required to represent the eigenfunctions of an operator can thus be eliminated and refined adaptively. Moreover,

dictionary-learning methods could be applied to *learn* a basis set and to adapt the dictionary to specific data [116]. Future work includes evaluating and combining different dimensionality reduction, tensor decomposition, and sparsification methods to mitigate the curse of dimensionality.

## Acknowledgements

This research has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114 *"Scaling Cascades in Complex Systems"*, Project A04 *"Efficient calculation of slow and stationary scales in molecular dynamics"* and Project B03 *"Multilevel coarse graining of multi-scale problems"*, and by the Einstein Foundation Berlin (Einstein Center ECMath). Frank Noé acknowledges European Commission for funding through ERC starting grant "pcCell". Furthermore, we would like to thank the reviewers for their helpful comments and suggestions.

# 4

# Towards tensor-based methods for the numerical approximation of the Perron–Frobenius and Koopman operator

Stefan Klus[1] and Christof Schütte[1,2]

[1] Department of Mathematics and Computer Science, Freie Universität Berlin, Germany
[2] Zuse Institute Berlin, Germany

*Journal of Computational Dynamics*, 3(2):139–161, 2016

DOI: 10.3934/jcd.2016007

**Abstract**

The global behavior of dynamical systems can be studied by analyzing the eigenvalues and corresponding eigenfunctions of linear operators associated with the system. Two important operators which are frequently used to gain insight into the system's behavior are the Perron–Frobenius operator and the Koopman operator. Due to the curse of dimensionality, computing the eigenfunctions of high-dimensional systems is in general infeasible. We will propose a tensor-based reformulation of two numerical methods for computing finite-dimensional approximations of the aforementioned infinite-dimensional operators, namely Ulam's method and Extended Dynamic Mode Decomposition (EDMD). The aim of the tensor formulation is to approximate the eigenfunctions by low-rank tensors, potentially resulting in a significant reduction of the time and memory required to solve the resulting eigenvalue problems, provided that such a low-rank tensor decomposition exists. Typically, not all variables of a high-dimensional dynamical system contribute equally to the system's behavior, often the dynamics can be decomposed into slow and fast processes, which is also reflected in the eigenfunctions. Thus, the weak coupling between different variables might be approximated by low-rank tensor cores. We will illustrate the efficiency of the tensor-based formulation of Ulam's method and EDMD using simple stochastic differential equations.

## 4.1 Introduction

The Perron–Frobenius operator and the Koopman operator enable the analysis of the global behavior of dynamical systems. Eigenfunctions of these operators can be used to extract the dominant dynamics, to detect almost invariant sets, or to decompose the

system into fast and slow processes [40, 38, 152, 22, 195, 64, 65]. Assume the state space of your system is $\mathbb{R}^d$ and you want to discretize each direction using $n$ grid points (or boxes if Ulam's method is used), then overall $n^d$ values need to be stored. Even for $d = 10$ and $n = 10$, more than 70 gigabyte of storage space would be required, whereas typical systems might have hundreds or thousands of dimensions and naturally also require a more fine-grained discretization. This so-called *curse of dimensionality* can be overcome by using tensor formats which compress the data and store only the information that is relevant for the reconstruction. In general, only an approximation of the original data can be retrieved. Approximating the objects under consideration by sums of low-rank tensor products has become a powerful approach for tackling high-dimensional problems [80] and in many physically significant problems near-linear complexity can be achieved since the separation rank depends only weakly on the dimension [8]. For high-dimensional systems exhibiting multiscale behavior, it might be possible to represent the weak coupling between different variables by low-rank tensor cores. The leading eigenfunctions of the Koopman operator, for instance, are typically almost constant for the fast variables of the system and depend mainly on the slowly changing variables (see [64]). Thus, using tensor-based algorithms could reduce the amount of time and memory required to compute and store eigenfunctions significantly. In this way, analyzing high-dimensional systems that could not be tackled using standard methods might become feasible.

Tensors, in our sense, are just multidimensional arrays as shown in Figure 4.1. Here and in what follows, standard vectors will be denoted by lower-case letters, e.g. $v$, matrices by upper-case letters, e.g. $A$, and tensors by the corresponding bold symbols, e.g. $\mathbf{x}$. It is important to note that tensors are typically not explicitly given – for example by observed data –, but only implicitly as solutions of systems of linear or nonlinear equations or eigenvalue problems [76]. Thus, numerical methods that operate directly on tensor approximations need to be developed since the full tensors cannot be stored or handled anymore in practice.

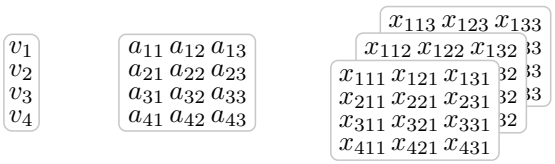



**Figure 4.1:** Tensors as multidimensional arrays. Here, $v = (v_i) \in \mathbb{R}^4$, $A = (a_{ij}) \in \mathbb{R}^{4\times 3}$, and $\mathbf{x} = (x_{ijk}) \in \mathbb{R}^{4\times 3\times 3}$.

Over the last years, low-rank tensor approximation approaches have become increasingly popular in the scientific computing community and are now becoming a standard tool to cope with large-scale problems that could not be handled before by standard numerical methods. An overview of different low-rank tensor approximation approaches can be found in [76].

In this paper, we will show that the use of low-rank tensor approximation schemes potentially enables the computation of eigenfunctions of high-dimensional systems. The aim

of this paper, however, is not to show that our approach is more efficient – the tensor algorithms are mainly implemented in Matlab, a comparison with highly optimized numerical libraries implemented in C or C++ would not lead to meaningful results, developing high performance libraries for large-scale tensor problems is a separate task –, but to derive a tensor-based reformulation of existing methods and to show equivalency so that the theory available for the conventional matrix-vector based formulation can be carried over to multi-dimensional arrays. Furthermore, tensors could enable low-rank approximations of the Perron–Frobenius and Koopman operator as well as their eigenfunctions. One of the main future goals is to combine low-rank tensor decomposition techniques and the splitting of the dynamics into fast and slow processes. In [64], it has been shown that such a splitting of a multi-scale system exists and can be exploited to extract the slow dynamics. Another open problem is the generation of the low-rank approximations of the operators. Currently, the canonical tensor format representations are converted to the tensor-train format, which is time-consuming and in general leads to high ranks. Ideally, low-rank TT approximations should be directly generated from the given data.

We will start by introducing standard methods such as Ulam's method and the recently developed Extended Dynamic Mode Decomposition (EDMD) to approximate the eigenfunctions of the Perron–Frobenius operator and the Koopman operator in Section 4.2. Then, in Section 4.3, different tensor formats will be described. In Section 4.4, we will reformulate Ulam's method and EDMD as tensor-based methods. Section 4.5 contains a brief summary of simple power iteration schemes for the resulting tensor-based (generalized) eigenvalue problems. Simple examples which illustrate the proposed approaches are shown in Section 4.6. In Section 4.7, we will conclude with a short summary and possible future work.

## 4.2 Perron–Frobenius and Koopman operator approximation

In this section, we will briefly introduce the Perron–Frobenius operator $\mathcal{P}$ and the Koopman operator $\mathcal{K}$ as well as numerical methods to compute finite-dimensional approximations, namely Ulam's method and EDMD. The main difference between Ulam's method and EDMD is that the former uses indicator functions[1] for a given box discretization of the domain while the latter allows arbitrary ansatz functions such as monomials, Hermite polynomials, trigonometric functions, or radial basis functions. Although EDMD was primarily developed for the approximation of the Koopman operator, it can be used to compute eigenfunctions of the Perron–Frobenius operator as well [92]. Analogously, Ulam's method can also be used to compute eigenfunctions of the Koopman operator.

[1] Higher-order methods for the approximation of the Perron–Frobenius operator have been proposed in [44].

### 4.2.1 Perron–Frobenius and Koopman operator

Let $S : \mathcal{X} \to \mathcal{X}$ be a dynamical system defined on a domain $\mathcal{X}$, for example $\mathcal{X} \subseteq \mathbb{R}^d$. Then the Perron–Frobenius operator or transfer operator $\mathcal{P}$ is defined by

$$\int g \cdot \mathcal{P} f \, dm = \int (g \circ S) \cdot f \, dm, \tag{4.1}$$

for all $f, g \in \mathcal{F}$, where $\mathcal{F}$ is an appropriately defined function space and $\circ$ denotes function composition. We assume in what follows that $\mathcal{F} = L^2(\mathcal{X})$. The aim is to compute eigenfunctions of the Perron–Frobenius operator, given by

$$\mathcal{P}\varphi_i = \lambda_i \varphi_i.$$

The eigenfunction $\varphi_1$ corresponding to $\lambda_1 = 1$ is the invariant density of the system, i.e. $\mathcal{P}\varphi_1 = \varphi_1$. The magnitude of the second largest eigenvalue $\lambda_2$ can be interpreted as the rate at which initial densities converge to the invariant density (for more details and assumptions about the dynamical system, see e.g. [64] and references therein). More generally, the leading eigenvalues of the Perron–Frobenius operator close to one correspond to the slowly converging transients of the system. The Koopman operator $\mathcal{K}$, on the other hand, is defined by

$$\mathcal{K} f = f \circ S$$

and acts on functions $f : \mathcal{X} \to \mathbb{C}$, $f \in \mathcal{F}$. Correspondingly, the stochastic Koopman operator is defined by $\mathcal{K} f = \mathbb{E}[\, f \circ S \,]$, where $\mathbb{E}[\,\cdot\,]$ denotes the expected value with respect to the probability measure underlying $S(x)$. We will only introduce the required notation and focus mainly on discrete-time dynamical systems, for more details on the Koopman operator and its properties, we refer to [22, 195, 196]. While the Perron–Frobenius operator describes the evolution of *densities*, the Koopman operator describes the evolution of *observables*, which could be measurements or sensor probes [22]. Instead of analyzing orbits $\{x, S(x), S^2(x), \dots\}$ of the dynamical system, we now analyze the measurements $\{f(x), f(S(x)), f(S^2(x)), \dots\}$ at these points.

The Koopman operator $\mathcal{K}$ is the adjoint of the Perron–Frobenius operator $\mathcal{P}$ and thus an infinite-dimensional but linear operator. A finite-dimensional approximation (computed using generalized Galerkin methods) of this operator captures the dynamics of a nonlinear dynamical system without necessitating a linearization around a fixed point [22, 195]. We are again interested in eigenfunctions of the operator, given by

$$\mathcal{K}\varphi_i = \lambda_i \varphi_i.$$

Let $f : \mathcal{X} \to \mathbb{R}$ be an observable of the system that can be written as a linear combination of the linearly independent eigenfunctions $\varphi_i$, i.e.

$$f(x) = \sum_i c_i \varphi_i(x),$$

with $c_i \in \mathbb{C}$. Then

$$(\mathcal{K}f)(x) = \sum_i \lambda_i c_i \varphi_i(x).$$

Analogously, for vector-valued functions $F = [f_1, \ldots, f_n]^T$, we obtain

$$\mathcal{K}F = \begin{bmatrix} \sum_i \lambda_i c_{i,1} \varphi_i \\ \vdots \\ \sum_i \lambda_i c_{i,n} \varphi_i \end{bmatrix} = \sum_i \lambda_i \varphi_i \begin{bmatrix} c_{i,1} \\ \vdots \\ c_{i,n} \end{bmatrix} = \sum_i \lambda_i \varphi_i v_i,$$

where $v_i = [c_{i,1}, \ldots, c_{i,n}]^T$. These vectors $v_i$ corresponding to the eigenfunctions $\varphi_i$ are called Koopman modes.

### 4.2.2 Ulam's method

A frequently used method to compute an approximation of the Perron–Frobenius operator is Ulam's method, see e.g. [38, 27, 10, 64]. First, the state space $\mathcal{X}$ is covered by a finite number of disjoint boxes $\{\mathcal{B}_1, \ldots, \mathcal{B}_k\}$. Let $\mathbb{1}_{\mathcal{B}_i}$ be the indicator function for box $\mathcal{B}_i$, i.e.

$$\mathbb{1}_{\mathcal{B}_i}(x) = \begin{cases} 1, & \text{if } x \in \mathcal{B}_i, \\ 0, & \text{otherwise.} \end{cases}$$

Then a finite-dimensional approximation of the operator can be obtained as follows: Using definition (4.1) leads to

$$\int \mathbb{1}_{\mathcal{B}_j} \cdot \mathcal{P}\mathbb{1}_{\mathcal{B}_i} \, dm = \int (\mathbb{1}_{\mathcal{B}_j} \circ S) \cdot \mathbb{1}_{\mathcal{B}_i} \, dm = \int \mathbb{1}_{S^{-1}(\mathcal{B}_j)} \cdot \mathbb{1}_{\mathcal{B}_i} \, dm = m(S^{-1}(\mathcal{B}_j) \cap \mathcal{B}_i).$$

This relationship can be represented by a matrix $\hat{P} = (\hat{p}_{ij}) \in \mathbb{R}^{k \times k}$ with

$$\hat{p}_{ij} = \frac{m\left(S^{-1}(\mathcal{B}_j) \cap \mathcal{B}_i\right)}{m(\mathcal{B}_i)}.$$

Here, in order to avoid confusion with the approximation $\mathbf{P}$ on a tensor space introduced below, we denote the matrix representation $\hat{P}$ instead of $P$. The denominator $m(\mathcal{B}_i)$ normalizes the entries $\hat{p}_{ij}$ so that $\hat{P}$ becomes a row-stochastic matrix and defines a finite Markov chain. The left eigenvector corresponding to the eigenvalue $\lambda_1 = 1$ approximates the invariant measure of the Perron–Frobenius operator $\mathcal{P}$.

The entries $\hat{p}_{ij}$ of the matrix $\hat{P}$ represent the probabilities of points being mapped from box $\mathcal{B}_i$ to box $\mathcal{B}_j$ by the dynamical system $S$. These entries can be estimated by randomly choosing a large number of test points $x_i^{(l)}$, $l = 1, \ldots, n$, in each box $\mathcal{B}_i$ and by counting how many times test points were mapped from box $\mathcal{B}_i$ to box $\mathcal{B}_j$ by $S$, i.e.

$$\hat{p}_{ij} \approx \frac{1}{n} \sum_{l=1}^{n} \mathbb{1}_{\mathcal{B}_j}\left(S\left(x_i^{(l)}\right)\right). \tag{4.2}$$

The eigenfunctions of the Perron–Frobenius operator are then approximated by the left eigenvectors of the matrix $\hat{P}$, the eigenfunctions of the Koopman operator by the right eigenvectors.

### 4.2.3 Extended dynamic mode decomposition

An approximation of the Koopman operator, the Koopman eigenvalues, eigenfunctions, and eigenmodes can be computed using EDMD. The method requires data, i.e. a set of values $x_i$ and the corresponding $y_i = S(x_i)$ values, $i = 1, \dots, m$, written in matrix form as

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix},$$

and additionally a set of ansatz functions or observables

$$\mathcal{D} = \{\psi_1, \psi_2, \dots, \psi_k\},$$

with $\psi_i : \mathcal{X} \to \mathbb{R}$. Thus, $X, Y \in \mathbb{R}^{d \times m}$. The vectors $x_i$ are used as collocation points to approximate the integrals required for the approximation of the Koopman operator. Let

$$\Psi = \begin{bmatrix} \psi_1 & \psi_2 & \cdots & \psi_k \end{bmatrix}^T, \tag{4.3}$$

$\Psi : \mathcal{X} \to \mathbb{R}^k$, be the vector of all ansatz functions, then $\mathcal{K}$ can be approximated by a matrix $\hat{K} \in \mathbb{R}^{k \times k}$, with

$$\hat{K}^T = \hat{A}\hat{G}^+,$$

where $^+$ denotes the pseudoinverse. The matrices $\hat{A}$, $\hat{G} \in \mathbb{R}^{k \times k}$ are defined as

$$\begin{aligned} \hat{A} &= \frac{1}{m} \sum_{l=1}^{m} \Psi(y_l)\Psi(x_l)^T, \\ \hat{G} &= \frac{1}{m} \sum_{l=1}^{m} \Psi(x_l)\Psi(x_l)^T. \end{aligned} \tag{4.4}$$

As before, we use the $\hat{}$ symbol to distinguish the matrices from the tensor approximations that will be introduced in Section 4.4. An approximation of the eigenfunction $\varphi_i$ of the Koopman operator $\mathcal{K}$ is then given by

$$\varphi_i = \xi_i \Psi,$$

where $\xi_i$ is the $i$-th left eigenvector of the matrix $\hat{K}^T$. Alternatively, the generalized eigenvalue problem

$$\xi_i \hat{A} = \lambda_i \xi_i \hat{G} \tag{4.5}$$

can be solved, provided that $\hat{G}$ is regular. To compute eigenfunctions of the Perron–Frobenius operator, the corresponding eigenvalue problem

$$\xi_i \hat{A}^T = \lambda_i \xi_i \hat{G}$$

needs to be solved. Note that this formulation is similar to the variational approach to compute eigenfunctions of transfer operators of reversible processes presented in [137, 138]. For more details, we refer the reader to [92].

## 4.3 Tensor formats

Several different tensor formats have been developed in the past, e.g. the canonical format, the Tucker format, and the tensor-train format. In this section, we will briefly introduce tensors and the required notation. The overall goal is to rewrite the methods presented in the previous section as tensor-based methods and to take advantage of low-rank tensor approximations and the fact that the dynamics of high-dimensional systems can often be decomposed.

### 4.3.1 Full format

A tensor in full format is simply a multidimensional array $\mathbf{v} \in \mathbb{R}^{k_1 \times \cdots \times k_d}$. (A variation of this format is the sparse format which stores only the nonzero entries and is used, for example, in the sparse grid approach [78].) The entries of a tensor $\mathbf{v}$ are indexed by $\mathbf{v}_\mathbf{i} = \mathbf{v}_{i_1,\ldots,i_d} = \mathbf{v}[\mathbf{i}] = \mathbf{v}[i_1, \ldots, i_d]$, where $\mathbf{i} = (i_1, \ldots, i_d)$ is a multi-index. Addition and subtraction are trivially defined element-wise. Multiplication of a tensor $\mathbf{v}$ by a scalar $c \in \mathbb{R}$ is naturally generalized as $(c\mathbf{v})[i_1, \ldots, i_d] = c\mathbf{v}[i_1, \ldots, i_d]$. Matrix-vector multiplication is defined as follows: Given a linear operator $\mathbf{A}$ defined on a tensor space $\mathbb{R}^{k_1 \times \cdots \times k_d}$, with

$$\mathbf{A} = \mathbf{A}[i_1, \ldots, i_d, j_1, \ldots, j_d] \in \mathbb{R}^{k_1 \times \cdots \times k_d \times k_1 \times \cdots \times k_d},$$

the product of $\mathbf{A}$ and $\mathbf{v}$ is

$$(\mathbf{A}\mathbf{v})[i_1, \ldots, i_d] = \sum_{j_1=1}^{k_1} \cdots \sum_{j_d=1}^{k_d} \mathbf{A}[i_1, \ldots, i_d, j_1, \ldots, j_d]\mathbf{v}[j_1, \ldots, j_d]$$

or in shorthand notation, using multi-indices,

$$(\mathbf{A}\mathbf{v})_\mathbf{i} = \sum_\mathbf{j} \mathbf{A}_\mathbf{ij}\mathbf{v}_\mathbf{j}.$$

Furthermore, the inner product of two tensors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^{k_1 \times \cdots \times k_d}$ is defined as

$$\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i_1=1}^{k_1} \cdots \sum_{i_d=1}^{k_d} \mathbf{v}[i_1, \ldots, i_d]\mathbf{w}[i_1, \ldots, i_d]$$

and the outer product $\mathbf{v} \otimes \mathbf{w} \in \mathbb{R}^{k_1 \times \cdots \times k_d \times k_1 \times \cdots \times k_d}$ as a tensor with entries

$$(\mathbf{v} \otimes \mathbf{w})[i_1, \ldots, i_d, j_1, \ldots, j_d] = \mathbf{v}[i_1, \ldots, i_d]\mathbf{w}[j_1, \ldots, j_d].$$

The outer product $\mathbf{v} \otimes \mathbf{w}$ can be regarded as a linear map that acts on tensors $\mathbb{R}^{k_1 \times \cdots \times k_d}$. Often it is required or convenient to rewrite a tensor as a vector. The *vectorization* of

a tensor, denoted $\mathrm{vec}(\mathbf{v})$, where $\mathrm{vec} : \mathbb{R}^{k_1 \times \cdots \times k_d} \to \mathbb{R}^{k_1 \cdots k_d}$, reorders the entries of $\mathbf{v}$ into one column vector. For $\mathbf{v} \in \mathbb{R}^{2\times 3\times 2}$, for example,

$$\mathrm{vec}(\mathbf{v}) = \begin{bmatrix} v_{111} & v_{211} \mid v_{121} & v_{221} \mid \ldots \mid v_{122} & v_{222} \mid v_{132} & v_{232} \end{bmatrix}^T.$$

For our purposes, we will mainly be interested in eigenvalue problems of the form

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v} \quad \text{or} \quad \mathbf{A}\mathbf{v} = \lambda \mathbf{B}\mathbf{v}.$$

The treatment of tensors in the full format often leads to storage problems. Thus, different formats have been developed to overcome this problem. Instead of working with the full format, we will use compressed formats such as the $r$-term or TT format for numerical computations in order to minimize computational costs as well as storage requirements. Except for very particular examples, it is impossible to compress the data without any compression error [78]. Typically, the tensor representation is just an approximation of the original data. Below, we will describe different compressed tensor formats, the introduction is based on [78].

### 4.3.2 Canonical format

A tensor space is given by $\mathbf{V} = \bigotimes_{\mu=1}^{d} V_\mu$, where $V_1, \ldots, V_d$ are vector spaces defined over the same field $\mathbb{K}$, typically $\mathbb{R}$ or $\mathbb{C}$. An *elementary tensor* is defined to be a product of the form

$$\mathbf{v} = v_1 \otimes \cdots \otimes v_d,$$

with $v_\mu \in V_\mu$, $\mu = 1, \ldots, d$. An *algebraic tensor* is then a linear combination of elementary tensors, i.e.

$$\mathbf{v} = \sum_{l=1}^{r} v_1^{(l)} \otimes \cdots \otimes v_d^{(l)}.$$

This format is also called $r$-term format or CP format. That is, instead of trying to store the tensor in the dense format, one only considers tensors that can be written as products of the form $\mathbf{v}[i_1, \ldots, i_d] = v_1[i_1] \cdots v_d[i_d]$. If the best approximation of this form is not good enough, then the natural extension is to consider $\mathbf{v}[i_1, \ldots, i_d] = \sum_{l=1}^{r} v_1^{(l)}[i_1] \cdots v_d^{(l)}[i_d]$, cf. [7, 8]. Defining

$$\mathcal{R}_r = \left\{ \sum_{l=1}^{r} v_1^{(l)} \otimes \cdots \otimes v_d^{(l)} : v_\mu^{(l)} \in V_\mu \right\}$$

for $r \in \mathbb{N}_0$, which implies that $\{0\} = \mathcal{R}_0 \subset \mathcal{R}_1 \subset \cdots \subset \mathbf{V}$, we call

$$\mathrm{rank}(\mathbf{v}) = \min\{r : \mathbf{v} \in \mathcal{R}_r\}$$

the *tensor rank* of $\mathbf{v}$.

**Example 4.3.1.** Let us consider a system of the form $dx_t = -\nabla_x V(x_t)\,dt + \sigma\,dW_t$, where $V$ is the energy landscape associated with the system. The invariant density of this process is given by $\mu(x) = \frac{1}{Z}e^{-\beta V(x)}$, where $\beta$ and $Z$ are constants [171]. Assume that the state space is three-dimensional and that the potential function can be written as $V(x) = V_1(x_1) + V_2(x_2) + V_3(x_3)$, then

$$\mu(x) = \frac{1}{Z}e^{-\beta V_1(x_1)}e^{-\beta V_2(x_2)}e^{-\beta V_3(x_3)}.$$

Thus, storing the invariant density for a grid with $k$ grid points in each direction in the full format would require an array of size $k^3$ while storing it in the canonical tensor format would require only a tensor of rank 1 and thus an array of size $3k$. △

Given tensors in the $r$-term format, basic operations are defined as follows [78]:

- **Addition:**

$$\mathbf{v} = \sum_{l=1}^{r_v}\bigotimes_{\mu=1}^{d} v_\mu^{(l)}, \quad \mathbf{w} = \sum_{l=1}^{r_w}\bigotimes_{\mu=1}^{d} w_\mu^{(l)} \quad \Rightarrow \quad \mathbf{x} = \mathbf{v} + \mathbf{w} = \sum_{l=1}^{r_v+r_w}\bigotimes_{\mu=1}^{d} x_\mu^{(l)},$$

  where

$$x_\mu^{(l)} = \begin{cases} v_\mu^{(l)}, & 1 \le l \le r_v, \\ w_\mu^{(l-r_v)}, & r_v + 1 \le l \le r_v + r_w. \end{cases}$$

- **Matrix-vector multiplication:**

$$\mathbf{A} = \sum_{l_A=1}^{r_A}\bigotimes_{\mu=1}^{d} A_\mu^{(l_A)}, \quad \mathbf{v} = \sum_{l_v=1}^{r_v}\bigotimes_{\mu=1}^{d} v_\mu^{(l_v)} \quad \Rightarrow \quad \mathbf{A}\mathbf{v} = \sum_{l_A=1}^{r_A}\sum_{l_v=1}^{r_v}\bigotimes_{\mu=1}^{d} A_\mu^{(l_A)} v_\mu^{(l_v)}.$$

Since these operations increase the rank, truncation is typically required to approximate the resulting tensor $\mathbf{v}$ with rank $r_v$ by a tensor $\tilde{\mathbf{v}}$ with a lower rank $r_{\tilde{v}}$ by either fixing the rank $r_{\tilde{v}}$ or by fixing $\varepsilon$ such that $\|\mathbf{v} - \tilde{\mathbf{v}}\| < \varepsilon$.

### 4.3.3 TT format

Another frequently used tensor format is the TT format – TT now stands for *tensor train* instead of the former *tree tensor* –, which can be obtained by successive singular value decompositions. This is a special case of a more general hierarchical format and has been introduced in quantum physics under the name *Matrix Product States* (MPS), see [78] for details. A tensor $\mathbf{v} \in \mathbb{R}^{k_1 \times \cdots \times k_d}$ is decomposed into $d$ component tensors $\mathbf{v}_i$ of at most order three (the first and last are of order two and are often, for the sake of simplicity, considered as tensors of order three with "boundary condition" $\rho_0 = \rho_d = 1$). That is, the entries of $\mathbf{v}$ are given by

$$\mathbf{v}[i_1, \ldots, i_d] = \sum_{k_1=1}^{\rho_1} \cdots \sum_{k_{d-1}=1}^{\rho_{d-1}} \mathbf{v}_1[1, i_1, k_1]\,\mathbf{v}_2[k_1, i_2, k_2]\,\ldots\,\mathbf{v}_d[k_{d-1}, i_d, 1].$$

For fixed indices, the component tensors of rank three can be regarded as matrices, which leads to a more compact representation

$$\mathbf{v}[i_1, \ldots, i_d] = V_1[i_1]\, V_2[i_2] \cdots V_d[i_d]$$

and justifies the original name MPS. Here, the numbers $\rho_i$ are the ranks of the TT tensor, resulting in a rank vector $\rho = [\rho_0, \rho_1, \ldots, \rho_d]$ which determines the complexity of the representation. The main advantages of the TT format are its stability from an algorithmic point of view and reasonable computational costs, provided that the ranks of the tensors are small [80]. The basic operations such as addition and matrix-vector multiplication are more complex than in the canonical format and can be found, for example, in [142]. Converting a tensor from the canonical format to the TT format is trivial, but the TT representation requires more memory. Numerical toolboxes for the TT decomposition of tensors and several algorithms for solving linear systems of equations are available online, see e.g. [143].

### 4.3.4 Comparison

Complexity-wise, the canonical format would be the ideal candidate for representing tensors since the number of required parameters depends only linearly on the dimension $d$, the rank $r$, and the sizes of the individual vector spaces. It turned out, however, that solving even simple problems using the canonical format is hard in practice due to redundancies and instabilities which can lead to numerical problems [80]. The main advantage of the TT format is its structural simplicity, higher-order tensors are reduced to $d$ products of tensors of at most order three. Similar approaches have been known in quantum physics for a long time, the rigorous mathematical analysis, however, is still work in progress (see [80] and references therein). For our purposes, we will rely on the TT format and the TT toolbox developed by Oseledets et al. [143] and implement simple power iteration schemes to solve the resulting eigenvalue problems as we will show in Section 4.5. Other tensor formats, however, might be advantageous as well for the analysis of the Perron–Frobenius and Koopman operator. This should be investigated further in the future. One drawback of the TT format is that the decomposition depends on the ordering of the dimensions and thus results in different tensor ranks for different orderings.

## 4.4 Tensor-based approximation

In this section, we will present a tensor-based reformulation of Ulam's method and EDMD and show that these methods are equivalent to the corresponding vector-based counterparts. The new formulation enables the use of the low-rank tensor approximation approaches described in the previous section. That is, variables can be approximated with different degrees of accuracy.

### 4.4.1 Reformulation of Ulam's method

Given a dynamical system $S : \mathcal{X} \to \mathcal{X}$, $\mathcal{X} \subset \mathbb{R}^d$, define a box $\mathcal{B}$ that contains $\mathcal{X}$, i.e.

$$\mathcal{B} = \mathcal{I}_1 \times \cdots \times \mathcal{I}_d \supset \mathcal{X},$$

where each $\mathcal{I}_\mu = [a_\mu, b_\mu] \subset \mathbb{R}$ is an interval. Furthermore, let each $\mathcal{I}_\mu$ be partitioned into $k_\mu$ subintervals $\mathcal{I}_\mu^{i_\mu}$, $i_\mu = 1, \ldots, k_\mu$, such that $\mathcal{I}_\mu^{i_\mu} \cap \mathcal{I}_\mu^{j_\mu} = \varnothing$ for $i_\mu \neq j_\mu$. This results in a partitioning of $\mathcal{B}$ into $\hat{k} = \prod_{\mu=1}^{d} k_\mu$ boxes. Using again standard multi-index notation, we will denote $\mathbf{i} = (i_1, \ldots, i_d)$. Equipped with the mapping

$$\mathbf{i} = (i_1, \ldots, i_d) \mapsto \hat{i} = 1 + \sum_{\mu=1}^{d} \left( \prod_{\nu=1}^{\mu-1} k_\nu \right) (i_\mu - 1), \tag{4.6}$$

each multi-index $\mathbf{i}$ corresponds to a number $\hat{i} \in \{1, \ldots, \hat{k}\}$. This induces a canonical numbering of the boxes

$$\mathcal{B}_{\mathbf{i}} = \mathcal{I}_1^{i_1} \times \cdots \times \mathcal{I}_d^{i_d} = \mathcal{B}_{\hat{i}}$$

and the entries of tensors $\mathbf{x} \in \mathbb{R}^{k_1 \times \cdots \times k_d}$ such that $\mathbf{x}_{\mathbf{i}} = x_{\hat{i}}$, where $x = \operatorname{vec}(\mathbf{x})$. Thus, with the aid of Ulam's method we could now generate the finite-dimensional representation of the Perron–Frobenius operator $\hat{P} \in \mathbb{R}^{\hat{k} \times \hat{k}}$ as described in Section 4.2. Our goal, however, is to approximate the operator by a tensor $\mathbf{P} \in \mathbb{R}^{k_1 \times \cdots \times k_d \times k_1 \times \cdots \times k_d}$. Note that the indicator function for the box $\mathcal{B}_{\mathbf{i}}$ can be written as

$$\mathbb{1}_{\mathcal{B}_{\mathbf{i}}}(x) = \prod_{\mu=1}^{d} \mathbb{1}_{\mathcal{I}_\mu^{i_\mu}}(x_\mu) = \mathbb{1}_{\mathcal{B}_{\hat{i}}}(x). \tag{4.7}$$

That is, each $d$-dimensional indicator function $\mathbb{1}_{\mathcal{B}_{\mathbf{i}}}(x)$ is now written as a product of $d$ one-dimensional indicator functions $\mathbb{1}_{\mathcal{I}_\mu^{i_\mu}}(x_\mu)$.

**Example 4.4.1.** Let us start with a simple example which illustrates the idea behind the tensor-based formulation. Consider the box discretization $\{\mathcal{B}_1, \ldots, \mathcal{B}_9\}$ of $\mathcal{B} = [0, 3]^2$ shown in Figure 4.2(a). Thus, using Ulam's method, we would obtain 9 indicator functions $\{\mathbb{1}_{\mathcal{B}_1}, \ldots, \mathbb{1}_{\mathcal{B}_9}\}$ and the matrix $\hat{P}$ that approximates the Perron–Frobenius operator $\mathcal{P}$ would be a row-stochastic $(9 \times 9)$-matrix. An example of such a matrix is shown in Figure 4.2(b), the underlying dynamical system is not relevant here. The goal now is to rewrite this matrix using tensors (cf. Example 4.4.3).

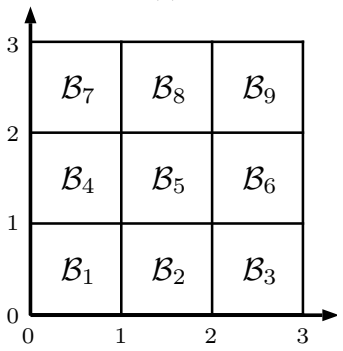
Defining intervals $\mathcal{I}_\mu^1 = [0, 1]$, $\mathcal{I}_\mu^2 = [1, 2]$, and $\mathcal{I}_\mu^3 = [2, 3]$ as well as indicator functions

$$\mathbb{1}_{\mathcal{I}_\mu^{i_\mu}}(x_\mu) = \begin{cases} 1, & x_\mu \in \mathcal{I}_\mu^{i_\mu}, \\ 0, & \text{otherwise}, \end{cases}$$

for $\mu = 1, 2$ and $i_\mu = 1, 2, 3$, the ansatz functions for the box discretization can be written as

$$\mathbb{1}_{\mathcal{B}_{1,1}}(x) = \mathbb{1}_{\mathcal{I}_1^1}(x_1) \mathbb{1}_{\mathcal{I}_2^1}(x_2) = \mathbb{1}_{\mathcal{B}_1}(x),$$

(a)




(b)

$$\hat{P} = \begin{bmatrix} 0.68 & 0.09 & 0.07 & 0.04 & 0.02 & 0 & 0.09 & 0.01 & 0 \\ 0.36 & 0.06 & 0.40 & 0.03 & 0 & 0.02 & 0.05 & 0.03 & 0.05 \\ 0.07 & 0.12 & 0.64 & 0.02 & 0 & 0.07 & 0.03 & 0 & 0.05 \\ 0.31 & 0.07 & 0.02 & 0.09 & 0.01 & 0.02 & 0.39 & 0.05 & 0.04 \\ 0.25 & 0.05 & 0.18 & 0.06 & 0.01 & 0.02 & 0.17 & 0.05 & 0.21 \\ 0.06 & 0.06 & 0.37 & 0.01 & 0 & 0.04 & 0.07 & 0.03 & 0.36 \\ 0.17 & 0.01 & 0.01 & 0.09 & 0.01 & 0.01 & 0.60 & 0.05 & 0.05 \\ 0.05 & 0 & 0.06 & 0.08 & 0 & 0.05 & 0.29 & 0.13 & 0.34 \\ 0.01 & 0 & 0.03 & 0.02 & 0.02 & 0.11 & 0.05 & 0.09 & 0.67 \end{bmatrix}$$

**Figure 4.2:** (a) Box discretization of $\mathcal{B} = [0,3]^2$. (b) Example of a resulting approximation $\hat{P}$ of the Perron–Frobenius operator $\mathcal{P}$, obtained by applying Ulam's method.

$$\begin{aligned} \mathbb{1}_{\mathcal{B}_{2,1}}(x) &= \mathbb{1}_{\mathcal{I}_1^2}(x_1)\mathbb{1}_{\mathcal{I}_2^1}(x_2) = \mathbb{1}_{\mathcal{B}_2}(x), \\ &\vdots \qquad\qquad\qquad \vdots \\ \mathbb{1}_{\mathcal{B}_{2,3}}(x) &= \mathbb{1}_{\mathcal{I}_1^2}(x_1)\mathbb{1}_{\mathcal{I}_2^3}(x_2) = \mathbb{1}_{\mathcal{B}_8}(x), \\ \mathbb{1}_{\mathcal{B}_{3,3}}(x) &= \mathbb{1}_{\mathcal{I}_1^3}(x_1)\mathbb{1}_{\mathcal{I}_2^3}(x_2) = \mathbb{1}_{\mathcal{B}_9}(x). \end{aligned} \qquad \triangle$$

The product formulation of the indicator functions naturally leads to a tensor approximation $\mathbf{P}$ of the Perron–Frobenius operator $\mathcal{P}$. Let $Q_\mu : \mathbb{R}^d \to \mathbb{R}$ be the projection onto the $\mu$-th component of a vector, i.e. $Q_\mu(x) = x_\mu$. Then we define

$$\mathbf{P}[i_1, \ldots, i_d, j_1, \ldots, j_d] = \frac{1}{n} \sum_{l=1}^{n} \prod_{\mu=1}^{d} \mathbb{1}_{\mathcal{I}_\mu^{j_\mu}} \left( Q_\mu \left( S \left( x_{i_1,\ldots,i_d}^{(l)} \right) \right) \right), \tag{4.8}$$

where $x_{i_1,\ldots,i_d}^{(l)}$, $l = 1, \ldots, n$, are the test points generated for box $\mathcal{B}_{\mathbf{i}}$. Instead of checking to which $d$-dimensional box the test points are mapped, the $d$ dimensions are now treated separately.

**Proposition 4.4.2.** *It holds that*

$$\mathbf{P}\mathbf{v} = \lambda \mathbf{v} \quad \Leftrightarrow \quad \hat{P}v = \lambda v.$$

*Proof.* It suffices to show that $\mathbf{P}[i_1, \ldots, i_d, j_1, \ldots, j_d] = \hat{P}_{\hat{i}\hat{j}}$ and that $(\mathbf{P}\mathbf{v})[i_1, \ldots, i_d] = (\hat{P}v)_{\hat{i}}$. The entries of $\mathbf{P}$ and $\hat{P}$ are identical since with (4.2) and (4.7)

$$\begin{aligned} \hat{P}_{\hat{i}\hat{j}} &= \frac{1}{n} \sum_{l=1}^{n} \mathbb{1}_{\mathcal{B}_{\hat{j}}} \left( S \left( x_{\hat{i}}^{(l)} \right) \right) \\ &= \frac{1}{n} \sum_{l=1}^{n} \prod_{\mu=1}^{d} \mathbb{1}_{\mathcal{I}_\mu^{j_\mu}} \left( Q_\mu \left( S \left( x_{i_d,\ldots,i_d}^{(l)} \right) \right) \right) = \mathbf{P}[i_1, \ldots, i_d, j_1, \ldots, j_d]. \end{aligned}$$

For $d = 1$, the multi-index $\mathbf{i}$ is mapped to $\hat{i} = i_1$ and $\mathbf{j}$ to $\hat{j} = j_1$ by (4.6). Furthermore, $\hat{k} = k_1$ and

$$(\mathbf{P}\mathbf{v})[i_1] = \sum_{j_1=1}^{k_1} \mathbf{P}[i_1, j_1]\mathbf{v}[j_1] = \sum_{\hat{j}=1}^{\hat{k}} \hat{P}_{\hat{i}\hat{j}} v_{\hat{j}} = (\hat{P}v)_{\hat{i}}.$$

Then, we obtain by induction

$$\begin{aligned}
&(\mathbf{P}\mathbf{v})[i_1, \ldots, i_{d+1}] \\
&\quad = \sum_{j_1=1}^{k_1} \cdots \sum_{j_{d+1}=1}^{k_{d+1}} \mathbf{P}[i_1, \ldots, i_{d+1}, j_1, \ldots, j_{d+1}]\mathbf{v}[j_1, \ldots, j_{d+1}] \\
&\quad = \sum_{j_{d+1}=1}^{k_{d+1}} \left( \sum_{j_1=1}^{k_1} \cdots \sum_{j_d=1}^{k_d} \mathbf{P}[i_1, \ldots, i_d, i_{d+1}, j_1, \ldots, j_d, j_{d+1}]\mathbf{v}[j_1, \ldots, j_d, j_{d+1}] \right) \\
&\quad = \sum_{j_{d+1}=1}^{k_{d+1}} \left( \sum_{j_1=1}^{k_1} \cdots \sum_{j_d=1}^{k_d} \mathbf{P}^{(i_{d+1}, j_{d+1})}[i_1, \ldots, i_d, j_1, \ldots, j_d]\mathbf{v}^{(j_{d+1})}[j_1, \ldots, j_d] \right) \\
&\quad = \begin{bmatrix} \hat{P}^{(i_{d+1},1)} & \ldots & \hat{P}^{(i_{d+1},k_{d+1})} \end{bmatrix} \begin{bmatrix} v^{(1)} \\ \vdots \\ v^{(k_{d+1})} \end{bmatrix} \begin{matrix} \} \in \mathbb{R}^{k_1 \cdots k_d} \\ \vdots \\ \} \in \mathbb{R}^{k_1 \cdots k_d} \end{matrix} \\
&\quad = (\hat{P}v)_{\hat{i}}.
\end{aligned}$$

Here, the matrices and vectors with the superscripts $(i_{d+1}, j_{d+1})$ and $(j_{d+1})$, respectively, are obtained by fixing the corresponding indices, that is, these matrices and vectors are lower-dimensional slices of the corresponding higher-dimensional objects. Since the entries of $v^{(j_{d+1})}$ are indexed by $(j_1, \ldots, j_d) \mapsto \hat{j} = 1 + \sum_{\mu=1}^{d} \left( \prod_{\nu=1}^{\mu-1} k_\nu \right) (j_\mu - 1)$, the entries of the larger vector $v$ – obtained by stacking the vectors $v^{(j_{d+1})} \in \mathbb{R}^{k_1 \cdots k_d}$ – are indexed by

$$\begin{aligned}
(j_1, \ldots, j_{d+1}) \mapsto \hat{j} &= 1 + \sum_{\mu=1}^{d} \left( \prod_{\nu=1}^{\mu-1} k_\nu \right) (j_\mu - 1) + (k_1 \cdots k_d)(j_{d+1} - 1) \\
&= 1 + \sum_{\mu=1}^{d+1} \left( \prod_{\nu=1}^{\mu-1} k_\nu \right) (j_\mu - 1).
\end{aligned}$$

□

Analogously, left eigenvectors of $\mathbf{P}$ correspond to left eigenvectors of $\hat{P}$. The implementation of the tensor-based formulation is straightforward since only index computations for intervals are required, a numbering of the $d$-dimensional boxes is not needed anymore. Let $\mathbb{T}$ be the set of all test points and $\mathrm{ind} : \mathbb{R}^d \to \mathbb{N}^d$ the functions that returns the corresponding multi-index $\mathbf{i}$ for a point $x \in \mathbb{R}^d$ so that $x_\mu \in I_\mu^{i_\mu}$, $\mu = 1, \ldots, d$. Then, Ulam's method can simply be expressed as:

**for each** test point $x \in \mathbb{T}$ **do**
$y = S(x)$
$\mathbf{i} = \mathrm{ind}(x)$
$\mathbf{j} = \mathrm{ind}(y)$
$\mathbf{P}[i_1, \ldots, i_d, j_1, \ldots, j_d] \leftarrow \mathbf{P}[i_1, \ldots, i_d, j_1, \ldots, j_d] + \frac{1}{n}$
**end for**

In the standard formulation, the rows of the matrix $\hat{P}$ sum up to one. Correspondingly, the sum of all entries of each subtensor of $\mathbf{P}$ with fixed multi-index $\mathbf{i}$ is one, i.e.

$$\sum_{j_1=1}^{k_1} \cdots \sum_{j_d=1}^{k_d} \mathbf{P}[i_1, \ldots, i_d, j_1, \ldots, j_d] = 1.$$

**Example 4.4.3.** Let us consider Example 4.4.1 again. We select $n$ random test points $x_{i_1,i_2}^{(l)}$ for each box $\mathcal{B}_{i_1,i_2}$, $i_1, i_2 = 1, \ldots, 3$ and $l = 1, \ldots, n$. This leads to a new approximation $\mathbf{P} \in \mathbb{R}^{3\times3\times3\times3}$. Written in Matlab notation, we would obtain

$$\mathbf{P}[:,:,1,1] = \begin{bmatrix} 0.68 & 0.31 & 0.17 \\ 0.36 & 0.25 & 0.05 \\ 0.07 & 0.06 & 0.01 \end{bmatrix}, \mathbf{P}[:,:,1,2] = \begin{bmatrix} 0.04 & 0.09 & 0.09 \\ 0.03 & 0.06 & 0.08 \\ 0.02 & 0.01 & 0.02 \end{bmatrix}, \mathbf{P}[:,:,1,3] = \begin{bmatrix} 0.09 & 0.39 & 0.60 \\ 0.05 & 0.17 & 0.29 \\ 0.03 & 0.07 & 0.05 \end{bmatrix},$$

$$\mathbf{P}[:,:,2,1] = \begin{bmatrix} 0.09 & 0.07 & 0.01 \\ 0.06 & 0.05 & 0 \\ 0.12 & 0.06 & 0 \end{bmatrix}, \mathbf{P}[:,:,2,2] = \begin{bmatrix} 0.02 & 0.01 & 0.01 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.02 \end{bmatrix}, \mathbf{P}[:,:,2,3] = \begin{bmatrix} 0.01 & 0.05 & 0.05 \\ 0.03 & 0.05 & 0.13 \\ 0 & 0.03 & 0.09 \end{bmatrix},$$

$$\mathbf{P}[:,:,3,1] = \begin{bmatrix} 0.07 & 0.02 & 0.01 \\ 0.40 & 0.18 & 0.06 \\ 0.64 & 0.37 & 0.03 \end{bmatrix}, \mathbf{P}[:,:,3,2] = \begin{bmatrix} 0 & 0.02 & 0.01 \\ 0.02 & 0.02 & 0.05 \\ 0.07 & 0.04 & 0.11 \end{bmatrix}, \mathbf{P}[:,:,3,3] = \begin{bmatrix} 0 & 0.04 & 0.05 \\ 0.05 & 0.21 & 0.34 \\ 0.05 & 0.36 & 0.67 \end{bmatrix}.$$

Note that each matrix $\mathbf{P}[:,:,j_1,j_2]$ corresponds to a column of matrix $\hat{P}$ in Figure 4.2. For the resulting eigenvalue problem, we obtain – using a simple power iteration, see Section 4.5 – the left eigenvector $\mathbf{v}_1$ corresponding to the largest eigenvalue $\lambda_1 = 1$

$$\mathbf{v}_1 = \begin{bmatrix} 0.6503 & 0.1393 & 0.4501 \\ 0.1046 & 0.0261 & 0.0901 \\ 0.4355 & 0.0864 & 0.3719 \end{bmatrix},$$

which is a good approximation of the largest left eigenvector $v_1$ of the matrix $\hat{P}$ given by

$$v_1 = \begin{bmatrix} 0.6503 & 0.1393 & 0.4501 & 0.1046 & 0.0261 & 0.0901 & 0.4355 & 0.0864 & 0.3719 \end{bmatrix}. \quad \triangle$$

Instead of working with the full format, the matrix $\mathbf{P}$ can also be expressed directly using the canonical tensor format. Assume that a test point $x$ is mapped from box $\mathcal{B}_\mathbf{i}$ to box $\mathcal{B}_\mathbf{j}$, where $\mathbf{i} = (i_1, \ldots, i_d)$ and $\mathbf{j} = (j_1, \ldots, j_d)$ are again multi-indices. Now let $e_\mu^{i_\mu} \in \mathbb{R}^{k_\mu}$ be the $i_\mu$-th unit vector of size $k_\mu$ and let

$$\mathbf{e}^\mathbf{i} = \bigotimes_{\mu=1}^{d} e_\mu^{i_\mu}.$$

Then the elementary tensor $\mathbf{e}^{\mathbf{i}} \otimes \mathbf{e}^{\mathbf{j}} \in \mathbb{R}^{k_1 \times \cdots \times k_d \times k_1 \times \cdots \times k_d}$ describes the mapping of this point. Furthermore, let ind be again the function that returns the multi-index of the box that contains the point $x$ and $\mathbb{T}$ the set of all test points, then the matrix $\mathbf{P}$ can be represented as a sum of elementary tensors of this form, i.e.

$$\mathbf{P} = \frac{1}{n} \sum_{x \in \mathbb{T}} \mathbf{e}^{\mathrm{ind}(x)} \otimes \mathbf{e}^{\mathrm{ind}(S(x))}.$$

That is, the number of elementary tensors is $\hat{k}n$, i.e. the number of boxes multiplied by the number of test points per box, and thus potentially too large to store (unless sparse tensor formats are used). However, we will store only a low-rank approximation to reduce the required storage space. Note that this elementary tensor representation can also be easily converted into the TT format for numerical computations using the TT toolbox.

The question now is whether the tensor representation offers advantages over the standard formulation of Ulam's method. The goal is to approximate the eigenfunctions of the Perron–Frobenius operator or Koopman operator using low-rank tensors, reducing the computational cost as well as the memory consumption. Before we present numerical results, let us also rewrite EDMD in tensor form.

### 4.4.2 Reformulation of EDMD

Instead of writing $\Psi$ as a vector of functions $\Psi = [\psi_1, \psi_2, \ldots, \psi_{\hat{k}}]^T$, we now write $\Psi$ as a tensor of functions. We start by selecting basis functions for each dimension separately. Let

$$\mathcal{D}_\mu = \{\psi_\mu^1, \ldots, \psi_\mu^{k_\mu}\}$$

be the set of basis functions for dimension $\mu$, $\mu = 1, \ldots, d$. Here, each $\psi_\mu^{i_\mu} : \mathbb{R} \to \mathbb{R}$ depends only on $x_\mu$. Then our tensor basis for EDMD contains all functions of the form

$$\boldsymbol{\psi}^{\mathbf{i}}(x) = \prod_{\mu=1}^{d} \psi_\mu^{i_\mu}(x_\mu), \tag{4.9}$$

where $\mathbf{i} = (i_1, \ldots, i_d)$ is a multi-index. Thus,

$$\mathcal{D} = \left\{ \prod_{\mu=1}^{d} \psi_\mu^{i_\mu}, \psi_\mu^{i_\mu} \in \mathcal{D}_\mu \right\}.$$

That is, we have again $\hat{k} = \prod_{\mu=1}^{d} k_\mu$ basis functions and $\boldsymbol{\Psi} : \mathbb{R}^d \to \mathbb{R}^{k_1 \times \cdots \times k_d}$, with

$$\boldsymbol{\Psi}[i_1, \ldots, i_d](x) = \boldsymbol{\psi}^{\mathbf{i}}(x).$$

**Example 4.4.4.** Let us begin with a simple example: Assume we have a two-dimensional domain $\mathcal{X} \subset \mathbb{R}^2$ and we want to use monomials of order up to three $\{1, x_\mu, x_\mu^2, x_\mu^3\}$ in $x_1$

and $x_2$ direction to approximate the eigenfunctions of the Koopman operator. Written in tensor form, we obtain

$$\boldsymbol{\Psi}(x) = \begin{bmatrix} 1 & x_2 & x_2^2 & x_2^3 \\ x_1 & x_1x_2 & x_1x_2^2 & x_1x_2^3 \\ x_1^2 & x_1^2x_2 & x_1^2x_2^2 & x_1^2x_2^3 \\ x_1^3 & x_1^3x_2 & x_1^3x_2^2 & x_1^3x_2^3 \end{bmatrix}.$$

That is, $\boldsymbol{\Psi}[i_1, i_2](x) = x_1^{i_1-1}x_2^{i_2-1}$. Analogously, for a $d$-dimensional domain, we would obtain $\boldsymbol{\Psi}(x) \in \mathbb{R}^{k_1 \times \cdots \times k_d}$ with

$$\boldsymbol{\Psi}[i_1, \ldots, i_d](x) = x_1^{i_1-1} \cdots x_d^{i_d-1}. \qquad \triangle$$

Such a tensor basis is often used for high-dimensional problems, see also [61]. Typical basis functions are monomials, Hermite polynomials, or trigonometric functions. In the standard formulation, all basis functions are enumerated and rewritten in vector form (4.3). The difference here is that the tensor form will be preserved. We will use again (4.6) as a mapping from multi-index to single index when required.

Now $\mathbf{A}, \mathbf{G} \in \mathbb{R}^{k_1 \times \cdots \times k_d \times k_1 \times \cdots \times k_d}$ can be constructed as follows:

$$\begin{aligned} \mathbf{A}[i_1, \ldots, i_d, j_1, \ldots, j_d] &= \langle \mathcal{K}\boldsymbol{\Psi}[i_1, \ldots, i_d],\ \boldsymbol{\Psi}[j_1, \ldots, j_d] \rangle, \\ \mathbf{G}[i_1, \ldots, i_d, j_1, \ldots, j_d] &= \langle \boldsymbol{\Psi}[i_1, \ldots, i_d],\ \boldsymbol{\Psi}[j_1, \ldots, j_d] \rangle. \end{aligned}$$

The entries are again – as in the standard EDMD formulation – approximated using a collocation approach. EDMD computes the entries as shown in (4.4), for the new tensor-based formulation this results in

$$\begin{aligned} \mathbf{A}[i_1, \ldots, i_d, j_1, \ldots, j_d] &= \frac{1}{m} \sum_{l=1}^{m} \boldsymbol{\Psi}[i_1, \ldots, i_d](y_l)\boldsymbol{\Psi}[j_1, \ldots, j_d](x_l), \\ \mathbf{G}[i_1, \ldots, i_d, j_1, \ldots, j_d] &= \frac{1}{m} \sum_{l=1}^{m} \boldsymbol{\Psi}[i_1, \ldots, i_d](x_l)\boldsymbol{\Psi}[j_1, \ldots, j_d](x_l), \end{aligned}$$

or in short form, using the outer product,

$$\begin{aligned} \mathbf{A} &= \frac{1}{m} \sum_{l=1}^{m} \boldsymbol{\Psi}(y_l) \otimes \boldsymbol{\Psi}(x_l), \\ \mathbf{G} &= \frac{1}{m} \sum_{l=1}^{m} \boldsymbol{\Psi}(x_l) \otimes \boldsymbol{\Psi}(x_l), \end{aligned} \tag{4.10}$$

which in turn results in a generalized eigenvalue problem of the form

$$\boldsymbol{\xi}\mathbf{A} = \lambda\boldsymbol{\xi}\mathbf{G}.$$

Note that the eigenvalue problem is the same as (4.5) in the standard case. For the sake of simplicity, we are omitting the index $\mathbf{i}$ here.

**Proposition 4.4.5.** *Provided that the basis functions can be written in tensor product form (4.9),*

$$\boldsymbol{\xi}\mathbf{A} = \lambda\boldsymbol{\xi}\mathbf{G} \quad \Leftrightarrow \quad \xi\hat{A} = \lambda\xi\hat{G}.$$

*Proof.* We just show that the entries of $\mathbf{A}$ and $\hat{A}$ as well as the entries of $\mathbf{G}$ and $\hat{G}$ are identical, the rest – the equivalency of the matrix-vector and tensor products – follows from Proposition 4.4.2. Assuming that the basis can be written in the product form, we obtain from (4.4)

$$\begin{aligned}
\hat{a}_{\hat{i}\hat{j}} &= \frac{1}{m}\sum_{l=1}^{m}\psi_{\hat{i}}(y_l)\psi_{\hat{j}}(x_l) \\
&= \frac{1}{m}\sum_{l=1}^{m}\prod_{\mu=1}^{d}\psi_\mu^{i_\mu}(Q_\mu(y_l))\prod_{\mu=1}^{d}\psi_\mu^{j_\mu}(Q_\mu(x_l)) \\
&= \frac{1}{m}\sum_{l=1}^{m}\boldsymbol{\Psi}[i_1,\ldots,i_d](y_l)\boldsymbol{\Psi}[j_1,\ldots,j_d](x_l) \\
&= \mathbf{A}[i_1,\ldots,i_d,j_1,\ldots,j_d]
\end{aligned}$$

and analogously $\hat{g}_{\hat{i}\hat{j}} = \mathbf{G}[i_1,\ldots,i_d,j_1,\ldots,j_d]$. Here, $Q_\mu$ is again the projection onto the $\mu$-th component of a vector, cf. (4.8). □

Instead of storing the dense matrices $\mathbf{A}$ and $\mathbf{G}$, we can again directly represent these matrices using the canonical tensor format. The basis was chosen in such a way that $\boldsymbol{\Psi}(x)$ can be written as

$$\boldsymbol{\Psi}(x) = \bigotimes_{\mu=1}^{d}\tilde{\psi}_\mu(x_\mu),$$

where $\tilde{\psi}_\mu = [\psi_\mu^1,\ldots,\psi_\mu^{k_\mu}]^T \in \mathbb{R}^{k_\mu}$. With (4.10) it follows that $\mathbf{A}$ and $\mathbf{G}$ can be written as sums of $m$ elementary tensors. As before, we are not storing the full-rank tensor, but only low-rank approximations.

The eigentensors $\boldsymbol{\xi} \in \mathbb{R}^{k_1\times\cdots\times k_d}$ of the generalized eigenvalue problem can then be used to approximate the eigenfunctions of the Perron–Frobenius operator or Koopman operator: Let $\boldsymbol{\xi}$ be a left eigentensor, then

$$\varphi(x) = \langle\boldsymbol{\xi},\, \boldsymbol{\Psi}(x)\rangle$$

approximates an eigenfunction of the Koopman operator. Analogously, if $\boldsymbol{\xi}$ is a right eigentensor of the generalized eigenvalue problem – observe that $\mathbf{G}[i_1,\ldots,i_d,j_1,\ldots,j_d] = \mathbf{G}[j_1,\ldots,j_d,i_1,\ldots,i_d]$ –, then $\varphi(x)$ is an approximation of the corresponding eigenfunction of the Perron–Frobenius operator, see also [92].

To compute the dominant eigenfunctions of the Koopman operator or Perron–Frobenius operator, we will use simple power iteration schemes outlined in the next section. General purpose eigenvalue solvers for nonsymmetric generalized eigenvalue problems are, to our knowledge, not part of the tensor libraries yet. Solvers for symmetric (non-generalized) eigenvalue problems already exist and are part of the TT toolbox [143].

## 4.5 Eigenvalue problems

In Section 4.2 and Section 4.4, we have shown that in order to compute the eigenfunctions of the Perron–Frobenius operator and Koopman operator, respectively, using either Ulam's method or EDMD, we need to solve standard eigenvalue problems or generalized eigenvalue problems. For the reformulated version of these methods, we have to develop the required numerical algorithms to solve the resulting tensor-based eigenvalue problems. At the time of writing, we are not aware of any tensor toolbox containing numerical methods for nonsymmetric generalized eigenvalue problems. Methods for the computation of eigenvectors of symmetric positive definite matrices in the TT format have been proposed in [80], where the eigenvalue problem is rewritten as a (Rayleigh quotient based) minimization problem which is then solved using the *Alternating Linear Scheme* (ALS). In practice, these methods have recently also been successfully used for nonsymmetric problems, although convergence has not been shown yet [73].

Suitable methods for eigenvalue problems can be subdivided into two main categories as explained in [76] (see also references therein, e.g. [8]): The first category of methods is based on combining classical iterative algorithms with low-rank truncation after each step, the second is based on a reformulation as an optimization problem, where admissible solutions are constrained to the set of low-rank tensors. In this section, we will describe a generalization of simple power iteration methods – belonging to the first category – to tensor-based eigenvalue problems. For a detailed description of general power iteration methods, we refer to [75]. Power iteration and inverse power iteration for tensors have also been proposed in [8]. The main difference between the standard algorithms and the tensor-based counterpart is that for the latter truncation is used to keep the ranks of the tensors low. It is important that the iteration moves from the initial state to the final state without creating intermediate solutions with an excessive rank [8].

### 4.5.1 Power iteration methods for standard eigenvalue problems

In what follows, let $\mathcal{T}$ denote the truncation of a tensor. Then instead of a classical iteration scheme of the form $x_{k+1} = F(x_k)$, we simply obtain $\mathbf{x}_{k+1} = \mathcal{T}(F(\mathbf{x}_k))$. For an eigenvalue problem of the form $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$, given an initial guess $\mathbf{v}_0$ for the dominant eigenvector, the power iteration algorithm computes:

**for** $k = 1, 2, \ldots$ **do**
$\mathbf{w}^{(k)} = \mathcal{T}\left(\mathbf{A}\mathbf{v}^{(k-1)}\right)$
$\mathbf{v}^{(k)} = \mathbf{w}^{(k)} / \left\|\mathbf{w}^{(k)}\right\|$
$\lambda^{(k)} = \left\langle \mathbf{v}^{(k)}, \mathcal{T}(\mathbf{A}\mathbf{v}^{(k)}) \right\rangle$
**end for**

The iteration converges to an eigenvector associated with the largest eigenvalue $\lambda_1$ of the truncated operator $\mathbf{A}$ if the eigenvalue is simple and the initial guess $\mathbf{v}^{(0)}$ has a component in the direction of the corresponding dominant eigenvector $\mathbf{v}_1$ [75]. Even if the initial guess does not have a component in the direction of $\mathbf{v}_1$, rounding errors typically ensure that this direction will be picked up during the iteration. The rate of convergence depends on

the ratio between the second-largest and largest eigenvalue $\lambda_2/\lambda_1$. The main advantage of this method is that it requires only matrix-vector multiplications and can thus easily be used for tensor eigenvalue problems.

A modification of this algorithm to compute eigenvectors corresponding to any eigenvalue is the inverse power iteration with shift, which – assuming that $\mathbf{A}-\theta\mathbf{I}$ is nonsingular – can be written in the form:

**for** $k = 1, 2, \ldots$ **do**
  Solve $(\mathbf{A} - \theta\mathbf{I})\mathbf{w}^{(k)} = \mathbf{v}^{(k-1)}$
  $\mathbf{v}^{(k)} = \mathbf{w}^{(k)} / \left\|\mathbf{w}^{(k)}\right\|$
  $\lambda^{(k)} = \left\langle \mathbf{v}^{(k)}, \mathbf{A}\mathbf{v}^{(k)} \right\rangle$
**end for**

The parameter $\theta$ is called shift and the iteration converges to the eigenvalue closest to $\theta$. This method is just the standard power iteration applied to the matrix $(\mathbf{A}-\theta\mathbf{I})^{-1}$. Here, the linear solver computes a low-rank approximation of the solution so that truncation is not required.

### 4.5.2 Power iteration methods for generalized eigenvalue problems

Given matrices $\mathbf{A}$ and $\mathbf{B}$, the generalized eigenvalue problem is given by $\mathbf{A}\mathbf{v} = \lambda\mathbf{B}\mathbf{v}$. In this case, the power iteration method also requires the solution of a linear system of equations. Thus, we can also directly apply the inverse power iteration, where the resulting systems of linear equations are again solved with ALS:

**for** $k = 1, 2, \ldots$ **do**
  Solve $(\mathbf{A} - \theta\mathbf{B})\mathbf{w}^{(k)} = \mathbf{B}\mathbf{v}^{(k-1)}$
  $\mathbf{v}^{(k)} = \mathbf{w}^{(k)} / \left\|\mathbf{w}^{(k)}\right\|$
  $\lambda^{(k)} = \left\langle \mathbf{v}^{(k)}, \mathbf{A}\mathbf{v}^{(k)} \right\rangle / \left\langle \mathbf{v}^{(k)}, \mathbf{B}\mathbf{v}^{(k)} \right\rangle$
**end for**

In order to keep the ranks of the intermediate solutions low, we also approximate the matrices $\mathbf{P}$ (Ulam's method) or $\mathbf{A}$ and $\mathbf{G}$ (EDMD) by low-rank tensors. That is, the initial matrices are converted to $\tilde{\mathbf{P}}$, $\tilde{\mathbf{A}}$, and $\tilde{\mathbf{G}}$ with a lower rank since the rank of these matrices can initially be very high. We are typically only interested in the general behavior of the eigenfunctions, a highly accurate representation of the eigenfunctions is often not needed.

## 4.6 Examples

All examples presented within this section have been implemented in Matlab using – for the sake of efficiency – `mex`-functions to integrate the SDEs. All tensor computations were carried out with the *TT toolbox* [143]. For the eigenvector computations, we used our implementation of the simple power iteration methods described in Section 4.5.

### 4.6.1 2-dimensional double-well problem

Let us start with a simple 2-dimensional example, a stochastic differential equation of the form

$$dx_1 = -\nabla_{x_1} V(x_1, x_2)\, dt + \sigma\, dW_1,$$
$$dx_2 = -\nabla_{x_2} V(x_1, x_2)\, dt + \sigma\, dW_2,$$

where $W_1$ and $W_2$ are two independent standard Wiener processes. Here, the potential is given by

$$V(x_1, x_2) = (x_1^2 - 1)^2 + x_2^2,$$

see Figure 4.3 (cf. [92]). Furthermore, we set $\sigma = 0.7$. Note that in this case the potential can be written as $V(x_1, x_2) = V_1(x_1) + V_2(x_2)$. In order to analyze the tensor-based

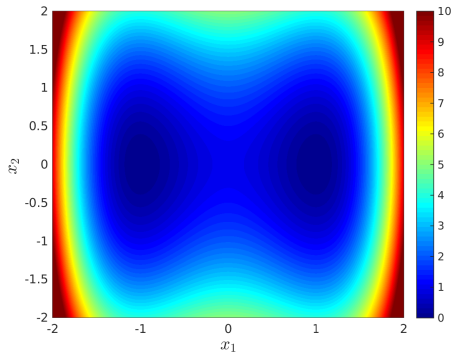



**Figure 4.3:** Double-well potential $V(x_1, x_2) = (x_1^2 - 1)^2 + x_2^2$.

methods, we rotate the potential by an angle $\alpha$ and obtain

$$\widetilde{V}(x_1, x_2) = \left((\cos(\alpha)\, x_1 - \sin(\alpha)\, x_2)^2 - 1\right)^2 + (\sin(\alpha)\, x_1 + \cos(\alpha)\, x_2)^2\,.$$

The two independent Wiener processes $W_1$ and $W_2$ are rotated accordingly. We would expect that the eigenfunctions of systems with small $\alpha$ can be accurately approximated by low-rank tensors, whereas systems with a larger value of $\alpha$ require higher ranks since the dynamics are not aligned with the axes anymore.

The second eigenfunctions of the Perron–Frobenius operator for the systems with potential $\widetilde{V}$ and different values of $\alpha$ computed using the tensor-based version of Ulam's method are shown in Figure 4.4. We chose $\alpha = 0$, $\alpha = \pi/12$, $\alpha = \pi/6$, and $\alpha = \pi/4$. The domain $\mathcal{X} = [-2, 2]^2$ was subdivided into $50 \times 50$ equally sized boxes. That is, $\mathbf{P} \in \mathbb{R}^{50\times 50\times 50\times 50}$. For each box, 100 randomly chosen test points were generated. The Euler–Maruyama method with a step size $h = 10^{-3}$ was used for the numerical integration, where one evaluation of $S$ corresponds to $10,000$ integration steps, that is, the integration interval is $[0, 10]$. The shift parameter $\theta$ of the power iteration method was

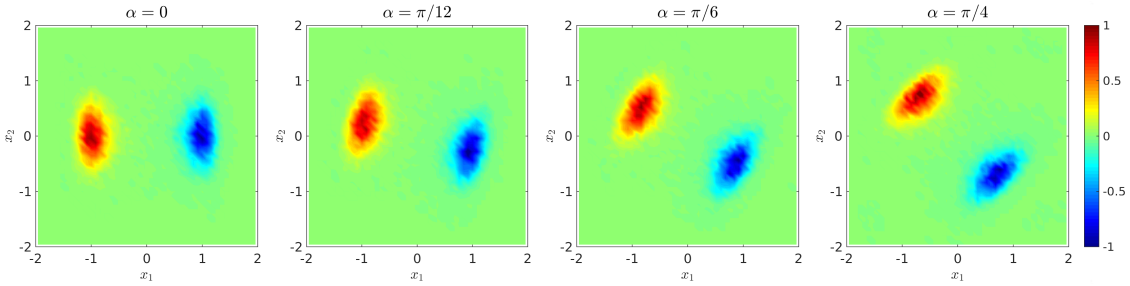



**Figure 4.4:** Second eigenfunction of the Perron–Frobenius operator for different values of $\alpha$.

set to a value slightly smaller than 1. Figure 4.5 illustrates how the tensor approximation, depending on the rank, successively picks up the information about the shape of the eigenfunction and generates more and more accurate representations. For $\alpha = \pi/4$, a tensor of rank 1 cannot represent the minimum and maximum simultaneously since this would lead to two additional peaks in the lower left and upper right corner. Here, the first pair of singular vectors represents the maximum, the second pair of singular vectors the minimum.

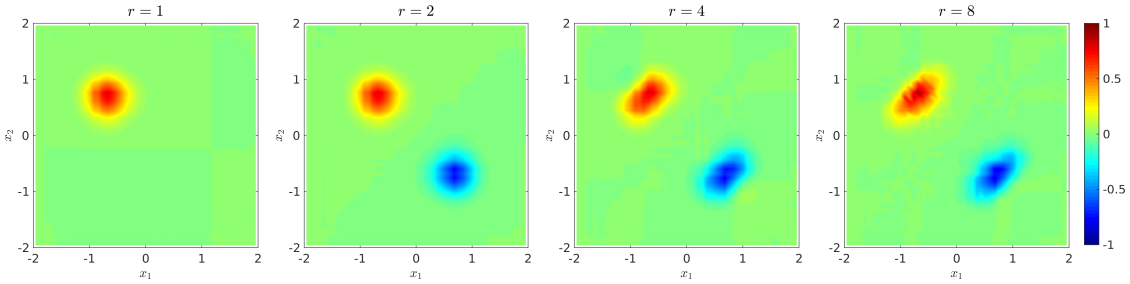



**Figure 4.5:** Tensor approximations of the second eigenfunction of the Perron–Frobenius operator for $\alpha = \pi/4$ with increasing rank $r$.

Additionally, we computed the eigenfunctions with the standard version of Ulam's method to evaluate the accuracy of the approximation and compared it with the results obtained by using the new tensor-based formulation. Figure 4.6 shows the influence of the truncation of the operator as well as the influence of the truncation of the resulting eigenfunctions. Here, in order to analyze the accuracy, we also compare the first eigenfunction with the analytically computed invariant density. Since we are computing eigenfunctions of the Perron–Frobenius operator associated with a stochastic differential equation, the results depend strongly on the number of test points chosen for each box. The higher the number of test points per box, the smoother the eigenfunction approximation. Thus, in this case, the smoother low-rank solutions can counterintuitively lead to better approximations of the true eigenfunctions. The high ranks are mainly required to resolve

the numerical noise introduced by the coarse approximation of the operator. This can be seen, for example, in Figure 4.6b. Decreasing the rank initially reduces the error – the truncation of the operator results in smoother eigenfunctions – until the shape of the eigenfunction cannot be described by a low-rank approximation anymore and the error increases. For $\alpha = 0$, the $x_1$ and $x_2$ dynamics are independent and a low-rank approximation is sufficient. Furthermore, the results illustrate that for a fixed-rank approximation, the error is smaller when the system's dynamics are aligned with the axes.

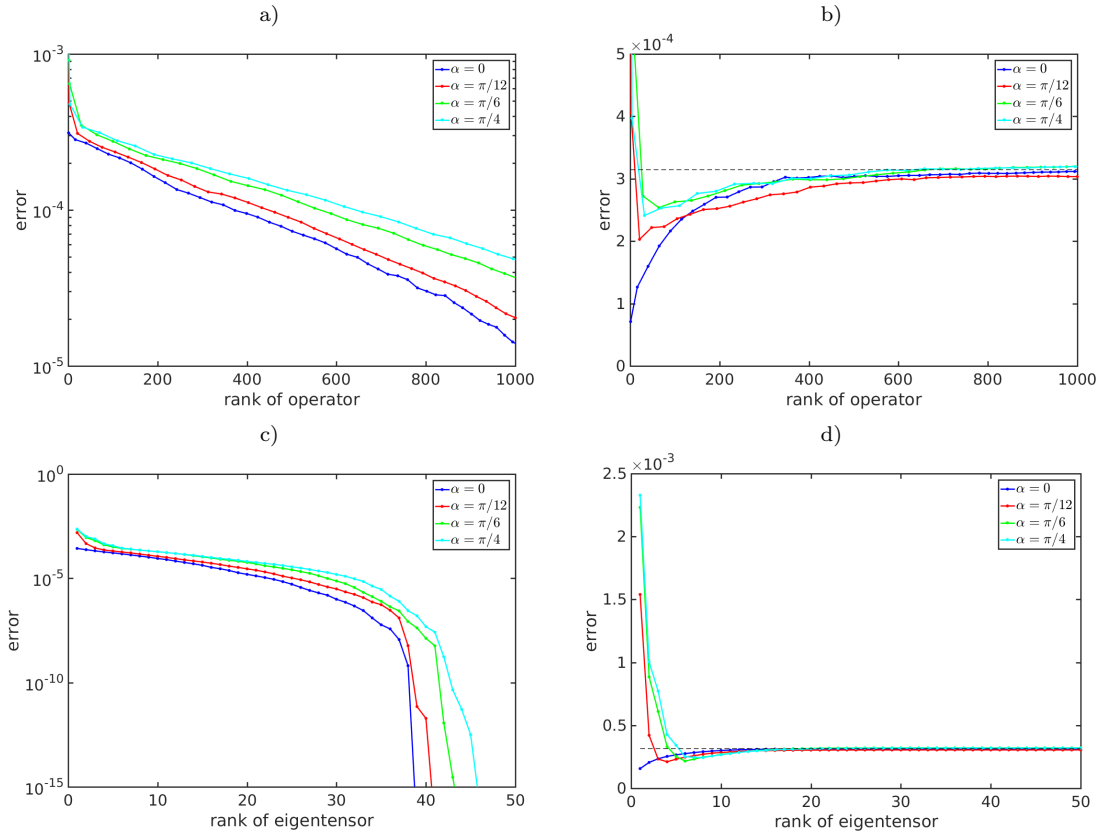



**Figure 4.6:** (Top) Let $\hat{v}_1$ denote the first eigenvector of $\hat{P}$, $v_1$ the (vectorized) eigentensor of the truncated tensor representation $\mathbf{P}$, and $\mu_{\text{inv}}$ the analytically computed invariant density. The error here is defined by $e = \frac{1}{k} \|v_1 - \hat{v}_1\|_2$ and $e = \frac{1}{k} \|v_1 - \mu_{\text{inv}}\|_2$, respectively, where $k$ is the number of boxes. a) Difference between $v_1$ and $\hat{v}_1$ depending on the rank of $\mathbf{P}$. b) Difference between $v_1$ and $\mu_{\text{inv}}$. (Bottom) Influence of the truncation of the first eigentensor $v_1$ of the full tensor representation $\mathbf{P}$ on the accuracy. c) Difference between the truncated eigentensor $v_1$ and $\hat{v_1}$. d) Difference between the truncated eigentensor $v_1$ and $\mu_{\text{inv}}$. The dashed lines show the error for the full-rank approximation which is almost identical for the different values of $\alpha$.

This example shows that in order to be able to approximate eigenfunctions by low-rank

tensors, the dynamics of the system should be aligned with the axes chosen, although even if the dynamics are not aligned, the tensor format might be advantageous. In general, the dynamics are unknown a priori and not necessarily aligned with the axes, but for higher-dimensional systems it is often possible to decompose a system into slow and fast subsystems. Not all variables of a system might be equally important to describe the system's behavior. The intuition would be that certain subsystems require less information and that the tensor approximation automatically captures the relevant dynamics, using high ranks only when necessary.

### 4.6.2 3-dimensional triple-well problem

Let us consider a more complex 3-dimensional example with the potential function

$$V(x_1, x_2, x_3) = 3e^{-x_1^2-(x_2-\frac{1}{3})^2} - 3e^{-x_1^2-(x_2-\frac{5}{3})^2} - 5e^{-(x_1-1)^2-x_2^2} - 5e^{-(x_1+1)^2-x_2^2} + \tfrac{2}{10}x_1^4 + \tfrac{2}{10}\left(x_2 - \tfrac{1}{3}\right)^4 + x_3^2.$$

This is a potential taken from [171], augmented by a third dimension. We subdivided the domain $\mathcal{X} = [-2, 2] \times [-1, 2] \times [-2, 2]$ into $20 \times 20 \times 20$ boxes of the same size. For each box, we randomly generated 1000 test points. The resulting finite-dimensional approximation is then a tensor $\mathbf{P} \in \mathbb{R}^{20\times20\times20\times20\times20\times20}$. Figure 4.7 shows a scatter plot of the first three eigenfunctions, which were computed using the inverse power iteration described in Section 4.5. The first eigenfunction clearly shows the three expected regions with high probabilities corresponding to the minima of the potential $V$. The second eigenfunction separates the two deeper wells at $(-1, 0, 0)$ and $(1, 0, 0)$, the third eigenfunction separates these wells from the third shallower well at $(0, 1.5, 0)$. The differences between the eigenfunctions computed using the conventional and the tensor-based version of Ulam's method are negligible, the average difference between the first eigenvector $v_1$ and the tensor $\mathbf{v}_1$ is of the order of $10^{-6}$, which is mainly due to the less accurate power iteration method applied to the tensor eigenvalue problem.

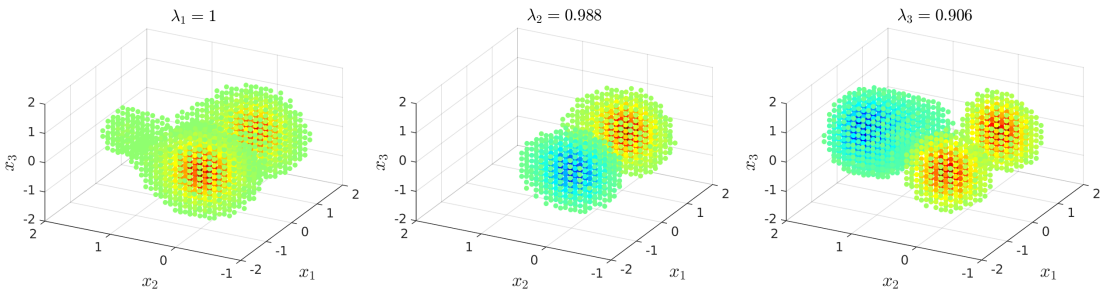



**Figure 4.7:** Scatter plot of the first three eigenfunctions of the Perron–Frobenius operator for the triple-well system. Only entries whose absolute value is larger than a given threshold are plotted, entries close to zero are omitted.

For the sake of comparison, we computed the eigenfunctions of the Koopman operator using EDMD. We chose basis functions $\mathcal{D} = \{x_1^{i_1} x_2^{i_2} x_3^{i_3},\, i_1, i_2, i_3 = 0, \ldots, 5\}$. Hence, $\mathbf{A}, \mathbf{G} \in \mathbb{R}^{6\times6\times6\times6\times6\times6}$. For higher-order monomials, the resulting matrices are ill-conditioned and the eigenfunctions cannot be computed accurately anymore. The results are shown in Figure 4.8. As before, the second eigenfunction separates the two deeper wells. The third eigenfunction separates these wells from the third shallower well and is close to zero for the regions around the deep wells. The trivial eigenfunction of the Koopman operator corresponding to $\lambda_1 = 1$ is not plotted here since it is constant and does not contain relevant information about the system. Note that the eigenvalues are slightly different due to the different set of basis functions used for EDMD.

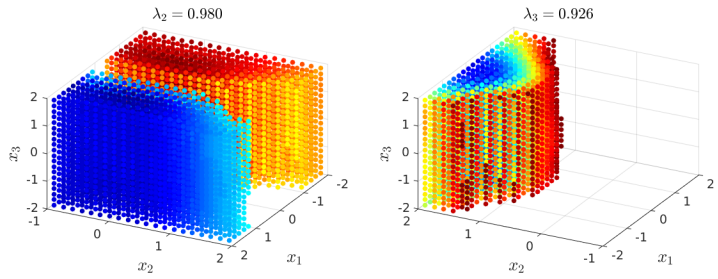



**Figure 4.8:** Scatter plot of the second and third eigenfunction of the Koopman operator for the triple-well system. Note that compared to the other plots the second eigenfunction of the Koopman operator is rotated by 180 degrees around the $x_3$ axis for a better visualization.

## 4.7 Conclusion

We have reformulated the problems of computing finite-dimensional approximations of the Perron–Frobenius and Koopman operator in a different format using tensors instead of vectors. The matrices $\mathbf{P}$ (if Ulam's method is used) or $\mathbf{A}$ and $\mathbf{G}$ (if EDMD is used) can now either be assembled in the dense tensor format or directly in the canonical tensor format – which can then easily be converted into the TT format –, enabling low-rank approximations of the aforementioned operators. The next step is to systematically develop the numerical methods required to efficiently solve the resulting nonsymmetric generalized tensor eigenvalue problems and also to store and handle these tensors minimizing memory requirements so that even high-dimensional problems can be solved. First results obtained by applying simple algorithms such as power iteration methods are promising and show that the approaches presented within this paper might be able to tackle high-dimensional problems. Currently, several toolboxes for tensor-based problems are under development. Once these toolboxes contain methods for solving nonsymmetric generalized eigenvalue problems, the proposed approaches can be implemented easily, potentially facilitating the computation of meta-stable sets or almost invariant sets of dynamical systems that could

not be handled before due to the curse of dimensionality. We demonstrated the tensor-based version of Ulam's method and EDMD using two- and three-dimensional problems mainly because the results can be easily validated and visualized.

Future work also includes determining which tensor format is suited best for our purposes. Currently, one of the main bottlenecks is the simulation required to obtain the data. Even simple molecular dynamics simulations, for instance, might easily take several days, but in order to capture the behavior of the system, long trajectories or a large number of short simulations with different initial conditions are required. This huge amount of data must then be processed. Thus, also the construction of the matrices $\mathbf{P}$ or $\mathbf{A}$ and $\mathbf{G}$ is time-consuming. The number of boxes or basis functions required to represent each variable of the system accurately is in general unknown a priori. If we are, for instance, only interested in the leading eigenfunctions of the Koopman operator, the fast variables of the system are typically almost constant and require less information to be captured. Starting with a set of only a few basis functions for each unknown, which is then, if needed, augmented adaptively based on the system's behavior would greatly improve the efficiency. Adaptive methods combined with (sparse) tensor approaches might be able to tackle high-dimensional systems and diminish the curse of dimensionality. Furthermore, a detailed numerical analysis of the efficiency and accuracy of the proposed algorithms would help understand the limitations and find opportunities for improvement.

Another open problem is the – depending on the number of dimensions $d$ – typically extremely large condition number of the matrices $\mathbf{A}$ or $\mathbf{G}$ if EDMD with, for instance, monomials are used to compute the eigenfunctions of the Perron–Frobenius or Koopman operator. Hence, the resulting eigenvalue problems cannot be solved accurately anymore for high-dimensional systems. A detailed understanding and numerical analysis of different basis functions might help mitigate this problem. Radial-basis functions or other more locally defined functions could lead to better results.

## Acknowledgements

This research has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114. Moreover, we would like to thank the reviewers for their helpful comments and suggestions.

# 5

# Tensor-based dynamic mode decomposition

Stefan Klus[1], Patrick Gelß[1], Sebastian Peitz[2], and Christof Schütte[1,3]

[1] Department of Mathematics and Computer Science, Freie Universität Berlin, Germany
[2] Department of Mathematics, University of Paderborn, Germany
[3] Zuse Institute Berlin, Germany

*Nonlinearity*, 31(7): 3359–3380, 2018

DOI: 10.1088/1361-6544/aabc8f

**Abstract**

Dynamic mode decomposition (DMD) is a recently developed tool for the analysis of the behavior of complex dynamical systems. In this paper, we will propose an extension of DMD that exploits low-rank tensor decompositions of potentially high-dimensional data sets to compute the corresponding DMD modes and eigenvalues. The goal is to reduce the computational complexity and also the amount of memory required to store the data in order to mitigate the curse of dimensionality. The efficiency of these tensor-based methods will be illustrated with the aid of several different fluid dynamics problems such as the von Kármán vortex street and the simulation of two merging vortices.

## 5.1 Introduction

Dynamic mode decomposition, which was first introduced in [163], is a powerful tool for analyzing the behavior of complex dynamical systems and can, for instance, be used to identify low-order dynamics [186]. Over the last years, several variants such as exact, optimized, or sparsity-promoting DMD have been proposed [163, 186, 28, 87]. It was also shown that DMD is closely related to the Koopman operator analysis. A generalization of DMD called extended dynamic mode decomposition (EDMD) is presented in [195] and has been developed for the approximation of the Koopman operator and its eigenvalues, eigenfunctions, and eigenmodes. In the same way, DMD and EDMD can be used to approximate the Perron–Frobenius operator – the adjoint of the Koopman operator – as shown in [92].

The amount of data that can be analyzed using methods like DMD or EDMD is limited. Due to the so-called *curse of dimensionality*, analyzing high-dimensional problems becomes infeasible. This can be mitigated by exploiting low-rank tensor approximation

approaches. Several tensor formats such as the canonical tensor format, the Tucker format, or the tensor-train format (TT-format) have been proposed. While the canonical format would be optimal from a simplicity and efficiency point of view, it was shown to be numerically unstable. This is due to the fact that the rank of canonical tensors is not lower semi-continuous and approximation problems using canonical tensors might become ill-posed [100]. The Tucker format, on the other hand, is numerically stable, the size of the core tensor, however, grows exponentially in the number of dimensions [80]. Our method relies on the tensor-train decomposition [142], which can be regarded as a multi-dimensional generalization of the conventional singular value decomposition (SVD) and combines the advantages of the canonical and Tucker format.

The goal of this paper is to extend DMD to use tensors instead of vectors so that low-rank approximations of the data can be utilized in order to reduce the computational complexity and also the amount of memory required to store the data and the resulting linear operators and their eigenfunctions, eigenvalues, and corresponding modes. To this end, we will have to compute singular value decompositions and pseudoinverses of tensor unfoldings. Recently, a method to compute an approximation of the pseudoinverse of operators in tensor-train format, which relies on the solution of an optimization problem, was proposed in [108]. The optimization problem is solved with the Modified Alternating Linear Scheme (MALS) [80], breaking the problem into smaller subproblems which can be solved with conventional methods. We will use a different approach here. Since the pseudoinverse of a matrix can be computed based on a singular value decomposition, the question now is whether the TT-representation of a tensor itself already contains information about the pseudoinverse that can be exploited without necessitating the solution of an optimization problem.

The outline of this paper is as follows: Chapter 5.2 briefly recapitulates the properties of the pseudoinverse of a matrix and describes standard and exact DMD. Chapter 5.3 introduces tensors and different tensor representations, in particular the TT-format. Furthermore, we will show how the pseudoinverse required for DMD can be obtained from a given tensor decomposition of the data. Then we will present a reformulation of DMD using tensors instead of matrices and vectors. In Chapter 5.4, numerical results for different fluid dynamics problems will be presented. Chapter 5.5 concludes with a brief summary, open problems, and future work.

## 5.2 Dynamic mode decomposition

In this section, we will introduce DMD and show how the DMD modes and eigenvalues can be computed efficiently using a reduced singular value decomposition of the data. The goal is then to rewrite these methods in terms of tensors, exploiting low-rank decompositions of potentially high-dimensional data. These tensor decomposition techniques will be introduced in Section 5.3.

DMD decomposes high-dimensional data into coupled spatial-temporal modes and can be regarded as a combination of a principal component analysis (PCA) in the spatial

domain and a Fourier analysis in the frequency domain [19]. The DMD modes often correspond to coherent structures in the flow, the DMD eigenvalues can be interpreted as growth rates and frequencies of the corresponding modes. Thus, it is often used to analyze oscillatory behavior [186]. Originally applied to time series data, DMD has been generalized to analyze pairs of $n$-dimensional data vectors $x_i$ and $y_i = F(x_i)$, $i = 1, \dots, m$, written in matrix form as

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}. \tag{5.1}$$

Here, $F : \mathbb{R}^n \to \mathbb{R}^n$ can be any (non-)linear dynamical system. Thus, for a given time series $\{z_0, z_1, \dots, z_m\}$, that is $z_i = F^i(z_0)$, we obtain

$$X = \begin{bmatrix} z_0 & z_1 & \cdots & z_{m-1} \end{bmatrix} \quad \text{and} \quad Y = \begin{bmatrix} z_1 & z_2 & \cdots & z_m \end{bmatrix}. \tag{5.2}$$

Assuming there exists a linear operator $A$ that describes the dynamics of the system such that $y_i = Ax_i$, define

$$A = YX^+, \tag{5.3}$$

where $^+$ denotes the pseudoinverse. The matrix $A$ then minimizes the cost function $\|AX - Y\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm. The DMD modes and eigenvalues are then defined to be the eigenvectors and eigenvalues of $A$. Using $X^+ = X^T(XX^T)^+$, we obtain

$$A = (YX^T)(XX^T)^+. \tag{5.4}$$

The difference is that now the pseudoinverse of an $(n \times n)$-matrix needs to be computed and not the pseudoinverse of an $(n \times m)$-matrix, which is advantageous if $n \ll m$. In our fluid dynamics examples, however, the dimension $n$ of the problem is typically much larger than the number of snapshots $m$.

### 5.2.1 Singular value decomposition and the pseudoinverse

Before we begin with the conventional formulation of the DMD algorithm, let us briefly recapitulate the properties of the standard singular value decomposition and the standard pseudoinverse of a matrix. These properties and definitions will be used and generalized in Section 5.3 for the computation of singular value decompositions and pseudoinverses of tensor unfoldings.

Let $M \in \mathbb{R}^{n_1 \times n_2}$ be a matrix. It is well known that the pseudoinverse $M^+ \in \mathbb{R}^{n_2 \times n_1}$ can be computed by a (compact/reduced) singular value decomposition of $M$. Assume that $M = U\,\Sigma\,V^T$, where $U \in \mathbb{R}^{n_1 \times s}$, $V \in \mathbb{R}^{n_2 \times s}$, and $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_s) \in \mathbb{R}^{s \times s}$ is a diagonal matrix containing only the nonzero singular values. Then the pseudoinverse is given by

$$M^+ = V\,\Sigma^{-1}\,U^T,$$

where $\Sigma^{-1} = \operatorname{diag}(\sigma_1^{-1}, \dots, \sigma_s^{-1})$. In what follows, we will assume that the singular values are sorted in decreasing order, i.e. $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_s > 0$. Let us now recall

the definition of the tensor product. For two vectors $v \in \mathbb{R}^{n_1}$ and $w \in \mathbb{R}^{n_2}$, the tensor product $v \otimes w \in \mathbb{R}^{n_1 \times n_2}$ is given by

$$(v \otimes w)_{i,j} = (v \cdot w^T)_{i,j} = v_i \cdot w_j. \tag{5.5}$$

**Example 5.2.1.** For a rank-$s$ matrix

$$M = U \Sigma V^T = \sum_{i=1}^{s} \sigma_i u_i v_i^T = \sum_{i=1}^{s} \sigma_i u_i \otimes v_i,$$

we would obtain

$$M^+ = V \Sigma^{-1} U^T = \sum_{i=1}^{s} \frac{1}{\sigma_i} v_i u_i^T = \sum_{i=1}^{s} \frac{1}{\sigma_i} v_i \otimes u_i. \qquad \triangle$$

The example shows that in order to obtain the pseudoinverse we simply exchange the roles of the left and right singular vectors and divide each tensor product $v_i \otimes u_i$ by the corresponding singular value $\sigma_i$. These properties will be used later on to construct the pseudoinverse of a tensor given in TT-format, which is based on successive singular value decompositions.

### 5.2.2 Computation of DMD modes and eigenvalues

There are different algorithms to compute the DMD modes and eigenvalues without explicitly computing $A$ which rely on a compact singular value decomposition of $X$. The standard DMD version is shown in Algorithm 5.1.

**Algorithm 5.1** Standard DMD algorithm.

1: Compute the compact SVD of $X$, given by $X = U \Sigma V^T$ with $U \in \mathbb{R}^{n \times s}$, $V \in \mathbb{R}^{m \times s}$, and $\Sigma \in \mathbb{R}^{s \times s}$.
2: Define $\tilde{A} = U^T Y V \Sigma^{-1}$.
3: Compute eigenvalues and eigenvectors of $\tilde{A}$, i.e. $\tilde{A} w = \lambda w$.
4: The DMD mode corresponding to the eigenvalue $\lambda$ is then defined as $\varphi = Uw$.

Algorithm 5.2 shows a variant called exact DMD. The modes computed by the standard DMD algorithm are simply the modes computed by the exact DMD algorithm projected onto the range of $X$. For a more detailed description, we refer to [186].

**Algorithm 5.2** Exact DMD algorithm.

1: Execute steps 1 to 3 of Algorithm 5.1.
2: The DMD mode corresponding to the eigenvalue $\lambda$ is then defined as $\varphi = \frac{1}{\lambda} Y V \Sigma^{-1} w$.

The matrix $A$ defined in (5.3) and the matrix $\tilde{A}$ given in Algorithm 5.1 share the same eigenvalue spectrum. However, the DMD algorithm computes only the nonzero

eigenvalues. If $\lambda$ is an eigenvalue of $A$ corresponding to the eigenvector $\varphi$, i.e. $A\varphi = \lambda\varphi$, then it follows that $\tilde{A}w = \lambda w$ with $w = U^T\varphi$. Conversely, if we have $\tilde{A}w = \lambda w$ and define $\varphi = \frac{1}{\lambda}YV\Sigma^{-1}w$ as in Algorithm 5.2, then $Av = \lambda v$. A proof of this result can also be found in [186]. In order to illustrate the idea behind DMD, let us analyze a simple example.

**Example 5.2.2.** Consider the von Kármán vortex street. The system describes the flow past a two-dimensional cylinder which – for a certain range of Reynolds numbers – results in a repeating pattern of vortices. A similar example and more details about the characteristic properties of the system can be found in [186]. Snapshots of the solution of this partial differential equation at intermediate time steps are shown in Figure 5.1a. We discretized the domain using a $60 \times 120$ grid and generated 101 snapshots of the solution at equidistant time points. The corresponding DMD modes are shown in Figure 5.1b. △

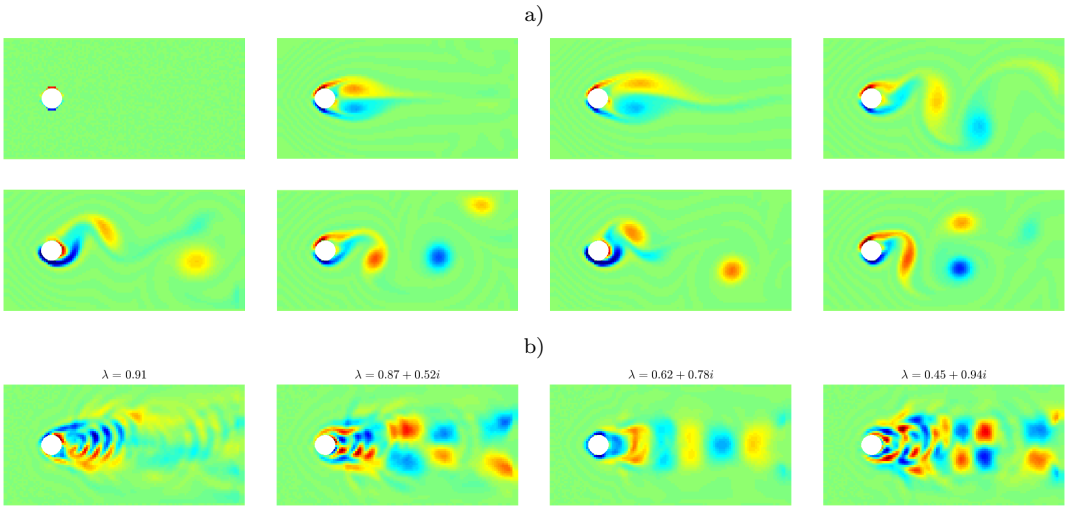



**Figure 5.1:** a) Simulation of the von Kármán vortex street. b) DMD modes corresponding to eigenvalues close to 1.

Note that for this example we reshaped each $60 \times 120$ snapshot matrix as a vector, resulting in data matrices $X, Y \in \mathbb{R}^{7200\times100}$. Our reformulation of DMD, on the other hand, will directly operate on tensors $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{60\times120\times100}$.

## 5.3 Tensor-based dynamic mode decomposition

In this section, we will first introduce different tensor formats and then show how the pseudoinverse of a tensor unfolding can be computed efficiently exploiting properties of the TT-decomposition. Furthermore, we will rewrite DMD in terms of tensors. The aim is to exploit low-rank tensor approximation methods to analyze high-dimensional dynamical systems that could not be handled before due to the curse of dimensionality.

### 5.3.1 The TT-format

Over the last years, different tensor formats such as the canonical format, the Tucker format, and the tensor-train format have been developed, see e.g. [7, 76, 77, 78]. A frequently used format is the TT-format [142], also known as *Matrix Product States (MPS)* in the physics literature [149, 165]. A tensor $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is decomposed into $d$ component tensors $\mathbf{x}^{(i)}$ of at most order three, where the first and last component tensors of order two are, for simplicity, often interpreted as tensors of order three with the additional requirement that $r_0 = r_d = 1$. The entries of $\mathbf{x}$ are given by

$$\begin{aligned} \mathbf{x}_{i_1,\ldots,i_d} &= \sum_{k_0=1}^{r_0} \cdots \sum_{k_d=1}^{r_d} \mathbf{x}^{(1)}_{k_0,i_1,k_1} \cdot \mathbf{x}^{(2)}_{k_1,i_2,k_2} \cdot \ldots \cdot \mathbf{x}^{(d-1)}_{k_{d-2},i_{d-1},k_{d-1}} \cdot \mathbf{x}^{(d)}_{k_{d-1},i_d,k_d} \\ &= \mathbf{x}^{(1)}_{:,i_1,:} \cdot \mathbf{x}^{(2)}_{:,i_2,:} \cdot \ldots \cdot \mathbf{x}^{(d-1)}_{:,i_{d-1},:} \cdot \mathbf{x}^{(d)}_{:,i_d,:}, \end{aligned} \tag{5.6}$$

where the more compact second formulation uses Matlab's colon notation. With the aid of the tensor product (5.5), the whole tensor can then be represented as

$$\mathbf{x} = \sum_{k_0=1}^{r_0} \cdots \sum_{k_d=1}^{r_d} \mathbf{x}^{(1)}_{k_0,:,k_1} \otimes \mathbf{x}^{(2)}_{k_1,:,k_2} \otimes \cdots \otimes \mathbf{x}^{(d-1)}_{k_{d-2},:,k_{d-1}} \otimes \mathbf{x}^{(d)}_{k_{d-1},:,k_d}. \tag{5.7}$$

The vector $r = [r_0, \ldots, r_d]$ contains the ranks of the TT-tensor and determines the complexity of the representation. The lower the ranks, the lower the memory consumption and the computational costs. One of the main advantages of the TT-format, compared to the canonical format, is its stability from an algorithmic point of view [36, 80].

For computational aspects, it is helpful to reshape tensors into matrices and vectors. In order to describe matricizations and vectorizations – also called *tensor unfoldings* –, we first define a bijection $\phi_N$ for the ordered set $N = (n_1, \ldots, n_d)$ by

$$\begin{aligned} \phi_N : \{1, \ldots, n_1\} \times \ldots \times \{1, \ldots, n_d\} &\to \{1, \ldots, \prod_{\mu=1}^{d} n_\mu\}, \\ (i_1, \ldots, i_d) &\mapsto \phi_N(i_1, \ldots, i_d). \end{aligned} \tag{5.8}$$

Using the *little-endian* convention, this bijection is defined as

$$\phi_N(i_1, \ldots, i_d) = 1 + (i_1 - 1) + \ldots + (i_d - 1) \cdot n_1 \cdot \ldots \cdot n_{d-1} = 1 + \sum_{\mu=1}^{d} (i_\mu - 1) \prod_{\nu=1}^{\mu-1} n_\nu. \tag{5.9}$$

If the definition of $N$ is clear from the context, we write $\phi_N(i_1, \ldots, i_d) = \overline{i_1, \ldots, i_d}$.

**Definition 5.3.1.** *Let $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a tensor. For the two ordered subsets $N' = (n_1, \ldots, n_l)$ and $N'' = (n_{l+1}, \ldots, n_d)$ of $N$, $1 \leq l \leq d-1$, the matricization $\mathbf{x}\Big|^{N''}_{N'} \in \mathbb{R}^{(n_1 \cdot \ldots \cdot n_l) \times (n_{l+1} \cdot \ldots \cdot n_d)}$ of $\mathbf{x}$ with respect to $N'$ and $N''$ is given by*

$$\left(\mathbf{x}\Big|^{N''}_{N'}\right)_{\phi_{N'}(i_1,\ldots,i_l),\phi_{N''}(i_{l+1},\ldots,i_d)} = \left(\mathbf{x}\Big|^{N''}_{N'}\right)_{\overline{i_1,\ldots,i_l},\overline{i_{l+1},\ldots,i_d}} = \mathbf{x}_{i_1,\ldots,i_d}. \tag{5.10}$$

*Accordingly, the vectorization* $\mathbf{x}\Big|_N \in \mathbb{R}^{n_1 \cdot \ldots \cdot n_d}$ *of a tensor* $\mathbf{x} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ *is given by a matricization of* $\mathbf{x}$ *with* $N' = N$ *and* $N'' = \varnothing$*, i.e.*

$$\left(\mathbf{x}\Big|_N\right)_{\phi_N(i_1,\ldots,i_d)} = \left(\mathbf{x}\Big|_N\right)_{\overline{i_1,\ldots,i_d}} = \mathbf{x}_{i_1,\ldots,i_d}. \tag{5.11}$$

**Remark 5.3.2.** Note that here and in what follows vectors $\mathbf{x}^{(\mu)}_{k_{\mu-1},:,k_\mu}$ as well as vectorizations $\mathbf{x}\Big|_N$ of tensors are regarded as column vectors.

If a tensor $\mathbf{x}$ is given in full format, i.e. $\mathbf{x}$ is indeed a $d$-dimensional array, Algorithm 5.3 can be used to compute an exact ($\varepsilon = 0$) or an approximated ($\varepsilon > 0$) TT-decomposition of $\mathbf{x}$, respectively. For more details, see [142]. Here, $U_\mu$ and $V_\mu$ denote the matrices containing the first $r_\mu$ singular vectors and $\Sigma_\mu$ denotes the diagonal matrix comprising the first $r_\mu$ singular values.

**Algorithm 5.3** Convert a tensor given in full format into the TT-format.

1: Given a tensor $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ in full format and a threshold $\varepsilon$.
2: **for** $\mu = 1, ..., d-1$ **do**
3: $\quad M = \mathbf{x}\Big|^{n_{\mu+1},\ldots,n_d}_{r_{\mu-1},n_\mu}$.
4: $\quad$ Compute SVD of $M$, i.e. $M = U_\mu \Sigma_\mu V_\mu^T$ with $\Sigma_\mu \in \mathbb{R}^{s \times s}$.
5: $\quad$ Set $r_\mu \leq s$ to the largest index such that $(\Sigma_\mu)_{i,i} > \varepsilon$ for $i \leq r_\mu$.
6: $\quad$ Discard rows and columns of $U_\mu$, $\Sigma_\mu$, and $V_\mu$ corresponding to singular values smaller than or equal to $\varepsilon$.
7: $\quad$ Set $\mathbf{y}^{(\mu)} \in \mathbb{R}^{r_{\mu-1} \times n_\mu \times r_\mu}$ to a reshaped version of $U_\mu$ with $\mathbf{y}^{(\mu)}_{k_{\mu-1},i_\mu,k_\mu} = (U_\mu)_{\overline{k_{\mu-1},i_\mu},k_\mu}$.
8: $\quad$ Define remainder $\mathbf{x} = \Sigma_\mu V_\mu^T \in \mathbb{R}^{r_\mu \times n_{\mu+1} \cdot \ldots \cdot n_d}$.
9: **end for**
10: Set $d$-th core to $\mathbf{y}^{(d)}_{:,:,1} = \mathbf{x}$.
11: The tensor $\mathbf{y}$ with cores $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(d)}$ and ranks $r_0, \ldots, r_d$ is then an approximation of the initial tensor $\mathbf{x}$.

Orthonormality of tensor trains plays an important role, in particular when we want to compute pseudoinverses of a tensor given in TT-format.

**Definition 5.3.3.** *A TT-core* $\mathbf{x}^{(\mu)} \in \mathbb{R}^{r_{\mu-1} \times n_\mu \times r_\mu}$ *is called* left-orthonormal *if*

$$\left(\mathbf{x}^{(\mu)}\Big|^{r_\mu}_{r_{\mu-1},n_\mu}\right)^T \cdot \left(\mathbf{x}^{(\mu)}\Big|^{r_\mu}_{r_{\mu-1},n_\mu}\right) = I \in \mathbb{R}^{r_\mu \times r_\mu}. \tag{5.12}$$

*Correspondingly,* $\mathbf{x}^{(\mu)}$ *is called* right-orthonormal *if*

$$\left(\mathbf{x}^{(\mu)}\Big|^{n_\mu,r_\mu}_{r_{\mu-1}}\right) \cdot \left(\mathbf{x}^{(\mu)}\Big|^{n_\mu,r_\mu}_{r_{\mu-1}}\right)^T = I \in \mathbb{R}^{r_{\mu-1} \times r_{\mu-1}}. \tag{5.13}$$

Naturally, the first $d-1$ TT-cores of $\mathbf{y}$ as constructed in Algorithm 5.3 are left-orthonormal while the last core is right-orthogonal, but not right-orthonormal, see the following Lemma.

**Lemma 5.3.4.** *Due to the construction of the TT-representation of a tensor as described in Algorithm 5.3, it holds that:*

1. $\mathbf{y}^{(\mu)}\Big|_{r_{\mu-1},n_\mu}^{r_\mu}$ *is left-orthonormal for* $\mu = 1,\dots,d-1$.

2. $\left(\mathbf{y}^{(d)}\Big|_{r_{d-1}}^{n_d,r_d}\right)\cdot\left(\mathbf{y}^{(d)}\Big|_{r_{d-1}}^{n_d,r_d}\right)^T = \Sigma_{d-1}^2$, *where* $\Sigma_{d-1}$ *is part of the last SVD in Algorithm 5.3.*

*Proof.* The properties are a result of the fact that the cores $\mathbf{y}^{(\mu)}$, $\mu = 1,\dots,d-1$, are reshaped versions of matrices $U_\mu$ with $U_\mu^T\cdot U_\mu = I$ and $\mathbf{y}^{(d)} = \Sigma_{d-1}\,V_{d-1}^T$ with $V_{d-1}^T\cdot V_{d-1} = I$. □

In general, we do, however, not want to compute the tensor in full format and then convert it to the TT-format. All the numerical computations should ideally be directly carried out in the TT-format. This could, for instance, mean solving systems of equations, eigenvalue problems, ordinary or partial differential equations, or completion problems using the TT-format, see e.g. [1, 49, 73, 48, 154]. In this way, we automatically compute a low-rank approximation of the solution without necessitating the conversion to the TT-format. The aim is now to exploit properties of the TT-decomposition in order to efficiently compute DMD modes and eigenvalues. For this purpose, we will need the following auxiliary results:

**Lemma 5.3.5.** *The tensor product satisfies*

$$\left(\left(\mathbf{x}^{(1)}\otimes\ldots\otimes\mathbf{x}^{(d)}\right)\Big|_{n_1,\ldots,n_d}\right)^T\cdot\left(\left(\mathbf{y}^{(1)}\otimes\ldots\otimes\mathbf{y}^{(d)}\right)\Big|_{n_1,\ldots,n_d}\right) = \prod_{\mu=1}^{d}\left(\mathbf{x}^{(\mu)}\right)^T\cdot\mathbf{y}^{(\mu)}$$

*and*

$$\left\|\left(\mathbf{x}^{(1)}\otimes\ldots\otimes\mathbf{x}^{(d)}\right)\Big|_{n_1,\ldots,n_d}\right\|_2 = \prod_{\mu=1}^{d}\left\|\mathbf{x}^{(\mu)}\right\|_2.$$

*Proof.* The first equation follows from

$$\begin{aligned}&\left(\left(\mathbf{x}^{(1)}\otimes\ldots\otimes\mathbf{x}^{(d)}\right)\Big|_{n_1,\ldots,n_d}\right)^T\cdot\left(\left(\mathbf{y}^{(1)}\otimes\ldots\otimes\mathbf{y}^{(d)}\right)\Big|_{n_1,\ldots,n_d}\right)\\ &\quad=\sum_{i_1=1}^{n_1}\cdots\sum_{i_d=1}^{n_d}\mathbf{x}_{i_1}^{(1)}\cdot\ldots\cdot\mathbf{x}_{i_d}^{(d)}\cdot\mathbf{y}_{i_1}^{(1)}\cdot\ldots\cdot\mathbf{y}_{i_d}^{(d)} = \prod_{\mu=1}^{d}\left(\mathbf{x}^{(\mu)}\right)^T\cdot\mathbf{y}^{(\mu)},\end{aligned}$$

the second is simply a special case of the first. □

### 5.3.2 Singular value decomposition and the pseudoinverse in TT-format

Before we consider TT-tensors with arbitrary ranks, let us illustrate the basic idea with a simple example.

**Example 5.3.6.** Assume that we have a rank-one tensor of the form $\mathbf{x} = \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)} \otimes \mathbf{x}^{(3)} \otimes \mathbf{x}^{(4)} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$, with $\mathbf{x}^{(i)} \in \mathbb{R}^{n_i}$, and we want to compute the pseudoinverse of the matricization with respect to the dimensions $(1, 2)$ and $(3, 4)$, given by

$$M = \left(\mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)}\right)\Big|_{n_1,n_2} \otimes \left(\mathbf{x}^{(3)} \otimes \mathbf{x}^{(4)}\right)\Big|_{n_3,n_4} \in \mathbb{R}^{(n_1 \cdot n_2) \times (n_3 \cdot n_4)}.$$

Then

$$M^+ = \frac{1}{\sigma^2}\left(\mathbf{x}^{(3)} \otimes \mathbf{x}^{(4)}\right)\Big|_{n_3,n_4} \otimes \left(\mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)}\right)\Big|_{n_1,n_2} \in \mathbb{R}^{(n_3 \cdot n_4) \times (n_1 \cdot n_2)},$$

with $\sigma = \prod_{i=1}^{4} \left\|\mathbf{x}^{(i)}\right\|_2$. This can be seen by writing

$$M = \underbrace{\left(\mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)}\right)\Big|_{n_1,n_2}}_{\tilde{u}} \otimes \underbrace{\left(\mathbf{x}^{(3)} \otimes \mathbf{x}^{(4)}\right)\Big|_{n_3,n_4}}_{\tilde{v}} = \tilde{u} \otimes \tilde{v} = \sigma \underbrace{\left(\frac{1}{\|\tilde{u}\|}\tilde{u}\right)}_{u} \otimes \underbrace{\left(\frac{1}{\|\tilde{v}\|}\tilde{v}\right)}_{v}$$

and thus

$$M^+ = \frac{1}{\sigma} v \otimes u = \frac{1}{\sigma^2} \tilde{v} \otimes \tilde{u}.$$

The formula for $\sigma$ follows from Lemma 5.3.5. With a slight abuse of notation, we will write

$$\mathbf{x}^+ = \frac{1}{\sigma^2} \mathbf{x}^{(3)} \otimes \mathbf{x}^{(4)} \otimes \mathbf{x}^{(1)} \otimes \mathbf{x}^{(2)}.$$

That is, the pseudoinverse can in this case simply be obtained by reordering the cores of the rank-1 tensor and by normalizing the tensor product. Note that if the tensor was obtained by the TT-decomposition, then the vectors $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, and $\mathbf{x}^{(3)}$ are already normalized and we have to divide only by $\left\|\mathbf{x}^{(4)}\right\|^2$. A graphical representation of this process is shown in Figure 5.2, where we use a similar diagrammatic notation as in [80]. △

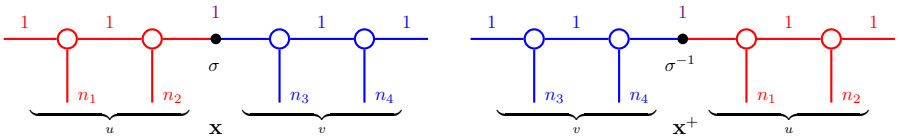



**Figure 5.2:** Pseudoinverse of a rank-1 tensor with respect to the dimensions $(1, 2)$ and $(3, 4)$. The first two and the last two cores are swapped and the tensor is divided by $\sigma$.

Note that the pseudoinverse depends on the matricization of the tensor $\mathbf{x}$. For a different matricization, for instance with respect to the dimensions $(1, 2, 3)$ and $(4)$, we would also obtain a different pseudoinverse $\mathbf{x}^+$. We now want to generalize this approach to compute pseudoinverses of arbitrary tensors $\mathbf{x}$ in TT-format for a given matricization. It turns out that also for tensors with a higher-rank coupling the pseudoinverse can be obtained by reordering the cores after some preprocessing steps. These include the left- and right-orthonormalization, respectively, of the given TT-cores.

**Algorithm 5.4** Left-orthonormalize TT-cores $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(l)}$.

1: Given tensor $\mathbf{x}$ and core number $l$, $1 \leq l \leq d-1$.
2: **for** $\mu = 1, \dots, l$ **do**
3: Compute QR factorization of $\mathbf{x}^{(\mu)}\big|^{r_\mu}_{r_{\mu-1},n_\mu}$, i.e. $\mathbf{x}^{(\mu)}\big|^{r_\mu}_{r_{\mu-1},n_\mu} = Q \cdot R$ with $Q \in \mathbb{R}^{r_{\mu-1}\cdot n_\mu \times s}$ and $Q^T \cdot Q = I$.
4: Define $\mathbf{y} \in \mathbb{R}^{r_{\mu-1}\times n_\mu \times s}$ as a reshaped version of $Q$ with $\mathbf{y}_{i,j,k} = Q_{\overline{i,j},k}$.
5: Define $\mathbf{z} \in \mathbb{R}^{s\times n_{\mu+1}\times r_{\mu+1}}$ by $\mathbf{z}\big|^{n_{\mu+1},r_{\mu+1}}_{s} = R \cdot \mathbf{x}^{(\mu+1)}\big|^{n_{\mu+1},r_{\mu+1}}_{r_\mu}$.
6: Set $\mathbf{x}^{(\mu)}$ to $\mathbf{y}$, $\mathbf{x}^{(\mu+1)}$ to $\mathbf{z}$ and $r_\mu$ to $s$.
7: **end for**

**Algorithm 5.5** Right-orthonormalize TT-cores $\mathbf{x}^{(l)}, \dots, \mathbf{x}^{(d)}$.

1: Given tensor $\mathbf{x}$ and core number $l$, $2 \leq l \leq d$.
2: **for** $\mu = d, \dots, l$ **do**
3: Compute QR factorization of $\left(\mathbf{x}^{(\mu)}\big|^{n_\mu,r_\mu}_{r_{\mu-1}}\right)^T$, i.e. $\mathbf{x}^{(\mu)}\big|^{n_\mu,r_\mu}_{r_{\mu-1}} = R^T \cdot Q^T$ with $Q^T \in \mathbb{R}^{s\times n_\mu \cdot r_\mu}$ and $Q^T \cdot Q = I$.
4: Define $\mathbf{y} \in \mathbb{R}^{s\times n_\mu \times r_\mu}$ as a reshaped version of $Q^T$ with $\mathbf{y}_{i,j,k} = Q^T_{i,\overline{j,k}}$.
5: Define $\mathbf{z} \in \mathbb{R}^{r_{\mu-2}\times n_{\mu-1}\times s}$ by $\mathbf{z}\big|^{s}_{r_{\mu-2},n_{\mu-1}} = \mathbf{x}^{(\mu-1)}\big|^{r_{\mu-1}}_{r_{\mu-2},n_{\mu-1}} \cdot R^T$.
6: Set $\mathbf{x}^{(\mu)}$ to $\mathbf{y}$, $\mathbf{x}^{(\mu-1)}$ to $\mathbf{z}$ and $r_{\mu-1}$ to $s$.
7: **end for**

Algorithm 5.4 shows the left-orthonormalization process. Similarly, the procedure for the right-orthonormalization is shown in Algorithm 5.5. If we apply the Algorithms 5.4 or 5.5 to a tensor $\mathbf{x}$, then the tensor itself remains the same, the algorithms simply compute a different but equivalent representation. It would also be possible to adapt both algorithms to use SVDs instead of QR factorizations. In this way, similar to Algorithm 5.3, it is possible to truncate the TT-cores during the orthonormalizations.

With the aid of the orthonormalization algorithms, we are now able to compute pseudoinverses of arbitrary tensor unfoldings. The idea is to left-orthonormalize all cores $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(l)}$ and right-orthonormalize all cores $\mathbf{x}^{(l+1)}, \dots, \mathbf{x}^{(d)}$ in order to determine

the pseudoinverse of a matricization of a tensor train $\mathbf{x}$ with respect to the dimensions $(1,\dots,l)$ and $(l+1,\dots,d)$. At the same time, we keep the matrix containing the singular values corresponding to an SVD applied to the core $\mathbf{x}^{(l)}$ intact and regard it as a diagonal matrix between two vectorizations. In this way, we can construct a singular value decomposition of the whole tensor.

The procedure for computing the pseudoinverse is described in Algorithm 5.6 and the steps of the algorithm are also illustrated in Figure 5.3. Again, we make use of the same diagrammatic notation as in [80]. Each TT-core is depicted by a circle with 3 arms indicating the set of free indices and tensor coupling, i.e. index contractions, are represented by joining corresponding arms. In order to visualize orthonormal tensor cores we draw half filled circles.

**Algorithm 5.6** Compute the pseudoinverse of a matricization of a TT-tensor $\mathbf{x}$.

1: Given a tensor $\mathbf{x}$ in TT-format and core number $1 \leq l \leq d-1$, compute pseudoinverse of $\mathbf{x}\Big|_{n_1,\dots,n_l}^{n_{l+1},\dots,n_d}$.
2: Left-orthonormalize $\mathbf{x}^{(1)},\dots,\mathbf{x}^{(l-1)}$ and right-orthonormalize $\mathbf{x}^{(d)},\dots,\mathbf{x}^{(l+1)}$ using Algorithms 5.4 and 5.5.
3: Compute SVD of $\mathbf{x}^{(l)}\Big|_{r_{l-1},n_l}^{r_l}$, i.e. $\mathbf{x}^{(l)}\Big|_{r_{l-1},n_l}^{r_l} = U\Sigma V^T$ with $\Sigma \in \mathbb{R}^{s\times s}$.
4: Define $\mathbf{y} \in \mathbb{R}^{r_{l-1}\times n_l \times s}$ as a reshaped version of $U$ with $\mathbf{y}_{i,j,k} = U_{\overline{i,j},k}$.
5: Define $\mathbf{z} \in \mathbb{R}^{s\times n_{l+1}\times r_{l+1}}$ by $\mathbf{z}\Big|_{s}^{n_{l+1},r_{l+1}} = V^T \cdot \mathbf{x}^{(l+1)}\Big|_{r_l}^{n_{l+1},r_{l+1}}$.
6: Set $\mathbf{x}^{(l)}$ to $\mathbf{y}$, $\mathbf{x}^{(l+1)}$ to $\mathbf{z}$ and $r_l$ to $s$.
7: Define $M = \left(\sum_{k_0=1}^{r_0}\cdots\sum_{k_{l-1}=1}^{r_{l-1}} \mathbf{x}^{(1)}_{k_0,:,k_1}\otimes\ldots\otimes\mathbf{x}^{(l)}_{k_{l-1},:,:}\right)\Big|_{n_1,\dots,n_l}^{r_l}$.
8: Define $N = \left(\sum_{k_{l+1}=1}^{r_{l+1}}\cdots\sum_{k_d=1}^{r_d} \mathbf{x}^{(l+1)}_{:,:,k_{l+1}}\otimes\ldots\otimes\mathbf{x}^{(d)}_{k_{d-1},:,k_d}\right)\Big|_{r_l}^{n_{l+1},\dots,n_d}$.
9: Then $\mathbf{x}\Big|_{n_1,\dots,n_l}^{n_{l+1},\dots,n_d} = M\,\Sigma\,N$ and $\left(\mathbf{x}\Big|_{n_1,\dots,n_l}^{n_{l+1},\dots,n_d}\right)^{+} = N^T\,\Sigma^{-1}\,M^T$.

**Proposition 5.3.7.** *Given a tensor* $\mathbf{x}$ *and core number* $1 \leq l \leq d-1$*, Algorithm 5.6 computes the pseudoinverse with respect to the dimensions* $(1,\dots,l)$ *and* $(l+1,\dots,d)$*.*

*Proof.* Since the left- and right-orthonormalization as well as the application of the SVD in line three of Algorithm 5.6 do not change the tensor $\mathbf{x}$ itself, we can express the matricization of $\mathbf{x}$ with respect to the dimensions $(1,\dots,l)$ and $(l+1,\dots,d)$ as

$$\mathbf{x}\Big|_{n_1,\dots,n_l}^{n_{l+1},\dots,n_d} = M\,\Sigma\,N,$$

with $M$, $\Sigma$, and $N$ as given in Algorithm 5.6. Now, we only have to show that $M^T\cdot M =$

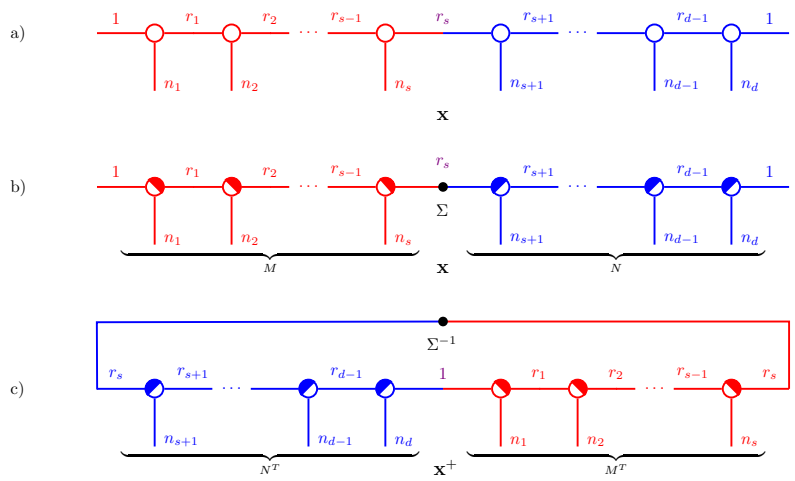



**Figure 5.3:** Illustration of the procedure to compute the pseudoinverse of a tensor in TT-format. a) Initial tensor $\mathbf{x}$. b) Left- and right-orthonormalization of the tensor cores. c) Representation of the pseudoinverse $\mathbf{x}^+$.

$N \cdot N^T = I \in \mathbb{R}^{r_l \times r_l}$. We obtain

$$M^T \cdot M = \left( \left( \sum_{k_0=1}^{r_0} \cdots \sum_{k_{l-1}=1}^{r_{l-1}} \mathbf{x}^{(1)}_{k_0,:,k_1} \otimes \ldots \otimes \mathbf{x}^{(l)}_{k_{l-1},:,:} \right) \Big|^{r_l}_{n_1,\ldots,n_l} \right)^T \cdot \left( \sum_{k'_0=1}^{r_0} \cdots \sum_{k'_{l-1}=1}^{r_{l-1}} \mathbf{x}^{(1)}_{k'_0,:,k'_1} \otimes \ldots \otimes \mathbf{x}^{(l)}_{k'_{l-1},:,:} \right) \Big|^{r_l}_{n_1,\ldots,n_l}.$$

Considering an entry of $M^T \cdot M$ and using Lemma 5.3.5, we then get

$$\begin{aligned} \left(M^T \cdot M\right)_{i,j} &= \left( \left( \sum_{k_0=1}^{r_0} \cdots \sum_{k_{l-1}=1}^{r_{l-1}} \mathbf{x}^{(1)}_{k_0,:,k_1} \otimes \ldots \otimes \mathbf{x}^{(l)}_{k_{l-1},:,i} \right) \Big|_{n_1,\ldots,n_l} \right)^T \\ &\quad \cdot \left( \sum_{k'_0=1}^{r_0} \cdots \sum_{k'_{l-1}=1}^{r_{l-1}} \mathbf{x}^{(1)}_{k'_0,:,k'_1} \otimes \ldots \otimes \mathbf{x}^{(l)}_{k'_{l-1},:,j} \right) \Big|_{n_1,\ldots,n_l} \\ &= \sum_{k_0=1}^{r_0} \cdots \sum_{k_{l-1}=1}^{r_{l-1}} \sum_{k'_0=1}^{r_0} \cdots \sum_{k'_{l-1}=1}^{r_{l-1}} \prod_{\mu=1}^{l} \left( \mathbf{x}^{(\mu)}_{k_{\mu-1},:,k_\mu} \right)^T \cdot \mathbf{x}^{(\mu)}_{k'_{\mu-1},:,k'_\mu}, \end{aligned}$$

with $k_l = i$ and $k'_l = j$. Since $\mathbf{x}^{(1)}$ is left-orthonormal and $r_0 = 1$, we obtain $\left(\mathbf{x}^{(1)}_{1,:,k_1}\right)^T \cdot \mathbf{x}^{(1)}_{1,:,k'_1} = \delta_{k_1,k'_1}$. This implies that $\left(M^T \cdot M\right)_{i,j}$ is only nonzero if $k_1 = k'_1$. Now, we include

the next core. This yields

$$\sum_{k_1=1}^{r_1}\sum_{k_1'=1}^{r_1}\delta_{k_1,k_1'}\cdot\left(\mathbf{x}^{(2)}_{k_1,:,k_2}\right)^T\cdot\mathbf{x}^{(2)}_{k_1',:,k_2'}$$
$$=\sum_{k_1=1}^{r_1}\left(\mathbf{x}^{(2)}_{k_1,:,k_2}\right)^T\cdot\mathbf{x}^{(2)}_{k_1,:,k_2'}=\left(\mathbf{x}^{(2)}_{:,:,k_2}\Big|_{r_1,n_1}\right)^T\cdot\mathbf{x}^{(2)}_{:,:,k_2'}\Big|_{r_1,n_1}=\delta_{k_2,k_2'}$$

since $\mathbf{x}^{(2)}$ is also left-orthonormal. Successively, it follows that, in order for $\left(M^T\cdot M\right)_{i,j}$ to be nonzero, it must hold that $k_\mu=k_\mu'$ for $\mu=2,\dots,l-1$. Thus, we obtain

$$\left(M^T\cdot M\right)_{i,j}=\sum_{k_{l-1}=1}^{r_{l-1}}\left(\mathbf{x}^{(l)}_{k_{l-1},:,i}\right)^T\cdot\mathbf{x}^{(l)}_{k_{l-1},:,j}=\left(\mathbf{x}^{(l)}_{:,:,i}\Big|_{r_{l-1},n_l}\right)^T\cdot\mathbf{x}^{(l)}_{:,:,j}\Big|_{r_{l-1},n_l}.$$

Note that due to the construction (see lines 3 & 4 of Algorithm 5.6) $\mathbf{x}^{(l)}$ is also left-orthonormal and therefore

$$M^T\cdot M=I\in\mathbb{R}^{r_l\times r_l}.$$

Analogously, using the right-orthonormality of $\mathbf{x}^{(l+1)},\dots,\mathbf{x}^{(d)}$, it can then be shown that $N\cdot N^T=I$. It follows that the pseudoinverse calculated by Algorithm 5.6 satisfies the necessary conditions, e.g.

$$\mathbf{x}\Big|^{n_{l+1},\dots,n_d}_{n_1,\dots,n_l}\cdot\left(\mathbf{x}\Big|^{n_{l+1},\dots,n_d}_{n_1,\dots,n_l}\right)^+\cdot\mathbf{x}\Big|^{n_{l+1},\dots,n_d}_{n_1,\dots,n_l}$$
$$=M\,\Sigma\,N\cdot N^T\,\Sigma^{-1}\,M^T\cdot M\,\Sigma\,N=M\,\Sigma\,N=\mathbf{x}\Big|^{n_{l+1},\dots,n_d}_{n_1,\dots,n_l}.\qquad\square$$

**Remark 5.3.8.** Algorithm 5.6 depicts just one possible way to compute the pseudoinverse of a given tensor. If the cores of the tensor train $\mathbf{x}$ are already left- or right-orthonormal, respectively, e.g. the decomposition of $\mathbf{x}$ was computed by Algorithm 5.3, we can adapt Algorithm 5.6. For instance, if all TT-cores (except the last one) are already left-orthonormal, we skip the application of Algorithm 5.4 and only right-orthonormalize the cores $\mathbf{x}^{(d)},\dots,\mathbf{x}^{(l+2)}$. Then the SVD can be applied to a matricization of $\mathbf{x}^{(l+1)}$ such that the matrix $V^T$ represents the updated version of $\mathbf{x}^{(l+1)}$ and $U$ is multiplied to the core $\mathbf{x}^{(l)}$ from the right.

Note that we do not need to compute $M$ and $N$ explicitly. Instead, we only execute the lines 1 to 6 of Algorithm 5.6 and then store the modified cores and the matrix $\Sigma$ containing the singular values. Thus, again with a slight abuse of the notation, we only store

$$\mathbf{x}^+=\sum_{k_l=1}^{r_l}\sigma_{k_l}^{-1}\cdot\left(\sum_{k_{l+1}=1}^{r_{l+1}}\cdots\sum_{k_{d-1}=1}^{r_{d-1}}\mathbf{x}^{(l+1)}_{k_l,:,k_{l+1}}\otimes\ldots\otimes\mathbf{x}^{(d)}_{k_{d-1},:,1}\right)$$
$$\cdot\left(\sum_{k_1=1}^{r_1}\cdots\sum_{k_{l-1}=1}^{r_{l-1}}\mathbf{x}^{(1)}_{1,:,k_1}\otimes\ldots\otimes\mathbf{x}^{(l)}_{k_{l-1},:,k_l}\right),$$

which can be either regarded as the sum of $r_l$ tensor trains scaled by $\sigma_1^{-1}, \dots, \sigma_{r_l}^{-1}$ or as a cyclic tensor train as depicted in Figure 5.3. For a detailed description of the cyclic TT-format – also called cyclic matrix product states – we refer to [77]. Note that here again the cores $1, \dots, l$ and $l+1, \dots, d$ are simply swapped as in Example 5.3.6.

### 5.3.3 Computation of DMD modes and eigenvalues in TT-format

Let us now assume that the $m$ snapshots are $d$-dimensional arrays – given for example by low-rank tensor representations – of the form

$$\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^{n_1 \times \cdots \times n_d}, \tag{5.14}$$

where $\mathbf{y}_i = F(\mathbf{x}_i)$. These snapshots can be stored in the $(d+1)$-dimensional tensor trains $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_d \times m}$, such that

$$\mathbf{X}_{:,\dots,:,i} = \mathbf{x}_i \quad \text{and} \quad \mathbf{Y}_{:,\dots,:,i} = \mathbf{y}_i, \tag{5.15}$$

for $i = 1, \dots, m$. Let $r_0, \dots, r_{d+1}$ be the TT-ranks of $\mathbf{X}$ and $s_0, \dots, s_{d+1}$ the TT-ranks of $\mathbf{Y}$. Now, let $X, Y \in \mathbb{R}^{n_1 \cdot \dots \cdot n_d \times m}$ be the specific matricizations of $\mathbf{X}$ and $\mathbf{Y}$, where we contract the dimensions $n_1, \dots, n_d$ such that every column of $X$ and $Y$, respectively, is the vectorization of the corresponding snapshot. We assume again that there is a linear relationship between the pairs of data vectors, i.e.

$$Y = AX, \tag{5.16}$$

with $A \in \mathbb{R}^{n_1 \cdot \dots \cdot n_d \times n_1 \cdot \dots \cdot n_d}$. We already stated in (5.3) that the linear operator $A$ can be computed by $Y \cdot X^+$. The pseudoinverse $X^+$ can be expressed – after applying Algorithm 5.6 – as

$$X^+ = \left( \mathbf{X} \Big|_{n_1,\dots,n_d}^{m} \right)^+ = N^T \Sigma^{-1} M^T. \tag{5.17}$$

Using similar matricizations, we can also represent the tensor unfolding $Y$ as a matrix product, i.e.

$$\begin{aligned} Y &= \mathbf{Y} \Big|_{n_1,\dots,n_d}^{m} \\ &= \left( \sum_{l_0=1}^{s_0} \cdots \sum_{l_{d-1}=1}^{s_{d-1}} \mathbf{Y}^{(1)}_{l_0,:,l_1} \otimes \dots \otimes \mathbf{Y}^{(d)}_{l_{d-1},:,:} \right) \Big|_{n_1,\dots,n_d}^{s_d} \cdot \mathbf{Y}^{(d+1)} \Big|_{s_d}^{m} = P\,Q. \end{aligned} \tag{5.18}$$

Note that we do not require any special properties of the tensor cores of $\mathbf{Y}$. Left- and right-orthonormality must only hold for the TT-cores of $\mathbf{X}$. Combining the representations of $X^+$ and $Y$, we can express the matrix $A$ as

$$A = Y \cdot X^+ = P\,Q \cdot N^T \Sigma^{-1} M^T. \tag{5.19}$$

As explained in Section 5.2, there are different algorithms to compute the eigenvalues and eigenvectors of $A$. Instead of computing $A$ explicitly, we are interested in the reduced matrix $\tilde{A} \in \mathbb{R}^{r_d \times r_d}$ as mentioned in Algorithm 5.1. Rewriting the expression for $\tilde{A}$ using the decompositions given in (5.17) and (5.18), this results in

$$\tilde{A} = M^T \cdot P\, Q \cdot N^T\, \Sigma^{-1}. \tag{5.20}$$

In order to compute $\tilde{A}$, we do not have to compute the matrices $M^T$ and $P$ explicitly. We bypass this computational cost by splitting (5.20) into different parts. First, consider $M^T \cdot P$, any entry is given by

$$\begin{aligned}\left(M^T \cdot P\right)_{i,j} = &\left(\left(\sum_{k_0=1}^{r_0} \cdots \sum_{k_{d-1}=1}^{r_{d-1}} \mathbf{X}^{(1)}_{k_0,:,k_1} \otimes \ldots \otimes \mathbf{X}^{(d)}_{k_{d-1},:,i}\right)\Big|_{n_1,\ldots,n_d}\right)^T \\ &\cdot \left(\sum_{l_0=1}^{s_0} \cdots \sum_{l_{d-1}=1}^{s_{d-1}} \mathbf{Y}^{(1)}_{l_0,:,l_1} \otimes \ldots \otimes \mathbf{Y}^{(d)}_{l_{d-1},:,j}\right)\Big|_{n_1,\ldots,n_d}.\end{aligned}$$

It follows from the linearity of matricizations and Lemma 5.3.5 that

$$\left(M^T \cdot P\right)_{i,j} = \sum_{k_0=1}^{r_0} \cdots \sum_{k_{d-1}=1}^{r_{d-1}} \sum_{l_0=1}^{s_0} \cdots \sum_{l_{d-1}=1}^{s_{d-1}} \left(\mathbf{X}^{(1)}_{k_0,:,k_1}\right)^T \mathbf{Y}^{(1)}_{l_0,:,l_1} \cdot \ldots \cdot \left(\mathbf{X}^{(d)}_{k_{d-1},:,i}\right)^T \mathbf{Y}^{(d)}_{l_{d-1},:,j}.$$

Thus, by defining the matrices $\Theta_\mu \in \mathbb{R}^{r_{\mu-1}\cdot s_{\mu-1} \times r_\mu \cdot s_\mu}$ as

$$(\Theta_\mu)_{\overline{k_{\mu-1},l_{\mu-1}},\overline{k_\mu,l_\mu}} = \left(\mathbf{X}^{(\mu)}_{k_{\mu-1},:,k_\mu}\right)^T \cdot \mathbf{Y}^{(\mu)}_{l_{\mu-1},:,l_\mu}, \tag{5.21}$$

for $\mu = 1, \ldots, d$, we can write any entry of $M^T \cdot P$ as

$$\left(M^T \cdot P\right)^T_{i,j} = \Theta_1 \cdot \Theta_2 \cdot \ldots \cdot \Theta_{d-1} \cdot (\Theta_d)_{:,\overline{i,j}}. \tag{5.22}$$

In this way, we can compute $M^T \cdot P$ without leaving the TT-format, we only have to reshape certain contractions of the TT-cores as depicted in (5.21) and (5.22). This computation can be implemented efficiently using Algorithm 4 from [142]. The result is then a low-dimensional matrix with $r_d$ rows and $s_d$ columns, assuming that the TT-ranks of $\mathbf{X}$ and $\mathbf{Y}$ are small compared to the whole state space of these tensors. Indeed, the tensor ranks $r_d$ and $s_d$ are both bounded by the number of snapshots $m$ due to the right-orthonormalization of the last TT-cores. For the second term of (5.20), $Q \cdot N^T$, we simply obtain

$$Q \cdot N^T = \mathbf{Y}^{(d+1)}\Big|_{s_d}^{m} \cdot \left(\mathbf{X}^{(d+1)}\Big|_{r_d}^{m}\right)^T.$$

Subsequently, we only have to multiply the three low-dimensional matrices $\left(M^T \cdot P\right)$, $\left(Q \cdot N^T\right)$ and $\Sigma^{-1}$. The latter is just a diagonal matrix containing the reciprocals of the

singular values occurring in $\Sigma$. Overall, we do not need to convert any tensor products of cores of $\mathbf{X}$ or $\mathbf{Y}$, respectively, into full tensors during our calculations. The results are naturally low-dimensional matrices and the reduced matrix $\tilde{A}$ can finally be used to compute the eigenvalues of the high-dimensional matrix $A$ since both have the same spectrum.

In order to compute the corresponding DMD modes of $A$, consider again Algorithm 5.1 and Algorithm 5.2. If $\lambda_1, \dots, \lambda_\nu$ are the eigenvalues of $\tilde{A}$ corresponding to the eigenvectors $w_1, \dots, w_\nu \in \mathbb{R}^{r_d}$, then the DMD modes $\varphi_1, \dots, \varphi_\nu \in \mathbb{R}^{n_1 \cdot \dots \cdot n_d}$ of $A$ according to the standard algorithm are given by

$$\varphi_\mu = M \cdot w_\mu, \tag{5.23}$$

for $\mu = 1, \dots, \nu$. At this point, we again benefit from using the TT-representations of $M$. What (5.23) tells us is actually just the replacement of the last TT-core. This can be seen by defining $W \in \mathbb{R}^{r_d \times \nu}$ as

$$W = \begin{bmatrix} w_1 & w_2 & \cdots & w_\nu \end{bmatrix} \tag{5.24}$$

and writing

$$M \cdot W = \left( \sum_{k_0=1}^{r_0} \cdots \sum_{k_{d-1}=1}^{r_{d-1}} \mathbf{X}^{(1)}_{k_0,:,k_1} \otimes \ldots \otimes \mathbf{X}^{(d)}_{k_{d-1},:,:} \right) \Bigg|_{n_1,\dots,n_d}^{r_d} \cdot W. \tag{5.25}$$

As a result, we can also express the DMD modes in a TT-representation, i.e.

$$\Phi = \sum_{k_0=1}^{r_0} \cdots \sum_{k_d=1}^{r_d} \mathbf{X}^{(1)}_{k_0,:,k_1} \otimes \ldots \otimes \mathbf{X}^{(d)}_{k_{d-1},:,k_d} \otimes W_{k_d,:}, \tag{5.26}$$

with $\Phi_{:,\dots,:,\mu} \Big|_{n_1,\dots,n_d} = \varphi_\mu$ for $\mu = 1, \dots, \nu$. Considering the exact DMD algorithm, the DMD modes are given by

$$\varphi_\mu = \frac{1}{\lambda} \cdot P\, Q \cdot N^T\, \Sigma^{-1} \cdot w_\mu, \tag{5.27}$$

for $\mu = 1, \dots, \nu$. The tensor train $\Phi$ representing all DMD modes is then given by

$$\Phi = \sum_{k_0=1}^{s_0} \cdots \sum_{k_d=1}^{s_d} \mathbf{Y}^{(1)}_{k_0,:,k_1} \otimes \ldots \otimes \mathbf{Y}^{(d)}_{k_{d-1},:,k_d} \otimes \underbrace{\left(Q \cdot N^T\, \Sigma^{-1} \cdot W \cdot \Lambda^{-1}\right)}_{\in \mathbb{R}^{s_d \times \nu}} {}_{k_d,:}, \tag{5.28}$$

again with $\Phi_{:,\dots,:,\mu} \Big|_{n_1,\dots,n_d} = \varphi_\mu$ for $\mu = 1, \dots, \nu$ and

$$\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_\nu \end{pmatrix}. \tag{5.29}$$

Summing up, we can express the DMD modes using a previously given tensor train, modifying just the last core. For the standard DMD algorithm, we can express $\Phi$ using the first $d$ cores of $\mathbf{X}$ and replacing the core $\mathbf{X}^{(d+1)}$ by $W$. As for the exact DMD algorithm, $\Phi$ is represented by the first $d$ cores of $\mathbf{Y}$, replacing $\mathbf{Y}^{(d+1)}$ by $Q\,N^T\,\Sigma^{-1}\,W\,\Lambda^{-1}$. In both cases, we benefit from not leaving the TT-representations of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

## 5.4 Numerical results

The following examples are mainly for illustration purposes, we will not describe the underlying mathematical models in full detail since the governing equations are not relevant here. Instead of analyzing simulation data, we could also process experimental measurement data. The goal is to detect the dominant dynamics of a dynamical system given only data. The first example, a simulation of two merging vortices, is two-dimensional and has been created with the MULTIMOD toolbox [158]. The second example is a three-dimensional simulation of the flow around a blunt body governed by the incompressible Navier–Stokes equations and has a significantly higher number of degrees of freedom.

The DMD experiments using the TT-format were performed on a Linux machine with 128 GB RAM and an Intel Xeon processor with a clock speed of 3 GHz and 8 cores. The algorithms were implemented in MATLAB R2015a using a compound of cell arrays and multidimensional matrices for tensors in the TT-format.

**Two merging vortices.** The first example shows two merging vortices. Here, the domain is discretized using an $n \times n$ grid with $n = 200, 400, \ldots, 1400$. We generated data for 447 equidistant time steps, thus $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times n \times 446}$. Intermediate solutions are shown in Figure 5.4a, the corresponding DMD modes in Figure 5.4b. We computed tensor representations of $X$ and $Y$ using Algorithm 5.3 in order to compare the results for different thresholds $\varepsilon \geq 0$. Nevertheless, we assume that the data matrices $\mathbf{X}, \mathbf{Y}$ are already given in the TT-format for the tensor-based formulation (the runtimes of Algorithm 5.3 are thus not included in the overall runtimes), i.e. the partial differential equation is directly solved using tensor representations. For example, that could mean that the numerical solution of the partial differential equation is obtained by applying an appropriate time-stepping scheme combined with the Alternating Linear Scheme (ALS) [80]. In this case, we can execute the half-sweeps of the ALS in a way that the resulting tensor is already left-orthonormal. Thus, all relevant calculations to compute the DMD modes are included in the runtimes presented in Figure 5.5. We compare the exact DMD implementations, i.e. Algorithm 5.2 and its tensor-based counterpart.

The DMD modes shown in Figure 5.4b are visually indistinguishable for the different thresholds $\varepsilon$ and the resulting tensor ranks. The influence of the low-rank approximation on the numerical errors is shown in Table 5.1, which contains the relative errors defined by $e_\lambda = |\lambda - \tilde{\lambda}| / |\lambda|$ and $e_\varphi = \|\varphi - \tilde{\varphi}\|_F / \|\varphi\|_F$, where $\lambda$ and $\varphi$ are the DMD eigenvalue and mode for $\varepsilon = 0$ and $\tilde{\lambda}$ and $\tilde{\varphi}$ the corresponding approximations for a given $\varepsilon > 0$. Here, we normalized each mode in such a way that the largest absolute value is 1. For

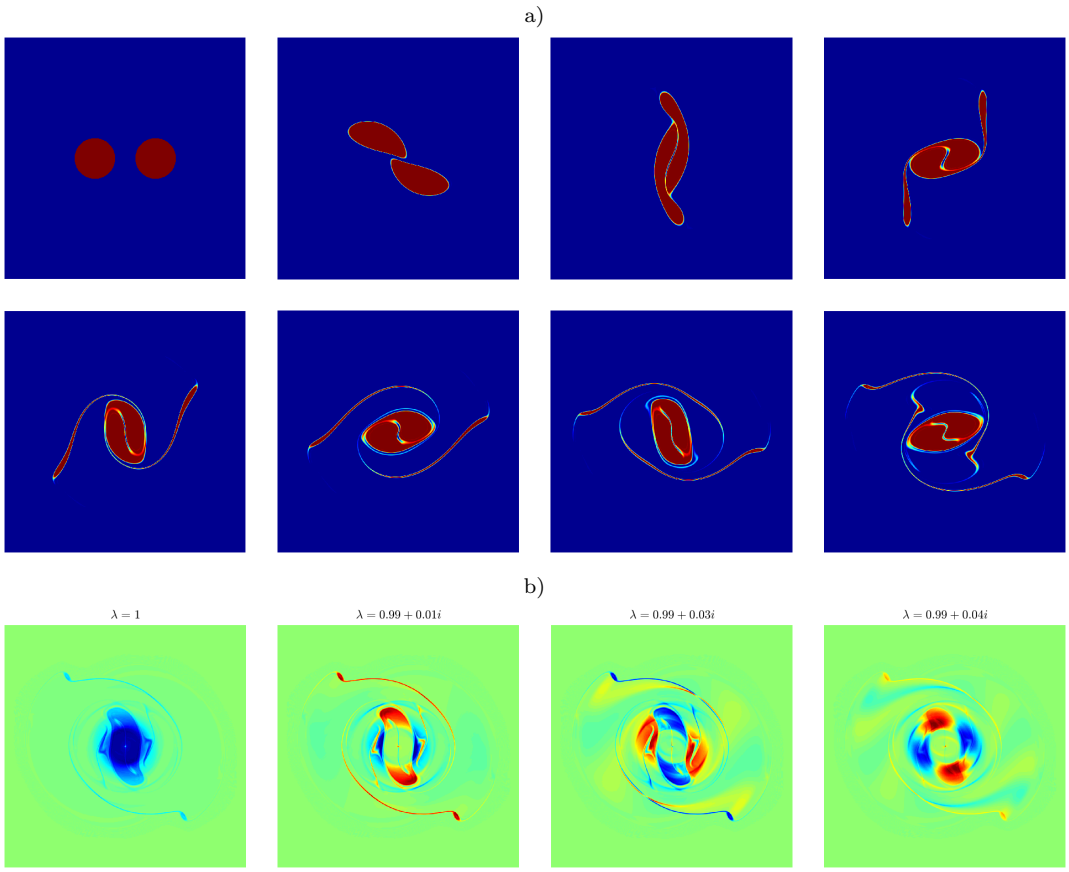



**Figure 5.4:** a) Simulation of two merging vortices. b) DMD modes corresponding to eigenvalues close to 1.

$n = 1400$, the speedup is approximately 7 for $\varepsilon = 0$ and 16 for $\varepsilon = 1\text{e}{-10}$. The ranks of the corresponding tensor trains $\mathbf{X}$ are $r = [1, 1400, 446, 1]$ and $r = [1, 732, 446, 1]$, respectively. That is, approximately half of the singular values between the first and second core are less than $\varepsilon = 1\text{e}{-10}$. This illustrates that a given tensor representation of the data can be exploited to efficiently compute the DMD modes and eigenvalues without converting the data set to the full format since the tensor-train format already contains information about the required pseudoinverse. Furthermore, the results show that the lower the rank of the tensor approximation – which depends on the parameter $\varepsilon$ – the higher the speedup.

**Flow around blunt body.** As a second example, we consider the flow around a blunt body governed by the three-dimensional incompressible Navier-Stokes equations. Similar experiments have been described in [159]. Here, the domain $\Omega \subset \mathbb{R}^3$ has a size of $L = (25, 15, 10)$. A conical object is placed inside the domain with the center axis at $(x_1, x_2) = (5, 7.5)$ and a diameter of $D_1 = 0.8$ at the boundaries and $D_2 = 1.6$ in the

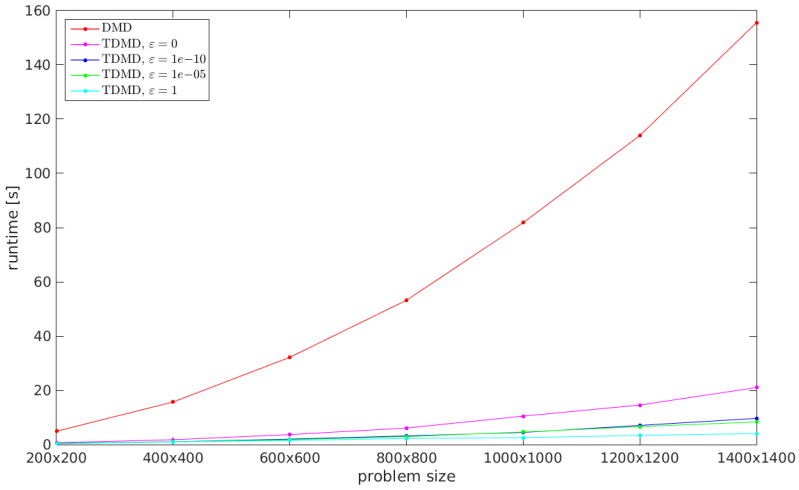



**Figure 5.5:** Comparison of the runtimes of DMD and tensor-based DMD applied to the vortex merging example for different values of $\varepsilon$ and different problem sizes.

**Table 5.1:** Influence of the truncation on the accuracy of the leading DMD modes for the $1400 \times 1400$ grid discretization of the vortex merging example. The entries show the relative errors $e_\lambda$ and $e_\varphi$ for different values of $\varepsilon$.

| | Mode 1 | | Mode 2 | | Mode 3 | | Mode 4 | |
|---|---|---|---|---|---|---|---|---|
| | $e_\lambda$ | $e_\varphi$ | $e_\lambda$ | $e_\varphi$ | $e_\lambda$ | $e_\varphi$ | $e_\lambda$ | $e_\varphi$ |
| $\varepsilon = 1e{-}10$ | 5.55e−15 | 5.92e−14 | 5.31e−15 | 3.32e−13 | 9.75e−15 | 3.50e−13 | 1.37e−14 | 3.28e−13 |
| $\varepsilon = 1e{-}05$ | 1.47e−14 | 2.23e−10 | 5.09e−15 | 4.27e−10 | 4.93e−15 | 9.73e−10 | 1.39e−14 | 7.77e−10 |
| $\varepsilon = 1$ | 6.10e−08 | 5.22e−04 | 2.90e−06 | 3.16e−03 | 3.02e−06 | 2.62e−03 | 4.00e−06 | 7.86e−04 |

middle of the channel (cf. Figure 5.6). We set the Reynolds number $Re = \overline{U}D/\nu = 240$, based on the inflow velocity $\overline{U} = (1, 0, 0)$, the kinematic viscosity $\nu = 5 \cdot 10^{-3}$, and the average cone diameter $D = 1.2$. Moreover, we apply periodic boundary conditions in the $x_2$ as well as the $x_3$ direction. The domain is discretized by a rectangular grid with approximately $10^6$ degrees of freedom and the computations are performed using OpenFOAM [86]. Before applying the tensor-based DMD algorithm, 1001 snapshots are interpolated on an equidistant, rectangular grid of dimension $150 \times 85 \times 80$, where the values for all grid points inside the object are set to zero.

For the DMD analysis, we consider the velocity magnitude $\mathbf{U} = \sqrt{U_1^2 + U_2^2 + U_3^2}$, thus $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{150 \times 85 \times 80 \times 1000}$. The resulting ranks, runtimes, and relative errors for different values of $\varepsilon$ are shown in Table 5.2. By increasing $\varepsilon$, the initially high ranks can be reduced without a huge loss of accuracy. Here, we compared only the first two eigenmodes shown in Figure 5.7 with the DMD modes for $\varepsilon = 0$. The runtime of conventional DMD for this problem is approximately 125 s. The results show that the efficiency of DMD can be improved significantly using tensor decompositions, provided that the ranks are

**Table 5.2:** Influence of the truncation on the ranks, runtimes, and accuracy of the leading DMD modes for the blunt body problem.

| | Ranks | Runtime | Mode 1 | | Mode 2 | |
|---|---|---|---|---|---|---|
| | | | $e_\lambda$ | $e_\varphi$ | $e_\lambda$ | $e_\varphi$ |
| $\varepsilon = 0$ | [1, 150, 6083, 1000, 1] | 134 s | 0 | 0 | 0 | 0 |
| $\varepsilon = 0.01$ | [1, 150, 4708, 1000, 1] | 102 s | 6.29e−07 | 6.33e−04 | 1.53e−07 | 2.88e−04 |
| $\varepsilon = 0.05$ | [1, 150, 3649, 641, 1] | 52 s | 1.11e−04 | 3.64e−02 | 3.05e−05 | 2.37e−02 |
| $\varepsilon = 0.1$ | [1, 148, 3003, 527, 1] | 35 s | 1.38e−04 | 6.92e−02 | 1.26e−04 | 3.65e−02 |
| $\varepsilon = 0.5$ | [1, 135, 1624, 343, 1] | 14 s | 2.62e−04 | 8.69e−02 | 9.61e−05 | 5.55e−02 |
| $\varepsilon = 1$ | [1, 130, 1199, 278, 1] | 8 s | 4.34e−04 | 1.72e−01 | 1.80e−04 | 1.06e−01 |

reasonably low. The aim is thus to directly compute low-rank solutions for such problems.

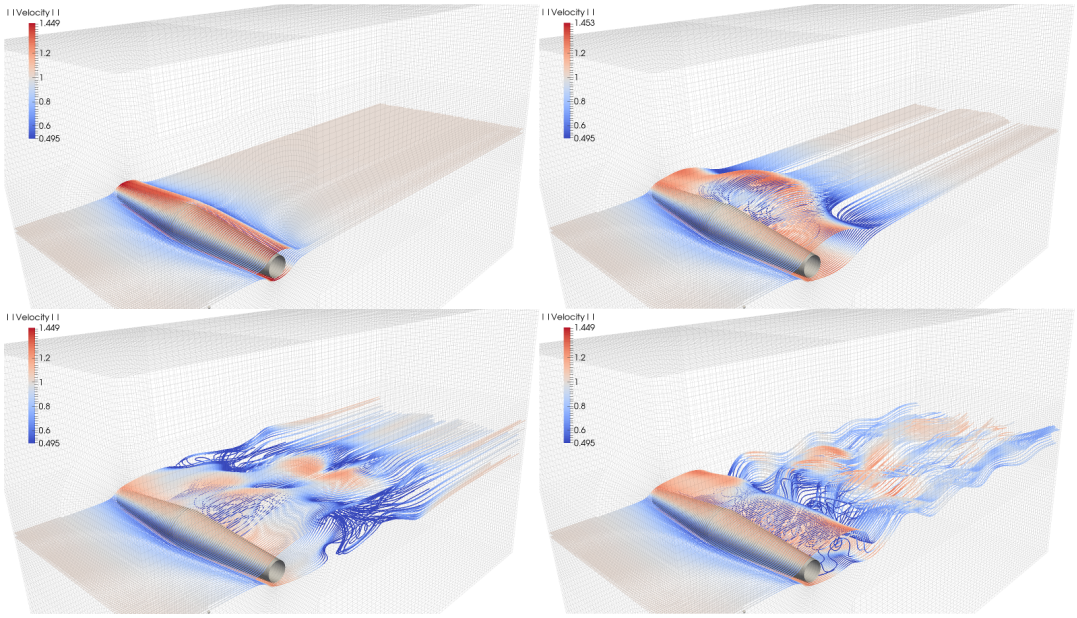



**Figure 5.6:** Simulation of the flow around a blunt body. The flow is visualized by streamlines which are inserted at the inflow and slightly above and below the cone axis, respectively. The streamlines are colored according to the velocity magnitude.

Figure 5.6 shows streamlines of the velocity field at different time steps. Similar to the two-dimensional von Kármán vortex street (Figure 5.1), vortices separate alternatingly from the lower and the upper side of the blunt body. However, due to the conical shape of the object, the flow pattern is much more complex than in the two-dimensional case. In this case, storing the full matrix $A$ would require more than 7.5 TB. The corresponding DMD modes, sorted by their respective frequency, are depicted in Figure 5.7, where we visualize the flow patterns by iso-surfaces of the velocity magnitude. In accordance with [159], we calculate the frequency of the modes by $\omega = \Im(\log(\lambda))/\Delta t$, where $\Delta t$ is the time step between the snapshots. We observe that larger structures correspond to

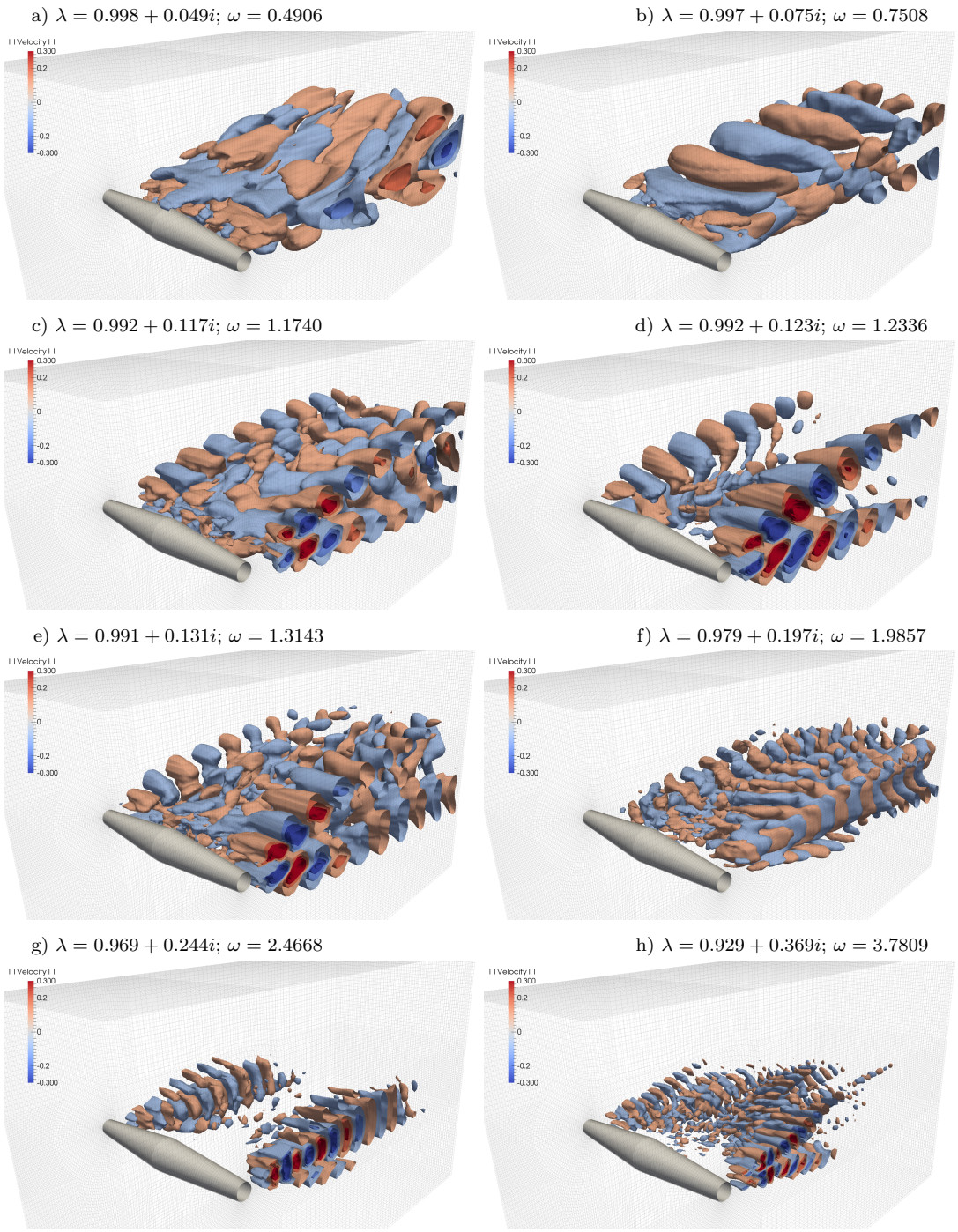



**Figure 5.7:** DMD modes for the flow around a cone corresponding to eigenvalues close to 1. The modes are visualized by iso-surfaces of the velocity magnitude.

lower frequencies, i.e. they indicate large but slowly rotating vortices, whereas the smaller structures possess higher frequencies. Moreover, we can see that large structures originate from the outward pointing kink in the middle of the domain and the smaller structures from the inward pointing kink at the periodic boundary in $x_2$-direction.

## 5.5 Conclusion

We showed that the TT-format – which is based on successive SVDs – implicitly already contains information about the pseudoinverse of certain tensor unfoldings. The goal is to gain insight into the characteristic properties of tensors and to develop tensor-based algorithms for solving, for instance, systems of linear equations or eigenvalue problems that directly exploit the inherent properties of the TT-decomposition instead of reformulating the problem as an equivalent optimization problem. One application which requires the computation of a pseudoinverse of such a tensor unfolding is DMD. If the data to be analyzed is already given in TT-format, our algorithm efficiently computes the DMD modes and eigenvalues directly on the low-rank representations of the data matrices $\mathbf{X}$ and $\mathbf{Y}$.

Analogously, variants of DMD such as sparsity-promoting DMD or kernel-based DMD could be reformulated as well. In the same way, the algorithms presented within this paper could be used to extend EDMD to compute the Koopman modes directly in the TT-format, provided that the simulation data is generated using low-rank tensor approximations. The only difference is that we then have to compute the matrix $A = \Psi_Y \Psi_X^+$, where $\Psi_X$ and $\Psi_Y$ are nonlinear transformations of the original data matrices $X$ and $Y$. A tensor-based method to compute the eigenfunctions of the Koopman operator has been proposed in [95]. The computations rely on the construction of a generalized eigenvalue problem of the form $\boldsymbol{\xi}\mathbf{A} = \lambda\boldsymbol{\xi}\mathbf{G}$, which is solved using simple power iteration schemes. Furthermore, our approach to compute pseudoinverses of tensors may be extended to the hierarchical Tucker format.

So far, we considered several two-dimensional problems and one three-dimensional problem to illustrate how tensor-based data-driven methods might help mitigate the curse of dimensionality and enable the analysis of more complex dynamical systems. Future work includes applying tensor-based DMD to higher-dimensional problems to analyze scalability and efficiency of the proposed methods as well as studying fluid flow applications with a larger number of degrees of freedom.

## Acknowledgements

This research has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114 as well as by the Berlin Mathematical School and the Einstein Center for Mathematics. The OpenFOAM calculations were performed on resources provided by the Paderborn Center for Parallel Computing ($PC^2$). Moreover, we would like to thank the reviewers for their helpful comments and suggestions for improvement.

# 6
# Conclusion and open problems

In this thesis, we presented different data-driven methods for the global analysis of complex dynamical systems. The methods were mainly illustrated with the aid of simple drift-diffusion processes, molecular dynamics applications, and fluid dynamics problems. However, the presented methods can be applied to any dynamical system, given simulation or measurement data. The main contributions of the respective publications are as follows:

**On the numerical approximation of the Perron–Frobenius and Koopman operator:**

- First, we gave an overview of different numerical methods developed for the approximation of transfer operators.
- We then showed that recently proposed data-driven methods for the approximation of the Koopman operator such as EDMD can be rewritten to approximate the Perron–Frobenius operator and its eigenfunctions.
- We applied the derived approach to different stochastic differential equations and a well-known molecular dynamics problem to illustrate the efficiency of the proposed methods.

**Data-driven model reduction and transfer operator approximation:**

- Several different data-driven methods for model reduction and the approximation of transfer operators were developed independently by the dynamical systems, fluid dynamics, and molecular dynamics communities. While the methods developed by the molecular dynamics community such as TICA and VAC were mainly developed for reversible dynamical systems, whereas DMD and EDMD proposed by the fluid dynamics and dynamical systems communities can be applied to more general dynamical systems, we showed that the resulting numerical methods share many similarities, although the derivations and intended applications are rather different.
- We unified the theory and thereby enabled the generalization of extensions and improvements developed for one particular method to other methods.
- We demonstrated how data-driven methods can be combined by first reducing the state space to the essential coordinates and then approximating the eigenfunctions of the reduced dynamical system.

**Towards tensor-based methods for the numerical approximation of the Perron–Frobenius and Koopman operator:**

- One drawback of most of the methods for the approximation of transfer operators is that the number of basis functions grows exponentially with the dimension of the state space. This is often referred to as the *curse of dimensionality*. Different tensor formats have been developed over the last years aiming at mitigating the curse of dimensionality by computing low-rank approximations of high-dimensional data arrays. We showed how Ulam's method and also EDMD can be rewritten in terms of tensors.

- We applied simple power iteration methods to the resulting (generalized) eigenvalue problems. It was shown in [71] that also the *alternating linear scheme (ALS)* can be applied to compute the dominant eigenvalues and eigenvectors.

- We demonstrated that the applicability of tensor approaches depends both on the ordering of variables and the alignment of the dynamics with the chosen coordinate system.

**Tensor-based dynamic mode decomposition:**

- We showed that the pseudoinverse of certain tensor unfoldings – required, for instance, when we want to apply DMD to compressed data represented in the TT format – can be computed efficiently directly in the TT format by exploiting characteristic properties of the TT decomposition of multi-dimensional arrays.

- We reformulated standard DMD and exact DMD for low-rank approximations of high-dimensional data matrices.

- The efficiency of the approach was illustrated with the aid of two-dimensional and three-dimensional fluid dynamics problems.

There are several open questions pertaining to the proposed methods. We approximate an infinite-dimensional operator by projecting it onto a space spanned by a finite set of basis functions. The approximation error depends strongly on the chosen basis. Preselecting a suitable set of basis functions that is able to represent the dynamics accurately is still an open problem. Future work could be to apply dictionary-learning methods in order to adaptively *learn* suitable basis functions. Recently, machine learning techniques using deep neural networks were proposed to learn basis functions and transfer operators from data [111, 144, 118].

Another approach would be to use kernel-based methods [196, 173, 94]. A Gaussian kernel, for instance, corresponds to an implicitly infinite-dimensional set of basis functions. The drawback then, however, is that the size of the resulting matrices is given by the number of snapshots, which must thus be small enough to be able to store the matrices and to solve the eigenvalue problems numerically. Furthermore, the convergence properties of these methods, in particular for kernels with infinite-dimensional feature spaces, is – to our

knowledge – not well understood yet. Kernel-based methods could again be augmented by using low-rank tensor representations.

The tensor-based methods described in Chapter 4 and Chapter 5 and the reduction to the essential dynamics based on transition manifolds presented in Appendix A can be seen as two different ways to mitigate the curse of dimensionality. The former compresses the multidimensional generalizations of the vectors and matrices representing eigenfunctions or finite-dimensional approximations of transfer operators using low-rank decompositions, the latter aims at finding a reduced transfer operator that retains the dominant spectrum of the full transfer operator by averaging along level-sets of reaction coordinates. In order to be able to find a low-rank representation of a transfer operator, the dynamics of the system should be aligned with the chosen coordinate system. Rotations or nonlinear transformations of the dynamics, for instance, lead to a strong coupling between variables and thus higher ranks. One possibility to overcome this would be to first project the dynamics onto the TICA coordinates, which represent the slowest relaxation processes of the system. Another drawback of the TT format is that it depends strongly on the ordering of variables. It was shown in [72] that the TT format is suited in particular for systems with a nearest neighbor coupling structure. A molecular dynamics problem with such a structure is deca-alanine, which was analyzed in [138] (see also Chapter 1 and Chapter 3). Ordering the unknowns in such a way that variables are connected mainly to adjacent variables and only weakly to other variables will typically result in lower ranks. Such an ordering, however, is in general unknown. For systems with arbitrary interconnectivity, tensor formats that are able to represent more complex coupling structures might be required. This is still an open problem. Also the extension of ALS to generalized eigenvalue problems, which is required for tensor-based EDMD, could be potential future work.

Furthermore, as briefly mentioned in the introduction, data-driven methods can be used to directly compute the governing equations of a dynamical system, see, e.g., [20, 121]. These methods are closely related to the methods presented above. So far, the proposed approaches have been applied mainly to ordinary differential equations and partial differential equations with a certain structure. Future work would be to extend and apply these methods to stochastic differential equations and, in particular, molecular dynamics problems. Moreover, it would be beneficial to derive error bounds and convergence results. Another application of methods for identifying governing equations is to detect the connectivity structure of a system from data, i.e., whether variables are strongly coupled or not, see [90]. This could be of particular interest for many biological systems.

# Bibliography


[1] J. Ballani and L. Grasedyck. A projection method to solve linear systems in tensor format. *Numerical linear algebra with applications*, 20(1):27–43, 2013.

[2] C. Bandle. *Isoperimetric inequalities and applications*. Monographs and studies in mathematics. Pitman, 1980.

[3] J. R. Baxter and J. S. Rosenthal. Rates of convergence for everywhere-positive Markov chains. *Statistics & probability letters*, 22(4):333–338, 1995.

[4] N. B. Becker, R. J. Allen, and P. R. ten Wolde. Non-stationary forward flux sampling. *J. Chem. Phys.*, 136(17):174118, 2012.

[5] N. B. Becker, R. J. Allen, and P. R. ten Wolde. Non-stationary forward flux sampling. *The Journal of Chemical Physics*, 136(17):05B607, 2012.

[6] R. B. Best and G. Hummer. Coordinate-dependent diffusion in protein folding. *Proceedings of the National Academy of Sciences*, 107(3):1088–1093, 2010.

[7] G. Beylkin and M. J. Mohlenkamp. Numerical operator calculus in higher dimensions. *Proceedings of the National Academy of Sciences*, 99(16):10246–10251, 2002.

[8] G. Beylkin and M. J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, 2005.

[9] A. Bittracher, P. Koltai, and O. Junge. Pseudogenerators of spatial transfer operators. *SIAM Journal on Applied Dynamical Systems*, 14(3):1478–1517, 2015.

[10] E. M. Bollt and N. Santitissadeekorn. *Applied and Computational Measurable Dynamics*. Society for Industrial and Applied Mathematics, 2013.

[11] C. J. Bose and R. Murray. The exact rate of approximation in Ulam's method. *Discrete and Continuous Dynamical Systems*, 7(1):219–235, 2001.

[12] C. J. Bose and R. Murray. Dynamical conditions for convergence of a maximum entropy method for Frobenius–Perron operator equations. *Applied mathematics and computation*, 182(1):210–212, 2006.

[13] C. J. Bose and R. Murray. Minimum 'energy' approximations of invariant measures for nonsingular transformations. *Discrete and Continuous Dynamical Systems*, 14(3):597–615, 2006.

[14] C. J. Bose and R. Murray. Duality and the computation of approximate invariant densities for nonsingular transformations. *SIAM Journal on Optimization*, 18(2):691–709, 2007.

[15] A. Bovier, V. Gayrard, and M. Klein. Metastability in reversible diffusion processes II: Precise asymptotics for small eigenvalues. *J. Eur. Math. Soc.*, 7:69–99, 2002.

[16] G. R. Bowman, V. S. Pande, and F. Noé, editors. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*, volume 797 of *Advances in Experimental Medicine and Biology*. Springer, 2014.

[17] A. Boyarsky and P. Gora. *Laws of Chaos: Invariant Measures and Dynamical Systems in One Dimension*. Springer Science & Business Media, 2012.

[18] J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, Inc., 2nd edition, 2001.

[19] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz. Extracting spatial-temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, 258:1–15, 2016.

[20] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

[21] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz. Compressed sensing and dynamic mode decomposition. *Journal of Computational Dynamics*, 2(2):165–191, 2015.

[22] M. Budišić, R. Mohr, and I. Mezić. Applied Koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4), 2012.

[23] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:1–123, 2004.

[24] C. J. Camacho and D. Thirumalai. Kinetics and thermodynamics of folding in model proteins. *Proceedings of the National Academy of Sciences*, 90(13):6369–6372, 1993.

[25] D. A. Case, J. T. Berryman, R. M. Betz, D. S. Cerutti, T. E. Cheatham, T. A. Darden, R. E. Duke, T. J. Giese, H. Gohlke, A. W. Goetz, N. Homeyer, S. Izadi, P. Janowski, J. Kaus, A. Kovalenko, T. S. Lee, S. LeGrand, P. Li, T. Luchko, R. Luo, B. Madej, K. M. Merz, G. Monard, P. Needham, H. Nguyen, H. T. Nguyen, I. Omelyan, A. Onufriev, D. R. Roe, A. Roitberg, R. Salomon-Ferrer, C. L. Simmerling, W. Smith, J. Swails, R. C. Walker, J. Wang, R. M. Wolf, X. Wu, D. M. York, and P. A. Kollman. *AMBER 2015*. University of California, San Francisco, 2015.

[26] M. D. Chekroun, J. D. Neelin, D. Kondrashov, J. C. McWilliams, and M. Ghil. Rough parameter dependence in climate models and the role of Ruelle–Pollicott resonances. *Proceedings of the National Academy of Sciences*, 111(5):1684–1690, 2014.

[27] G. Chen and T. Ueta, editors. *Chaos in Circuits and Systems*. World Scientific Series on Nonlinear Science, Series B, Volume 11. World Scientific, 2002.

[28] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses. *Journal of Nonlinear Science*, 22(6):887–915, 2012.

[29] E. Chiavazzo, R. R. Coifman, R. Covino, C. W. Gear, A. S. Georgiou, G. Hummer, and I. G. Kevrekidis. iMapD: intrinsic map dynamics exploration for uncharted effective free energy landscapes. *arXiv preprint arXiv:1701.01513*, 2016.

[30] J. D. Chodera and F. Noé. Markov state models of biomolecular conformational dynamics. *Current Opinion in Structural Biology*, 25:135–144, 2014.

[31] G. Ciccotti, R. Kapral, and E. Vanden-Eijnden. Blue moon sampling, vectorial reaction coordinates, and unbiased constrained dynamics. *ChemPhysChem*, 6(9):1809–1814, 2005.

[32] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler. Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems. *Multiscale Modeling & Simulation*, 7(2):842–864, 2008.

[33] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[34] M. Crosskey and M. Maggioni. ATLAS: A geometric approach to learning high-dimensional stochastic systems near manifolds. *Multiscale Modeling & Simulation*, 15(1):110–156, 2017.

[35] E. Darve, D. Rodríguez-Gómez, and A. Pohorille. Adaptive biasing force method for scalar and vector free energy calculations. *The Journal of Chemical Physics*, 128(14):144120, 2008.

[36] V. de Silva and L.-H. Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[37] C. Dellago and P. G. Bolhuis. Transition path sampling and other advanced simulation techniques for rare events. In C. Holm and K. Kremer, editors, *Advanced Computer Simulation Approaches for Soft Matter Sciences III*, pages 167–233. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[38] M. Dellnitz, G. Froyland, and O. Junge. The algorithms behind GAIO – Set oriented numerical methods for dynamical systems. In *Ergodic theory, analysis, and efficient simulation of dynamical systems*, pages 145–174. Springer, 2000.

[39] M. Dellnitz and A. Hohmann. A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numerische Mathematik*, 75(3):293–317, 1997.

[40] M. Dellnitz and O. Junge. On the approximation of complicated dynamical behavior. *SIAM Journal on Numerical Analysis*, 36(2):491–515, 1999.

[41] M. Dellnitz, M. H. von Molo, and A. Ziessler. On the computation of attractors for delay differential equations. *Journal of Computational Dynamics*, 3(1):93–112, 2016.

[42] P. Deuflhard and M. Weber. Robust Perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161 – 184, 2005.

[43] J. Ding. A maximum entropy method for solving Frobenius–Perron operator equations. *Applied Mathematics and Computation*, 93:155–168, 1998.

[44] J. Ding, Q. Du, and T.-Y. Li. High order approximation of the Frobenius–Perron operator. *Applied Mathematics and Computation*, 53:151–171, 1993.

[45] J. Ding and T.-Y. Li. Markov finite approximation of the Frobenius–Perron operator. *Nonlinear Analysis: Theory, Methods & Applications*, 17:759–772, 1991.

[46] J. Ding and A. Zhou. Finite approximations of Frobenius–Perron operators. A solution of Ulam's conjucture on multi-dimensional transformations. *Physica D*, 92:61–68, 1996.

[47] N. Djurdjevac, M. Sarich, and C. Schütte. Estimating the eigenvalue error of Markov state models. *Multiscale Modeling & Simulation*, 10(1):61–81, 2012.

[48] S. V. Dolgov. *Tensor product methods in numerical simulation of high-dimensional dynamical problems*. PhD thesis, Universität Leipzig, 2014.

[49] S. V. Dolgov, B. N. Khoromskij, I. V. Oseledets, and D. V. Savostyanov. Computation of extreme eigenvalues in higher dimensions using block tensor train format. *Computer Physics Communications*, 185(4):1207–1216, 2014.

[50] C. J. Dsilva, R. Talmon, C. W. Gear, R. R. Coifman, and I. G. Kevrekidis. Data-driven reduction for a class of multiscale fast-slow stochastic dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 15(3):1327–1351, 2016.

[51] R. Du, V. S. Pande, A. Y. Grosberg, T. Tanaka, and E. S. Shakhnovich. On the transition coordinate for protein folding. *The Journal of chemical physics*, 108(1):334–350, 1998.

[52] W. E and B. Engquist. The heterogeneous multiscale method. *Communications in Mathematical Sciences*, 1(1):87–132, 2003.

[53] W. E, W. Ren, and E. Vanden-Eijnden. String method for the study of rare events. *Physical Review B*, 66(5):052301, 2002.

[54] W. E and E. Vanden-Eijnden. Towards a theory of transition paths. *J. Stat. Phys.*, 123(3):503–523, 2006.

[55] W. E and E. Vanden-Eijnden. Transition-path theory and path-finding algorithms for the study of rare events. *Annu. Rev. Phys. Chem.*, 61(1):391–420, 2010.

[56] A. K. Faradjian and R. Elber. Computing time scales from reaction coordinates by milestoning. *J. Chem. Phys.*, 120:10880–10889, 2004.

[57] A. K. Faradjian and R. Elber. Computing time scales from reaction coordinates by milestoning. *The Journal of chemical physics*, 120(23):10880–10889, 2004.

[58] H. Federer. *Geometric measure theory*. Springer New York, 1969.

[59] A. L. Ferguson, A. Z. Panagiotopoulos, I. G. Kevrekidis, and P. G. Debenedetti. Nonlinear dimensionality reduction in molecular simulation: The diffusion map approach. *Chemical Physics Letters*, 509(1):1–11, 2011.

[60] M. Freidlin and A. D. Wentzell. *Random perturbations of dynamical systems*. Springer, New York, 1998.

[61] G. Friesecke, O. Junge, and P. Koltai. Mean field approximation in conformation dynamics. *Multiscale Modeling & Simulation*, 8(1):254–268, 2009.

[62] G. Froyland. Approximating physical invariant measures of mixing dynamical systems. *Nonlinear Analysis, Theory, Methods, & Applications*, 32:831–860, 1998.

[63] G. Froyland, C. González-Tokman, and T. M. Watson. Optimal mixing enhancement by local perturbation. 2015. Preprint.

[64] G. Froyland, G. Gottwald, and A. Hammerlindl. A computational method to extract macroscopic variables and their dynamics in multiscale systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1816–1846, 2014.

[65] G. Froyland, G. A. Gottwald, and A. Hammerlindl. A trajectory-free framework for analysing multiscale systems. *ArXiv e-prints*, 2014.

[66] G. Froyland and O. Junge. On fast computation of finite-time coherent sets using radial basis functions. *Chaos*, 25(8), 2015.

[67] G. Froyland, O. Junge, and P. Koltai. Estimating long term behavior of flows without trajectory integration: The infinitesimal generator approach. *SIAM Journal on Numerical Analysis*, 51(1):223–247, 2013.

[68] G. Froyland and K. Padberg. Almost-invariant sets and invariant manifolds — Connecting probabilistic and geometric descriptions of coherent structures in flows. *Physica D*, 238:1507–1523, 2009.

[69] G. Froyland and K. Padberg-Gehle. Almost-invariant and finite-time coherent sets: Directionality, duration, and diffusion. In W. Bahsoun, C. Bose, and G. Froyland, editors, *Ergodic Theory, Open Dynamics, and Coherent Structures*, pages 171–216. Springer New York, 2014.

[70] G. Froyland, R. M. Stuart, and E. van Sebille. How well-connected is the surface of the global ocean? *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 24(3), 2014.

[71] P. Gelß. *The tensor-train format and its applications: Modeling and analysis of chemical reaction networks, catalytic processes, fluid flows, and Brownian dynamics.* PhD thesis, Freie Universität Berlin, 2017.

[72] P. Gelß, S. Klus, S. Matera, and C. Schütte. Nearest-neighbor interaction systems in the tensor train format. *Journal of Computational Physics*, 341:140–162, 2017.

[73] P. Gelß, S. Matera, and C. Schütte. Solving the master equation without kinetic Monte Carlo: Tensor train approximations for a CO oxidation model. *Journal of Computational Physics*, 314:489–502, 2016.

[74] D. Giannakis. Data-driven spectral decomposition and forecasting of ergodic dynamical systems. *ArXiv e-prints*, 2015.

[75] G. H. Golub and C. F. V. Loan. *Matrix Computations.* Johns Hopkins University Press, 4th edition, 2013.

[76] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.

[77] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*, volume 42 of *Springer Series in Computational Mathematics.* Springer, 2012.

[78] W. Hackbusch. Numerical tensor calculus. *Acta Numerica*, 23:651–742, 2014.

[79] P. R. Halmos. *Lectures on ergodic theory*, volume 142. American Mathematical Soc., 1956.

[80] S. Holtz, T. Rohwedder, and R. Schneider. The alternating linear scheme for tensor optimization in the tensor train format. *SIAM Journal on Scientific Computing*, 34(2):A683–A713, 2012.

[81] E. Hopf. The general temporally discrete Markoff process. *Journal of Rational Mechanics and Analysis*, 3(1):13–45, 1954.

[82] P. Huber. Dünngitter-Spektralmethoden zur Approximation des Frobenius–Perron-Operators. Diploma thesis (in German), Technische Universität München, 2009.

[83] W. Huisinga, S. Meyn, and C. Schütte. Phase transitions & metastability in Markovian and molecular systems. *The Annals of Applied Probability*, 14 (1):419–458, 2004.

[84] B. Hunt and V. Kaloshin. Regularity of embeddings of infinite-dimensional fractal sets into finite-dimensional spaces. *Nonlinearity*, 12(5):1263–1275, 1999.

[85] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[86] H. Jasak, A. Jemcov, and Z. Tukovic. OpenFOAM: A C++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, volume 1000, pages 1–20, 2007.

[87] M. R. Jovanović, P. J. Schmid, and J. W. Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26(2), 2014.

[88] O. Junge and P. Koltai. Discretization of the Frobenius–Perron operator using a sparse Haar tensor basis: The Sparse Ulam method. *SIAM Journal on Numerical Analysis*, 47:3464–3485, 2009.

[89] I. G. Kevrekidis and G. Samaey. Equation-free multiscale computation: Algorithms and applications. *Annu. Rev. Phys. Chem.*, 60(1):321–344, 2009.

[90] A. Klimovskaia, S. Ganscha, and M. Claassen. Sparse regression based structure learning of stochastic reaction networks from single cell snapshot time series. *PLOS Computational Biology*, 12(12):1–20, 2016.

[91] S. Klus, P. Gelß, S. Peitz, and C. Schütte. Tensor-based dynamic mode decomposition. *Nonlinearity*, 31(7):3359–3380, 2018.

[92] S. Klus, P. Koltai, and C. Schütte. On the numerical approximation of the Perron–Frobenius and Koopman operator. *Journal of Computational Dynamics*, 3(1):51–79, 2016.

[93] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, 28:985–1010, 2018.

[94] S. Klus, I. Schuster, and K. Muandet. Eigendecompositions of transfer operators in reproducing kernel Hilbert spaces. *Journal of Nonlinear Science*, 30:283–315, 2019.

[95] S. Klus and C. Schütte. Towards tensor-based methods for the numerical approximation of the Perron–Frobenius and Koopman operator. *Journal of Computational Dynamics*, 3(2):139–161, 2016.

[96] P. Koltai. *Efficient approximation methods for the global long-term behavior of dynamical systems – Theory, algorithms and examples.* PhD thesis, Technische Universität München, 2010.

[97] B. Koopman. Hamiltonian systems and transformations in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315, 1931.

[98] M. Korda and I. Mezić. On Convergence of Extended Dynamic Mode Decomposition to the Koopman Operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018.

[99] U. Krengel. *Ergodic theorems*, volume 6 of *de Gruyter Studies in Mathematics*. Walter de Gruyter & Co., Berlin, 1985.

[100] D. Kressner and C. Tobler. Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1288–1316, 2011.

[101] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[102] J. B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964.

[103] S. Kumar, J. M. Rosenberg, D. Bouzida, R. H. Swendsen, and P. A. Kollman. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *Journal of Computational Chemistry*, 13(8):1011–1021, 1992.

[104] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems.* SIAM, 2016.

[105] A. Laio and F. L. Gervasio. Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. *Reports on Progress in Physics*, 71(12):126601, 2008.

[106] A. Laio and M. Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, 2002.

[107] A. Lasota and M. C. Mackey. *Chaos, fractals, and noise: Stochastic aspects of dynamics*, volume 97 of *Applied Mathematical Sciences*. Springer, 2nd edition, 1994.

[108] N. Lee and A. Cichocki. Regularized computation of approximate pseudoinverse of large matrices using low-rank tensor train decompositions. *SIAM Journal on Matrix Analysis and Applications*, 37(2):598–623, 2016.

[109] F. Legoll and T. Lelièvre. Effective dynamics using conditional expectations. *Nonlinearity*, 23(9):2131, 2010.

[110] B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, and R. Skeel. *New Algorithms for Macromolecular Simulation (Lecture Notes in Computational Science and Engineering)*. Springer-Verlag New York, 2006.

[111] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103111, 2017.

[112] T.-Y. Li. Finite approximation for the Frobenius–Perron operator. A solution to Ulam's conjecture. *Journal of Approximation Theory*, 17:177–186, 1976.

[113] W. Li and A. Ma. Recent developments in methods for identifying reaction coordinates. *Molecular simulation*, 40(10-11), 2014.

[114] J. Lu and E. Vanden-Eijnden. Exact dynamical coarse-graining without time-scale separation. *The Journal of chemical physics*, 141(4):07B619_1, 2014.

[115] A. Ma and A. R. Dinner. Automatic method for identifying reaction coordinates in complex systems. *J. Phys. Chem. B*, 109:6769–6779, 2005.

[116] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696. ACM, 2009.

[117] L. Maragliano and E. Vanden-Eijnden. A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations. *Chemical physics letters*, 426(1):168–175, 2006.

[118] A. Mardt, L. Pasquali, H. Wu, and F. Noé. VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9, 2018.

[119] J. C. Mattingly and A. M. Stuart. Geometric ergodicity of some hypo-elliptic diffusions for particle motions. *Markov Process. Related Fields*, 8(2):199–214, 2002.

[120] J. C. Mattingly, A. M. Stuart, and D. J. Higham. Ergodicity for SDEs and approximations: locally Lipschitz vector fields and degenerate noise. *Stochastic processes and their applications*, 101(2):185–232, 2002.

[121] A. Mauroy and J. Goncalves. Linear identification of nonlinear systems: A lifting technique based on the koopman operator. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6500–6505, 2016.

[122] R. T. McGibbon, B. E. Husic, and V. S. Pande. Identification of simple reaction coordinates from complex dynamics. *The Journal of Chemical Physics*, 146(4):044109, 2017.

[123] R. T. McGibbon and V. S. Pande. Variational cross-validation of slow dynamical modes in molecular kinetics. *The Journal of Chemical Physics*, 142(12), 2015.

[124] P. Metzner, C. Schütte, and E. Vanden-Eijnden. Transition path theory for Markov jump processes. *Multiscale Modeling and Simulation*, 7(3):1192–1219, 2009.

[125] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[126] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005.

[127] I. Mezić. Analysis of fluid flows via spectral properties of the Koopman operator. *Annual Review of Fluid Mechanics*, 45(1):357–378, 2013.

[128] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical Review Letters*, 72:3634–3637, 1994.

[129] D. Moroni, T. van Erp, and P. Bolhuis. Investigating rare events by transition interface sampling. *Physica A*, 340(1):395–401, 2004.

[130] R. Murray. *Discrete approximation of invariant densities*. PhD thesis, University of Cambridge, 1997.

[131] R. Murray. Optimal partition choice for invariant measure approximation for one-dimensional maps. *Nonlinearity*, 17(5):1623–1644, 2004.

[132] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006.

[133] F. Noé and C. Clementi. Kinetic distance and kinetic maps from molecular dynamics simulation. *Journal of Chemical Theory and Computation*, 11(10):5002–5011, 2015.

[134] F. Noé and F. Nüske. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Modeling & Simulation*, 11(2):635–655, 2013.

[135] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl. Constructing the full ensemble of folding pathways from short off-equilibrium simulations. *Proceedings of the National Academy of Sciences*, 106:19011–19016, 2009.

[136] F. Noé, H. Wu, J.-H. Prinz, and N. Plattner. Projected and Hidden Markov Models for Calculating Kinetics and Metastable States of Complex Molecules. *The Journal of Chemical Physics*, 139:184114, 2013.

[137] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé. Variational approach to molecular kinetics. *Journal of Chemical Theory and Computation*, 10(4):1739–1752, 2014.

[138] F. Nüske, R. Schneider, F. Vitalini, and F. Noé. Variational tensor approach for approximating the rare-event kinetics of macromolecular systems. *The Journal of Chemical Physics*, 144(5), 2016.

[139] S. Ober-Blöbaum and K. Padberg-Gehle. Multiobjective optimal control of fluid mixing. *PAMM*, 15(1):639–640, 2015.

[140] B. Øksendal. *Stochastic Differential Equations*. Springer, 6th edition, 2003.

[141] D. Ornstein. Bernoulli shifts with the same entropy are isomorphic. *Advances in Mathematics*, 4(3):337–352, 1970.

[142] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[143] I. V. Oseledets. *TT-toolbox 2.0: Fast multidimensional array operations in TT format*, 2011.

[144] S. E. Otto and C. W. Rowley. Linearly-recurrent autoencoder networks for learning dynamics. *ArXiv e-prints*, 2017.

[145] V. S. Pande, K. Beauchamp, and G. R. Bowman. Everything you wanted to know about Markov state models but were afraid to ask. *Methods*, 52(1):99–105, 2010.

[146] G. Pavliotis and A. Stuart. *Multiscale methods: averaging and homogenization*. Springer Science & Business Media, 2008.

[147] G. A. Pavliotis. *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, volume 60 of *Texts in Applied Mathematics*. Springer, 2014.

[148] A. Pazy. *Semigroups of linear operators and applications to partial differential equations*. Springer, 1983.

[149] D. Perez-Garcia, F. Verstraete, M. Wolf, and J. Cirac. Matrix product state representations. *Quantum Information & Computation*, 7(5):401–430, 2007.

[150] G. Pérez-Hernández, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé. Identification of slow molecular order parameters for Markov model construction. *The Journal of Chemical Physics*, 139(1), 2013.

[151] Z. D. Pozun, K. Hansen, D. Sheppard, M. Rupp, K.-R. Mller, and G. Henkelman. Optimizing transition states via kernel-based machine learning. *The Journal of Chemical Physics*, 136(17):174101, 2012.

[152] R. Preis, M. Dellnitz, M. Hessel, C. Schütte, and E. Meerbach. Dominant paths between almost invariant sets of dynamical systems. DFG Schwerpunktprogramm 1095, Technical Report 154, 2004.

[153] J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé. Markov Models of Molecular Kinetics: Generation and Validation. *Journal of Chemical Physics*, 134:174105, 2011.

[154] H. Rauhut, R. Schneider, and Z. Stojanac. Tensor completion in hierarchical tensor representations. In H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral, editors, *Compressed Sensing and its Applications: MATHEON Workshop 2013*, pages 419–450. Springer International Publishing, 2015.

[155] W. Ren, E. Vanden-Eijnden, P. Maragakis, and W. E. Transition pathways in complex systems: Application of the finite-temperature string method to the alanine dipeptide. *The Journal of chemical physics*, 123(13):134109, 2005.

[156] J. C. Robinson. A topological delay embedding theorem for infinite-dimensional dynamical systems. *Nonlinearity*, 18(5):2135–2143, 2005.

[157] S. Röblitz and M. Weber. Fuzzy spectral clustering by PCCA+: application to Markov state models and data classification. *Advances in Data Analysis and Classification*, 7(2):147–179, 2013.

[158] G. Roullet. 2D versatile model, 2013. `http://stockage.univ-brest.fr/~roullet/codes.html`.

[159] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.

[160] M. Sarich, F. Noé, and C. Schütte. On the approximation quality of Markov state models. *Multiscale Model. Simul.*, 8:1154–1177, 2010.

[161] T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *Journal of Statistical Physics*, 65(3):579–616, 1991.

[162] M. J. Schervish and B. P. Carlin. On the convergence of successive substitution sampling. *Journal of Computational and Graphical statistics*, 1(2):111–127, 1992.

[163] P. Schmid and J. Sesterhenn. Dynamic Mode Decomposition of numerical and experimental data. In *61st Annual Meeting of the APS Division of Fluid Dynamics*. American Physical Society, 2008.

[164] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

[165] U. Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1), 2011.

[166] Schrödinger, LLC. The PyMOL molecular graphics system, Version 1.7.4, 2014.

[167] C. Schütte. Conformational dynamics: Modelling, theory, algorithm, and application to biomolecules, 1999. Habilitation Thesis.

[168] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard. A direct approach to conformational dynamics based on hybrid Monte Carlo. *Journal of Computational Physics*, 151(1):146–168, 1999.

[169] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard. A direct approach to conformational dynamics based on hybrid Monte Carlo. *J. Comput. Phys.*, 151(1):146–168, 1999.

[170] C. Schütte, F. Noé, J. Lu, M. Sarich, and E. Vanden-Eijnden. Markov state models based on milestoning. *J. Chem. Phys.*, 134(20), 2011.

[171] C. Schütte and M. Sarich. *Metastability and Markov State Models in Molecular Dynamics: Modeling, Analysis, Algorithmic Approaches*. Number 24 in Courant Lecture Notes. American Mathematical Society, 2013.

[172] C. R. Schwantes and V. S. Pande. Improvements in Markov State Model construction reveal many non-native interactions in the folding of NTL9. *Journal of Chemical Theory and Computation*, 9:2000–2009, 2013.

[173] C. R. Schwantes and V. S. Pande. Modeling molecular kinetics with tICA and the kernel trick. *Journal of Chemical Theory and Computation*, 11(2):600–608, 2015.

[174] S. C. Shadden, F. Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3):271–304, 2005.

[175] Y. G. Sinai. On the notion of entropy of dynamical systems. In *Doklady Akademii Nauk*, volume 124, pages 768–771, 1959.

[176] A. Singer, R. Erban, I. G. Kevrekidis, and R. R. Coifman. Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps. *Proceedings of the National Academy of Sciences*, 106(38):16090–160955, 2009.

[177] N. Socci, J. N. Onuchic, and P. G. Wolynes. Diffusive dynamics of the reaction coordinate for protein folding funnels. *The Journal of chemical physics*, 104(15):5860–5868, 1996.

[178] F. Takens. Detecting strange attractors in turbulence. In *Springer Lecture Notes in Mathematics*, volume 898, pages 366–381. Springer, 1981.

[179] A. Tantet, V. Lucarini, F. Lunkeit, and H. A. Dijkstra. Crisis of the chaotic attractor of a climate model: a transfer operator approach. *Preprint, arXiv:1507.02228*, 2015.

[180] A. Tantet, F. R. van der Burgt, and H. A. Dijkstra. An early warning indicator for atmospheric blocking events using transfer operators. *Chaos*, 25(3), 2015.

[181] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[182] L. Tong, V. C. Soon, Y. F. Huang, and R. Liu. AMUSE: a new blind identification algorithm. In *IEEE International Symposium on Circuits and Systems*, pages 1784–1787, 1990.

[183] G. M. Torrie and J. P. Valleau. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *J. Comput. Phys.*, 23(2):187–199, 1977.

[184] G. M. Torrie and J. P. Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, 1977.

[185] L. N. Trefethen and M. Embree. *Spectra and pseudospectra: the behavior of nonnormal matrices and operators*. Princeton University Press, 2005.

[186] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2), 2014.

[187] S. M. Ulam. *A Collection of Mathematical Problems*. Interscience Publisher NY, 1960.

[188] U. Vaidya, P. G. Mehta, and U. V. Shanbhag. Nonlinear stabilization via control Lyapunov measure. *IEEE Transactions on Automatic Control*, 55(6):1314–1328, 2010.

[189] E. Vanden-Eijnden. Transition path theory. *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology Volume 1*, pages 453–493, 2006.

[190] E. Vanden-Eijnden. On HMM-like integrators and projective integration methods for systems with multiple time scales. *Commun. Math. Sci.*, 5(2):495–505, 2007.

[191] M. Weber. *Meshless Methods in Conformation Dynamics*. PhD thesis, Freie Universität Berlin, 2006.

[192] M. Weber. A subspace approach to molecular Markov state models via a new infinitesimal generator., 2012. Habilitation thesis.

[193] M. Weber, K. Fackeldey, and C. Schütte. Set-free Markov state model building. *J. Chem. Phys.*, 146:124133, 2017.

[194] H. Whitney. Differentiable manifolds. *Annals of Mathematics*, 37(3):645–680, 1936.

[195] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[196] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel-based method for data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2015.

[197] M. O. Williams, I. I. Rypina, and C. W. Rowley. Identifying finite-time coherent sets from limited quantities of Lagrangian data. *Chaos*, 25(8), 2015.

[198] H. Wu, F. Nüske, F. Paul, S. Klus, P. Koltai, and F. Noé. Variational Koopman models: Slow collective variables and molecular kinetics from short off-equilibrium simulations. *The Journal of Chemical Physics*, 146(15):154104, 2017.

[199] W. Zhang, C. Hartmann, and C. Schütte. Effective dynamics along given reaction coordinates, and reaction rate theory. *Faraday Discussions*, 195:365–394, 2016.

[200] W. Zhang and C. Schuette. Reliable approximation of long relaxation timescales in molecular dynamics. *Submitted to Entropy*, 2017.

[201] A. Ziehe and K.-R. Müller. TDSEP — an efficient algorithm for blind separation using time structure. In *ICANN 98*, pages 675–680. Springer Science and Business Media, 1998.

# A

# Transition manifolds of complex metastable systems: Theory and data-driven computation of effective dynamics

Andreas Bittracher[1], Péter Koltai[1], Stefan Klus[1], Ralf Banisch[1], Michael Dellnitz[2], and Christof Schütte[1,3]

[1] Department of Mathematics and Computer Science, Freie Universität Berlin, Germany
[2] Department of Mathematics, University of Paderborn, Germany
[3] Zuse Institute Berlin, Germany

*Journal of Nonlinear Science*, 28(2):471–512, 2018

DOI: 10.1007/s00332-017-9415-0

**Abstract**

We consider complex dynamical systems showing metastable behavior but no local separation of fast and slow time scales. The article raises the question of whether such systems exhibit a low-dimensional manifold supporting its effective dynamics. For answering this question, we aim at finding nonlinear coordinates, called reaction coordinates, such that the projection of the dynamics onto these coordinates preserves the dominant time scales of the dynamics. We show that, based on a specific reducibility property, the existence of good low-dimensional reaction coordinates preserving the dominant time scales is guaranteed. Based on this theoretical framework, we develop and test a novel numerical approach for computing good reaction coordinates. The proposed algorithmic approach is fully local and thus not prone to the curse of dimension with respect to the state space of the dynamics. Hence, it is a promising method for data-based model reduction of complex dynamical systems such as molecular dynamics.

## A.1 Introduction

With the advancement of computing power, we are able to simulate and analyze more and more complicated and high-dimensional models of dynamical systems, ranging from astronomical scales for the simulation of galaxies, over planetary and continental scales for climate and weather prediction, down to molecular and sub-atomistic scales via, e.g., Molecular Dynamics (MD) simulations aimed at gaining insight into complex biological

processes. Particular aspects of such processes, however, can often be described by much simpler means than the full process, thus *reducing* the full dynamics to some *essential* behavior or *effective dynamics* in terms of some essential observables of the system. Extracting these observables and the related effective dynamics from a dynamical system, though, is one of the most challenging problems in computational modeling [64].

One prominent example of dynamical reduction is arguably given by a variety of multiscale systems with explicit fast-slow time scale separation, mostly singularly perturbed systems, where either the fast component is considered in a quasi-stationary regime (i.e. the slow components are fixed and assumed not to change for the observation period), or the effective behavior of the fast components is injected into the slow processes, e.g. by averaging or homogenization [146]. Much of the recent attention has been directed to the case where the deduction of the slow (or fast) effective dynamics is not possible by purely analytic means, due to the lack of an analytic description of the system, or because the complexity of the system renders this task unfeasible [64, 65, 32, 50, 132, 176, 34, 190, 89]. However, all of these approaches still depend on some local form of time scale separation between the "fast" and the "slow" components of the dynamics.

The focus of this work is on specific multiscale systems *without* local dynamical slow-fast time scale separation, but for which a reduction to an effective dynamical behavior supported on some low-dimensional manifold is still possible. The dynamical property lying at the heart of our approach is that there is a time scale separation in the *global kinetic* behavior of the process, as opposed to the aforementioned slow-fast behavior encoded in the *local dynamics*. Here, global kinetic behavior means that the multiple scales show up if we consider the *Fokker–Planck equation* associated with the dynamics, say $\dot{u} = \mathcal{L}u$, where the Fokker–Planck operator $\mathcal{L}$ will have several small eigenvalues, while the rest of its spectrum is significantly larger. Such dynamical systems exhibit *metastable* behavior and the slow time scales are the time scales of statistical relaxation between the main metastable sets, while there is no time scale gap for the local dynamics within each of the metastable regions [15, 171].

Global time scale separation induced by metastability has been analyzed for deterministic [40] and stochastic dynamical systems [169, 83] for more than a decade. A typical trajectory of a metastable dynamical system will spend most time within the metastable sets, while rare transitions between these sets happen as sudden "jumps" roughly along low-dimensional *transition pathways* that connect the metastable sets [37, 135, 55]. For an example, see Figure A.1.

The tool to describe the global kinetic behavior of a metastable system is the so-called *transfer operator* (the evolution operator of the Fokker–Planck equation), which acts on functions on the state space. The time scale separation we rely on here implies a spectral gap for this operator. This fact has been exploited to find low-dimensional representations of the global kinetics in form of Markov chains whose (discrete) states represent the metastable sets while the transition probabilities between the states approximate the jump statistics between the sets on long time scales. Under the name "Markov State Models" (MSM), this approach has led to a variety of methods [16, 171] with broad application, e.g., in molecular dynamics, cf. [169, 145, 170, 30]. This reduction comes with a price:

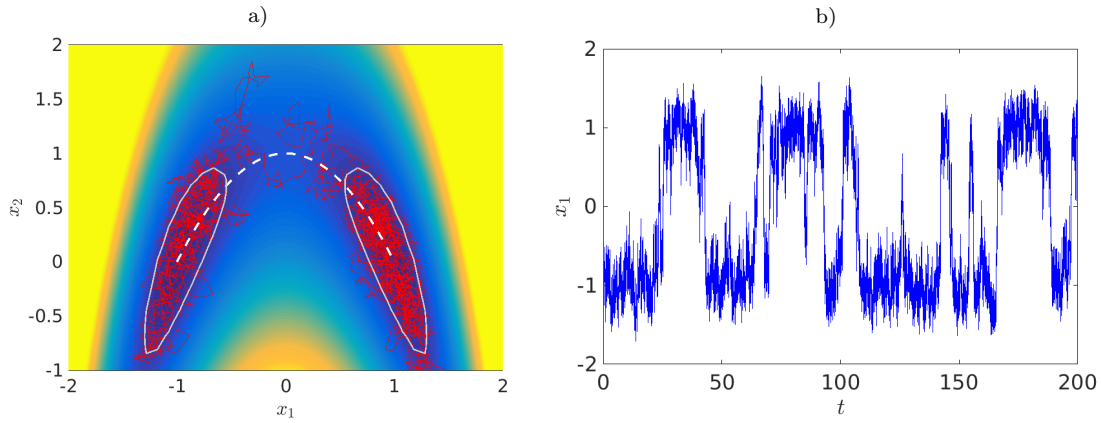



**Figure A.1:** a) Curved double-well potential with two metastable sets (areas encircled by light grey lines) around the global minima $(-1, 0)$ and $(1, 0)$. In a typical trajectory (red line), transitions between the metastable sets are rare events and generally happen along the transition path (white dashed line). b) The $x_1$-component of a longer trajectory that shows multiple rare transitions (or events).

Since the relaxation kinetics is described just by jumps between the metastable sets in a (finite) discrete state space, any information about the transition process and its dynamical features is lost. A variety of approaches have been developed for complementing the MSM approach appropriately [124], but a continuous (in time and space) low-dimensional effective description based on MSMs allowing to understand the transition mechanism is infeasible.

In another branch of the literature, again heavily influenced by molecular dynamics applications, model reduction techniques have been developed that assume the existence of a low-dimensional *reaction coordinate* or *order parameter* in order to construct an effective dynamics or kinetics: Examples are free energy based techniques [183, 105], trajectory-based sampling techniques [56, 4, 129, 150], methods based on diffusive processes [6, 199, 146], and many more that rely on the assumption that the reaction coordinates are known. The problem of actually *constructing* good reaction coordinates remains an area of ongoing research [113], to which this paper contributes. Typically, reaction coordinates are either postulated using system specific expert knowledge [24, 177], an approximation to the dominant eigenfunctions of the transfer operator is sought [171, 30, 150], or machine learning techniques are proposed [115]. Froyland et al. [64] show that these eigenfunctions are indeed optimal — in the sense of optimally representing the slow dynamics — but for high dimensional systems computational reaction coordinate identification still is often infeasible. In the context of *transition path theory* [189], the committor function is known to be an ideal [114] reaction coordinate. In [151], the authors construct a level set of the committor using support vector machines, but the computation of reaction coordinates is infeasible for high-dimensional systems. The main problem in computing reaction coordinates for high-dimensional metastable systems results from the fact that all of

these algorithms try to solve a *global* problem in the entire state space that cannot be decomposed easily into purely local computations.

In this article, we elaborate on the definition, existence and algorithmic identification of reaction coordinates for metastable systems: We define reaction coordinates as a small set of *nonlinear* coordinates on which a reduced system [109, 199] can be defined having the same dominant time scales (in terms of transfer operator eigenvalues) as the original system. We then consider a low-dimensional state space on which the reduced dynamics is a Markov process. Thus, our approach utilizes concepts and transfer operator theory developed previously, but in our case the projected transfer operator is still *infinite-dimensional*, in stark contrast to its reduction to a stochastic matrix in the MSM approach.

The contribution of this paper is twofold. First, we develop a conceptual framework that identifies *good* reaction coordinates as the ones that parameterize a low-dimensional *transition manifold* $\mathbb{M}$ in the function space $L^1$, which is the natural state space of the Fokker–Planck equation $\dot{u} = \mathcal{L}u$ associated with the dynamics. The property which defines $\mathbb{M}$ is that, on moderate time scales $t_{\text{fast}} < t \ll t_{\text{slow}}$, the *transition density functions* of the dynamics concentrate around $\mathbb{M}$. We provide evidence that such an $\mathbb{M}$ indeed exists due to metastability and the existence of transition pathways. Crucially, the dimension of $\mathbb{M}$ is often lower then the number of dominant eigenfunctions.

Second, we present an algorithm to construct approximate reaction coordinates. Our algorithm is data-driven and *fully local*, thus circumventing the main problem of previously proposed algorithms: In order to compute the value of the desired reaction coordinate $\xi$ at a location $x$ in the state space $\mathbb{X}$, only the ability to simulate short trajectories initialized at $x$ is needed. In particular, we assume no a priori knowledge of metastable sets, no global equilibration, and we do not need to resolve the slow time scales numerically. The algorithm is built on two pillars:

1. The simulation time scale $t$ can be chosen a lot smaller than the dominant time scales $t_{\text{slow}}$ of the system, such that it is feasible to simulate many short trajectories of length $t$.

2. We utilize *embedding techniques* inspired by the seminal work of Whitney [194] and the recent work [41] that allows one to take almost any mapping into a Euclidean space of more than twice the dimension of the manifold $\mathbb{M}$ and to obtain a one-to-one image of it.

These two pillars together with the low-dimensionality of $\mathbb{M}$ imply that we can represent the image of the reaction coordinate in a space with moderate (finite) dimension. Then, we can use established *manifold learning* techniques [132, 32, 176] to obtain a parametrization of the manifold in the embedding space and pull this parametrization back to the original state space, hence obtaining a reaction coordinate.

The locality of the algorithm also implies that reaction coordinates are only computed in the region of state space where sampled points are available. This is a common issue with manifold learning algorithms; here it manifests as the transition manifold being reliably learned only in regions we have good sampling coverage of. However, recently several

methods have appeared in the literature that allow a fast exploration of the state space. These methods do not provide equilibrium sampling, but instead try to rapidly cover the essential part of the state space with sampling points. This can be achieved with enhanced sampling methods such as Umbrella Sampling [103, 184], Metadynamics [105, 106], Blue-Moon sampling [31], Adaptive Biasing Force method [35], or Temperature-Accelerated Molecular Dynamics [117], as well as trajectory-based techniques like Milestoning [57], Transition Interface Sampling [129], or Forward Flux Sampling [5]. Alternatively, several techniques like the equation-free approach [89], the heterogeneous multiscale method (HMM) [52] and methods based on diffusion maps [29] have been developed to utilize short unbiased MD trajectories for extracting information that allows much larger timesteps. This can be combined with reaction coordinate based effective dynamics [199, 200].

In principle, the method we present in this article may be combined with any enhanced sampling technique in order to generate sampling points that cover a large part of the state space. For simplicity, we will use long MD trajectories to generate our sampling points, but we do not require that the points are distributed according to an equilibrium distribution.

The paper is organized as follows: Section A.2 introduces transfer operators, which describe the global kinetics of the stochastic process. Based on these transfer operators, we define metastability, i.e. the existence of dominant time scales. In Section A.3, we describe the model reduction techniques *Markov state modeling* and *coordinate projection* that are designed to capture the dominant time scales of metastable systems. Furthermore, we characterize *good* reaction coordinates. In the first part of Section A.4, we show that our dynamical assumption ensures the existence of good reaction coordinates, then in the second part we describe our approach to compute them. Several numerical examples are given in Section A.5. Concluding remarks and an outlook are provided in Section A.6.

## A.2 Transfer operators and their properties

As mentioned in the introduction, global properties of dynamical systems such as metastable sets or a partitioning into fast and slow subprocesses can be obtained using transfer operators associated with the system and their eigenfunctions. In this section, we will introduce different transfer operators needed for our considerations.

### A.2.1 Transfer operators

In what follows, $\mathsf{P}[\,\cdot \mid \mathfrak{E}]$ denotes probabilities conditioned on the event $\mathfrak{E}$ and $\mathsf{E}[\cdot \mid \mathfrak{E}]$ the expectation value. Furthermore, $\{\mathbf{X}_t\}_{t\geq 0}$ is a stochastic process defined on a state space $\mathbb{X} \subset \mathbb{R}^n$.

**Definition A.2.1** (Transition density function)**.** *Let $\mathbb{A}$ be any measurable set, then the* transition density function $p^t \colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}_{\geq 0}$ *of a time-homogeneous stochastic process*

$\{\mathbf{X}_t\}_{t\geq 0}$ *is defined by*

$$\mathsf{P}[\mathbf{X}_t \in \mathbb{A} \mid \mathbf{X}_0 = x] = \int_{\mathbb{A}} p^t(x,y)\,\mathrm{d}y.$$

*That is,* $p^t(x,y)$ *is the conditional probability density of* $\mathbf{X}_t = y$ *given that* $\mathbf{X}_0 = x$*.*

With the aid of the transition density function, we can now define transfer operators. Note, however, that the transition density is in general not known explicitly and needs to be estimated from simulation data. In what follows, we assume that there is a *unique* equilibrium density $\varrho$ that is invariant under $\{\mathbf{X}_t\}_{t\geq 0}$, that is, it satisfies

$$\varrho(x) = \int_{\mathbb{X}} p^t(y,x)\varrho(y)\,\mathrm{d}y,$$

a.e. on $\mathbb{X}$. Let $\mu$ denote the associated invariant measure $\mathrm{d}\mu = \varrho\,\mathrm{d}x$.

**Definition A.2.2** (Transfer operators)**.** *Let* $p \in L^1(\mathbb{X})$ *be a probability density*[1]*,* $u = p/\varrho \in L^1_\mu(\mathbb{X})$ *be a probability density with respect to the equilibrium density* $\varrho$*, and* $f \in L^\infty(\mathbb{X})$ *an observable of the system. For a given lag time* $t$*:*

*(a) The* Perron–Frobenius operator $\mathcal{P}^t\colon L^1(\mathbb{X}) \to L^1(\mathbb{X})$ *is defined by the unique linear extension of*

$$\mathcal{P}^t p(x) = \int_{\mathbb{X}} p^t(y,x)\,p(y)\,\mathrm{d}y$$

*to* $L^1(\mathbb{X})$*.*

*(b) The* Perron–Frobenius operator $\mathcal{T}^t\colon L^1_\mu(\mathbb{X}) \to L^1_\mu(\mathbb{X})$ with respect to the equilibrium density *is defined by the unique linear extension of*

$$\mathcal{T}^t u(x) = \int_{\mathbb{X}} \frac{\varrho(y)}{\varrho(x)}\,p^t(y,x)\,u(y)\,\mathrm{d}y$$

*to* $L^1_\mu(\mathbb{X})$*.*

*(c) The* Koopman operator $\mathcal{K}^t\colon L^\infty(\mathbb{X}) \to L^\infty(\mathbb{X})$ *is defined by*

$$\mathcal{K}^t f(x) = \int_{\mathbb{X}} p^t(x,y)\,f(y)\,\mathrm{d}y = \mathsf{E}[f(\mathbf{X}_t) \mid \mathbf{X}_0 = x]. \tag{A.1}$$

*All these are well-defined non-expanding operators on the respective spaces.*

[1] We denote by $L^q$ the space (equivalence class) of $q$-integrable functions with respect to the Lebesgue measure. $L^q_\nu$ denotes the same space of function, now integrable with respect to the measure $\nu$.

The equilibrium density $\varrho$ satisfies $\mathcal{P}^t\varrho = \varrho$, that is, $\varrho$ is an eigenfunction of $\mathcal{P}^t$ with associated eigenvalue $\lambda_0 = 1$. The definition of $\mathcal{T}^t$ relies on $\varrho$, we have $\varrho\,\mathcal{T}^t u = \mathcal{P}^t(u\varrho)$.

Instead of their natural domains from Definition A.2.2, all our transfer operators are considered on the following Hilbert spaces: $\mathcal{P}^t : L^2_{1/\mu}(\mathbb{X}) \to L^2_{1/\mu}(\mathbb{X})$, $\mathcal{T}^t : L^2_\mu(\mathbb{X}) \to L^2_\mu(\mathbb{X})$, and $\mathcal{K}^t : L^2_\mu(\mathbb{X}) \to L^2_\mu(\mathbb{X})$. They are still well-defined non-expansive operators on these spaces [3, 162, 93].

Furthermore, we will need the notion of reversibility for our considerations. Reversibility means that the process is statistically indistinguishable from its time-reversed counterpart.

**Definition A.2.3** (Reversibility)**.** *A system is said to be* reversible *if the detailed balance condition*

$$\varrho(x)\,p^t(x,y) = \varrho(y)\,p^t(y,x)$$

*is satisfied for all* $x, y \in \mathbb{X}$.

In what follows, we will assume that the system is reversible.

One prominent example for a class of SDEs satisfying uniqueness of the equilibrium density and reversibility is given by

$$\mathrm{d}\mathbf{X}_t = -\nabla V(\mathbf{X}_t)\,\mathrm{d}t + \sqrt{2\beta^{-1}}\,\mathrm{d}\mathbf{W}_t\,. \tag{A.2}$$

Here, $V$ is called the potential, $\beta$ is the non-dimensionalized inverse temperature, and $\mathbf{W}_t$ is a standard Wiener process. The process generated by (A.2) is ergodic and thus admits a unique positive equilibrium density, given by $\varrho(x) = \exp(-\beta V(x))/Z$, under mild growth conditions on the potential $V$ [119, 120]. Note that the subsequent considerations hold for all stochastic processes that satisfy reversibility and ergodicity with respect to a unique positive invariant measure and are *not* limited to the class of dynamical systems given by (A.2). See [171] for a discussion of a variety of stochastic dynamical systems that have been considered in this context.

As a result of the detailed balance condition, the Koopman operator $\mathcal{K}^t$ and the Perron–Frobenius operator with respect to the equilibrium density $\mathcal{T}^t$ become identical and we obtain

$$\big\langle \mathcal{P}^t f,\, g\big\rangle_{1/\mu} = \big\langle f,\, \mathcal{P}^t g\big\rangle_{1/\mu} \quad \text{and} \quad \big\langle \mathcal{T}^t f,\, g\big\rangle_{\mu} = \big\langle f,\, \mathcal{T}^t g\big\rangle_{\mu}\,,$$

i.e. all the transfer operators become self-adjoint on the respective Hilbert spaces from above. Here $\langle\cdot,\cdot\rangle_\mu$ and $\langle\cdot,\cdot\rangle_{1/\mu}$ denote the natural scalar products on the weighted spaces $L^2_\mu$ and $L^2_{1/\mu}$, respectively.

### A.2.2 Spectral decomposition

Due to the self-adjointness, the eigenvalues $\lambda_i^t$ of $\mathcal{P}^t$ and $\mathcal{T}^t$ are real-valued and the eigenfunctions form an orthogonal basis with respect to $\langle\cdot,\,\cdot\rangle_{1/\mu}$ and $\langle\cdot,\,\cdot\rangle_\mu$, respectively. In what follows, we assume that the spectrum of $\mathcal{T}^t$ is purely discrete given by (infinitely many) isolated eigenvalues. This assumption is made for the sake of simplicity. It is

actually not required for the rest of our considerations; it would be sufficient to assume that the spectral radius $R$ of the essential spectrum of $\mathcal{T}^t$ is strictly smaller than 1, and some isolated eigenvalues of modulus larger than $R$ exist. It has been shown that this condition is satisfied for a large class of metastable dynamical systems, see [171, Sec. 5.3] for details. For example, the process generated by (A.2) has purely discrete spectrum under mild growth and regularity assumptions on the potential $V$.

Under this condition, ergodicity implies that the dominant eigenvalue $\lambda_0$ is the only eigenvalue with absolute value 1 and we can thus order the eigenvalues so that

$$1 = \lambda_0^t > \lambda_1^t \geq \lambda_2^t \geq \dots.$$

The eigenfunction of $\mathcal{T}^t$ corresponding to $\lambda_0 = 1$ is the constant function $\varphi_0 = \mathbb{1}_{\mathbb{X}}$. Let $\varphi_i$ be the normalized eigenfunctions of $\mathcal{T}^t$, i.e. $\langle \varphi_i, \varphi_j \rangle_\mu = \delta_{ij}$, then any function $f \in L^2_\mu(\mathbb{X})$ can be written in terms of the eigenfunctions as $f = \sum_{i=0}^{\infty} \langle f, \varphi_i \rangle_\mu \varphi_i$. Applying $\mathcal{T}^t$ thus results in

$$\mathcal{T}^t f = \sum_{i=0}^{\infty} \lambda_i^t \langle f, \varphi_i \rangle_\mu \varphi_i.$$

For more details, we refer to [93] and references therein.

### A.2.3 Implied time scales

For some $d \in \mathbb{N}$, we call the $d+1$ dominant eigenvalues $\lambda_0^t, \dots, \lambda_d^t$ of $\mathcal{T}^t$ the *dominant spectrum* of $\mathcal{T}^t$, i.e.

$$\sigma_{\text{dom}}(\mathcal{T}^t) := \{\lambda_0^t, \dots, \lambda_d^t\}.$$

Usually, $d$ is chosen in such a way that there is a *spectral gap* after $\lambda_d^t$, i.e. $1 - \lambda_d^t \ll \lambda_d^t - \lambda_{d+1}^t$. The *(implied) time scales* on which the associated dominant eigenfunctions decay are given by

$$t_i = -t / \log(\lambda_i^t). \tag{A.3}$$

If $\mathcal{T}^t$ is a semigroup of operators, then there are $\kappa_i \leq 0$ with $\lambda_i^t = \exp(\kappa_i t)$ such that $t_i = -\kappa_i^{-1}$ holds. Assuming there is a spectral gap, the dominant time scales satisfy $t_1 \geq \dots \geq t_d \gg t_{d+1}$. These are the time scales of the *slow* dynamical processes, also called *rare events*, which are of primary interest in applications. The other, *fast* processes are regarded as fluctuations around the relative equilibria (or *metastable states*) between which the relevant slow processes travel.

## A.3 Projected transfer operators and reaction coordinates

The purpose of dimension reduction in molecular dynamics is to find a reduced dynamical model that captures the dominant time scales of the system correctly while keeping the model as simple as possible. In this section, we will introduce two different projections and the corresponding projected transfer operators. The goal is to find suitable projections onto the slow processes.

### A.3.1 Galerkin projections and Markov state models

One frequently used approach to obtain a reduced model is *Markov state modeling*. The goal is to find a model that is as simple as possible and yet correctly reproduces the dominant time scales. Given a fixed $t > 0$, most authors [135, 145] refer to a Markov state model (MSM) as a matrix $T^t \in \mathbb{R}^{(d+1)\times(d+1)}$ such that

$$\sigma_{\mathrm{dom}}(\mathcal{T}^t) \approx \sigma_{\mathrm{dom}}(T^t), \tag{A.4}$$

and it has been studied in detail under which condition this can be achieved [47, 160].

There are different ways of constructing an MSM, maybe the most intuitive one is also the simplest: Let the entries of $T^t$ be the transition rates between metastable sets. A typical molecular system with $d$ dominant time scales will have $d+1$ metastable sets $\mathbb{C}_1, \dots, \mathbb{C}_{d+1}$ (also called *cores*) and its dynamics is characterized by transitions between these sets and fluctuations inside the sets (see Figure A.1 for an illustration). Since the fluctuations are on faster time scales, we neglect them by setting [169]

$$T^t_{\mathrm{core},ij} = \mathsf{P}_\mu[\mathbf{X}_t \in \mathbb{C}_j \,|\, \mathbf{X}_0 \in \mathbb{C}_i]\,, \tag{A.5}$$

where $\mathsf{P}_\mu$ denotes the probability measure conditioned to the initial condition $\mathbf{X}_0$ being distributed according to $\mu$. Thus, $T^t_{\mathrm{core},ij}$ is the probability that the process in equilibrium jumps to the metastable set $\mathbb{C}_j$ in time $t$, given that it started in the metastable set $\mathbb{C}_i$. Note that (A.5) can be equivalently rewritten as

$$T^t_{\mathrm{core},ij} = \frac{\langle \mathcal{T}^t \mathbb{1}_{\mathbb{C}_i},\, \mathbb{1}_{\mathbb{C}_j} \rangle_\mu}{\langle \mathbb{1}_{\mathbb{C}_i},\, \mathbb{1}_{\mathbb{C}_i} \rangle_\mu}\,, \tag{A.6}$$

where $\mathbb{1}_{\mathbb{C}_i}$ is the characteristic function of the set $\mathbb{C}_i$.

Equation (A.6) readily suggests that $T^t_{\mathrm{core}}$ is a projection of the transfer operator $\mathcal{T}^t$, namely its *Galerkin projection* onto the space spanned by the characteristic functions $\mathbb{1}_{\mathbb{C}_1}, \dots, \mathbb{1}_{\mathbb{C}_{d+1}}$ [169].

**Definition A.3.1** (Galerkin projection)**.** *Given a set of basis functions $\psi_1, \dots, \psi_m \in L^2_\mu(\mathbb{X})$, let $\mathbb{V} := \mathrm{span}\{\psi_1, \dots, \psi_m\}$ and $\psi := (\psi_1, \dots, \psi_m)^\intercal$. The projection to $\mathbb{V}$ or, equivalently, to $\psi$, $\Pi_\mathbb{V} = \Pi_\psi \colon L^2_\mu(\mathbb{X}) \to \mathbb{V}$ is defined as*

$$\langle \Pi_\psi f - f,\, g \rangle_\mu = 0 \qquad \forall\, f \in L^2_\mu(\mathbb{X}),\, \forall\, g \in \mathbb{V}\,.$$

*The residual projection is given by $\Pi^\perp_\psi = \mathrm{Id} - \Pi_\psi$, where $\mathrm{Id}$ is the identity. The Galerkin projection of $\mathcal{T}^t$ to $\mathbb{V}$ is given by the linear operator $T^t \colon \mathbb{V} \to \mathbb{V}$ satisfying*

$$\langle \mathcal{T}^t f - T^t f,\, g \rangle_\mu = 0 \qquad \forall\, f, g \in \mathbb{V}\,.$$

Equivalently, $T^t = \Pi_\psi \mathcal{T}^t$. We also denote the extension of $T^t$ to the whole $L^2_\mu(\mathbb{X})$, given by $\Pi_\psi \mathcal{T}^t \Pi_\psi$, by $T^t$. Furthermore, we denote the matrix representation of $T^t$ with respect

to the basis $(\psi_0, \dots, \psi_d)$ by $T^t$ as well. Either it will be clear from the context which of the objects $T^t$ is meant or it will not matter; e.g., the dominant spectrum is the same for all of them.

We see that $T_{\text{core}}$ is the matrix representation of the Galerkin projection with respect to the basis functions $\langle \mathbb{1}_{\mathbb{C}_i}, \mathbb{1}_{\mathbb{C}_i} \rangle_\mu^{-1} \mathbb{1}_{\mathbb{C}_i}$, $i = 1, \dots, d+1$. More general MSMs can be built by Galerkin projections of the transfer operator to spaces spanned by other — not necessarily piecewise constant — basis functions [191, 170, 193, 92, 93, 150, 134]. However, in some of these methods, one also often loses the interpretation of the entries of the matrix $T^t$ as probabilities.

Ultimately, the best MSM in terms of approximation quality in (A.4) is given by the Galerkin projection of $\mathcal{T}^t$ onto the space spanned by its dominant eigenfunctions $\varphi_0, \dots, \varphi_d$. This space is invariant under $\mathcal{T}^t$ since $\mathcal{T}^t \varphi_i = \lambda_i^t \varphi_i$ and the dominant eigenvalues (and hence the time scales) are the same for the MSM and for $\mathcal{T}^t$. Due to the curse of dimensionality, however, the computation of the eigenfunctions $\varphi_i$ is in general infeasible for high-dimensional problems.

**Remark A.3.2.** There are quantitative results assessing the error in (A.4) of the MSM in terms of the projection errors $\|\Pi_\psi^\perp \varphi_i\|_{L^2_\mu}$, $i = 0, \dots, d$, cf. [171, Section 5.3]. One can obtain a weaker, but similar result from our Lemma A.3.5 in the next section.

### A.3.2 Coordinate projections and effective transfer operators

While the MSMs from above successfully reproduce the dominant time scales of the original system, they often discard all other information about the system, such as the transition paths between metastable sets. Minimal coordinates that describe these transitions are called *reaction coordinates* and reducing the dynamics onto these coordinates yields *effective dynamics* [109, 199]. The goal of the previous section — namely to retain the dominant time scales of the original dynamics in a reduced model — can now be reformulated for this lower-dimensional effective dynamics or, equivalently, for its (effective) transfer operator.

Let $\xi \colon \mathbb{X} \to \mathbb{R}^k$ be a $C^1$ function, where $k \leq n$. Let $\mathbb{L}_z = \{x \in \mathbb{X} \mid \xi(x) = z\}$ be the $z$-level set of $\xi$. The so-called *coarea formula* [58, Section 3.2], which can be considered as a nonlinear variant of Fubini's theorem, splits integrals over $\mathbb{X}$ into consecutive integrals over level sets of $\xi$ and then over the range of $\xi$. For $f \in L^2_\mu(\mathbb{X})$, we have[2]

$$\int_{\mathbb{X}} f(x)\, d\mu(x) = \int_{\xi(\mathbb{X})} \int_{\mathbb{L}_z} f(x') \varrho(x') \det\left(\nabla\xi(x')^\intercal \nabla\xi(x')\right)^{-1/2} d\sigma_z(x')\, dz\,, \tag{A.7}$$

where $z = \xi(x)$ and $\sigma_z$ is the surface measure on $\mathbb{L}_z$. The *coordinate projection*, defined next, averages a given function along the level sets of a coordinate function $\xi$.

[2] The coarea formula holds for $L^1$ functions, but $L^2_\mu \subset L^1_\mu$, since $\mu$ is a probability measure (i.e., it is finite).

**Definition A.3.3** (Coordinate projection)**.** *For $f \in L^2_\mu(\mathbb{X})$, we define*

$$P_\xi f(x) = \int_{\mathbb{L}_z} f(x')\, d\mu_z(x') \tag{A.8}$$

$$= \frac{1}{\Gamma(z)} \int_{\mathbb{L}_z} f(x')\varrho(x') \det(\nabla\xi(x')^\intercal \nabla\xi(x'))^{-1/2}\, d\sigma_z(x'), \tag{A.9}$$

*where $\mu_z$ is a probability measure on $\mathbb{L}_z$ with density $\frac{\varrho}{\Gamma(z)} \det(\nabla\xi^\intercal\nabla\xi)^{-1/2}$ with respect to $\sigma_z$. Here, $\Gamma(z)$ is just the normalization constant so that $\mu_z$ becomes a probability measure. The residual projection is given by $P_\xi^\perp = \mathrm{Id} - P_\xi$.*

To get a better feeling for the action of $P_\xi$, note that $P_\xi f(x)$ is the expectation of $f(\mathbf{x}')$ with respect to $\mu$ conditional to $\xi(\mathbf{x}') = \xi(x)$, i.e.

$$P_\xi f(x) = \mathsf{E}_\mu \left[ f(\mathbf{x}') \,\middle|\, \xi(\mathbf{x}') = \xi(x) \right] .$$

Or, in other words, $\mu_z$ is the marginal of $\mu$ conditional to $\xi(x) = z$. Note, in particular, that $P_\xi f$ is itself a function on $\mathbb{X}$, but it is constant on the level sets of $\xi$, and thus let us set $\widehat{P_\xi f}(\xi(x)) = P_\xi f(x)$ for $x \in \mathbb{L}_{\xi(x)}$. It follows from the coarea formula (A.7) and (A.9) that

$$\int_{\mathbb{X}} f(x)\, d\mu(x) = \int_{\xi(\mathbb{X})} \Gamma(z) \widehat{P_\xi f}(z)\, dz\,. \tag{A.10}$$

Next, we state some properties of the coordinate projection.

**Proposition A.3.4.** *The coordinate projection has the following properties.*

*(a) $P_\xi$ is a linear projection, i.e. $P_\xi^2 = P_\xi$.*

*(b) $P_\xi$ is self-adjoint with respect to $\langle \cdot, \cdot \rangle_\mu$.*

*(c) $P_\xi \colon L^2_\mu(\mathbb{X}) \to L^2_\mu(\mathbb{X})$ is orthogonal, hence non-expansive, i.e. $\|P_\xi f\|_{L^2_\mu} \le \|f\|_{L^2_\mu}$.*

*Proof.* See Appendix A.A. □

We use the coordinate projection to describe the dynamics-induced propagation of reduced distributions with respect to the variable $\xi$. To this end, we define the *effective transfer operator* $\mathcal{T}_\xi^t \colon L^2_\mu(\mathbb{X}) \to L^2_\mu(\mathbb{X})$ by

$$\mathcal{T}_\xi^t = P_\xi \mathcal{T}^t P_\xi. \tag{A.11}$$

We immediately obtain from the self-adjointness of $\mathcal{T}^t$ (see Section A.2) and Proposition A.3.4 (b) that $\mathcal{T}_\xi^t$ is a self-adjoint operator on $L^2_\mu(\mathbb{X})$. Moreover, $\|\mathcal{T}^t\|_{L^2_\mu} \le 1$ and Proposition A.3.4 (c) imply that $\|\mathcal{T}_\xi^t\|_{L^2_\mu} \le 1$. Thus, the spectrum of the effective transfer operator lies in the interval $[-1, 1]$, too.

Returning to the purpose of these constructions, we call $\xi$ a *good reaction coordinate* if

$$\sigma_{\mathrm{dom}}(\mathcal{T}^t) \approx \sigma_{\mathrm{dom}}(\mathcal{T}_\xi^t). \tag{A.12}$$

While the previously introduced Markov state model $T^t$ obtained by the Galerkin projection was approximating the dominant spectrum of the original transfer operator by a finite-dimensional operator (i.e. a matrix), the effective transfer operator still acts on an infinite-dimensional space. The reduction lies in the fact that $\mathcal{T}^t$ operates on functions over $\mathbb{X} \subseteq \mathbb{R}^n$, but the effective transfer operator $\mathcal{T}_\xi^t$ operates *essentially* on functions over $\xi(\mathbb{X}) \subset \mathbb{R}^k$, although we embed those into $\mathbb{X}$ through the level sets of $\xi$.

As mentioned above, a Galerkin projection of the transfer operator onto its dominant eigenfunctions is a perfect MSM. In the same vein, we ask here how we can characterize a good reaction coordinate. We can make use of the following general result.

**Lemma A.3.5.** *Let $\mathbb{H}$ be a Hilbert space with scalar product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$, let $Q : \mathbb{H} \to \mathbb{H}$ be some orthogonal projection on a linear subspace of $\mathbb{H}$, with $Q^\perp = \mathrm{Id} - Q$. Let $T : \mathbb{H} \to \mathbb{H}$ be a self-adjoint non-expansive linear operator, and $u$ with $\|u\| = 1$ its eigenvector, i.e., $Tu = \lambda u$ for some $\lambda \in \mathbb{R}$. If $\|Q^\perp u\| < \varepsilon$, then $T_Q := QTQ$ has an eigenvalue $\lambda_Q \in \mathbb{R}$ with $|\lambda - \lambda_Q| < \varepsilon/\sqrt{1-\varepsilon^2}$.*

*Proof.* Using $Q = \mathrm{Id} - Q^\perp$, we have

$$T_Q Q u = QT \underbrace{QQ}_{=Q} u = QTu - \underbrace{QTQ^\perp u}_{=:-\zeta} = \lambda Q u + \zeta,$$

where $\|\zeta\| \leq \|Q^\perp u\| < \varepsilon$ since $Q$ and $T$ are non-expanding. Thus, $u' := Qu/\|Qu\|$ satisfies $T_Q u' = \lambda u' + \zeta/\|Qu\|$, and the orthogonality of $Q$ gives $\|Qu\| > \sqrt{1-\varepsilon^2}$. Now, any orthogonal projection is self-adjoint, as is shown in the proof of Proposition A.3.4, hence the operator $QTQ$ is self-adjoint, too, and thus normal. From the theory of pseudospectra for normal operators [185, Theorems 2.1, 2.2, and §4], we know that if $\|T_Q u' - \lambda u'\| < \varepsilon/\sqrt{1-\varepsilon^2}$, then $T_Q$ has an eigenvalue $\lambda_Q \in \mathbb{R}$ in the $\varepsilon/\sqrt{1-\varepsilon^2}$-neighborhood of $\lambda$. □

With $\mathbb{H} = L^2_\mu$, $Q = P_\xi$, and $T = \mathcal{T}^t$ we immediately obtain the following result.

**Corollary A.3.6.** *As before, let $\lambda_i^t$ and $\varphi_i$, $i = 0, \ldots, d$, denote the dominant eigenvalues and eigenfunctions of $\mathcal{T}^t$, respectively. For any given $i$, if $\|P_\xi^\perp \varphi_i\|_{L^2_\mu} < \varepsilon$, then there is an eigenvalue $\tilde{\lambda}_i^t$ of $\mathcal{T}_\xi^t$ with $|\lambda_i^t - \tilde{\lambda}_i^t| < \varepsilon/\sqrt{1-\varepsilon^2}$.*

Corollary A.3.6 implies that if the projection error of *all dominant* eigenfunctions is small, then $\xi$ is a good reaction coordinate in the sense of (A.12). Very similar results are available for approximation of the eigenvalues of the infinitesimal generator of the Fokker–Planck equation associated with the transfer operator if the dynamical system under consideration is continuous in time [200].

Under which conditions is the projection error small? Let us consider the case where there are $\tilde{\varphi}_i : \mathbb{R}^k \to \mathbb{R}$, $i = 1, \ldots, d$, such that $\varphi_i(x) = \tilde{\varphi}_i(\xi(x))$. We then say that $\varphi_i$ *is a function of* $\xi$ or that $\xi$ *parametrizes* $\varphi_i$. If $\xi$ parametrizes $\varphi_i$ perfectly, the projection error obviously vanishes. Thus, trivially, by choosing $\xi = \varphi = (\varphi_1, \ldots, \varphi_d)^\intercal$, we obtain a perfect reaction coordinate since with $\tilde{\varphi}_i(z) := z_i$ with $\varphi_i = \tilde{\varphi}_i \circ \xi$. However, the eigenfunctions

are *global* objects, i.e., their computation is prohibitive in high dimensions. Since we are aiming at computing a reaction coordinate, we have to answer the question of whether there is a reaction coordinate $\xi$ that can be evaluated based on local computations only while it parametrizes the dominant eigenfunctions of $\mathcal{T}^t$ well enough such that it leads to a small projection error. We will see next that this question can be answered by utilizing a common property of most metastable systems: The transitions between the metastable sets happen along so-called *reaction pathways*, which imply the existence of *transition manifolds* in the space of transition densities. A *suitable* parametrization of this manifold results in a parametrization of the dominant eigenfunctions with a small error.

## A.4 Identifying good reaction coordinates

The goal is now to find a reaction coordinate $\xi$ that is as low-dimensional as possible and results in a good projected transfer operator in the sense of (A.12). As we saw in the previous section, the condition $\|P_\xi^\perp \varphi_i\|_{L^2_\mu} \approx 0$ is sufficient. Thus, the idea to numerically seek $\xi$ that parametrizes the dominant eigenfunctions of $\mathcal{T}^t$ in the $\|\cdot\|_{L^2_\mu}$-norm seems natural since this would lead to small projection error $\|P_\xi^\perp \varphi_i\|_{L^2_\mu}$.

In fact, eigenfunctions of transfer operators have been used before to compute reduced dynamics and reaction coordinates: In [64], methods to decompose multiscale systems into fast and slow processes and to project the dynamics onto these subprocesses based on eigenfunctions of the Koopman operator $\mathcal{K}^t$ are proposed. In [122], the dominant eigenfunctions of the transfer operator $\mathcal{T}^t$, which due to the assumed reversibility of the system is identical to $\mathcal{K}^t$, are shown to be good reaction coordinates. Also, commitor functions (introduced in Appendix A.B), which are closely related to the dominant eigenfunctions, have been used as reaction coordinates in [51, 114].

However, we propose a fundamentally different path in defining and finding reaction coordinates, as working with dominant eigenfunctions has two major disadvantages:

1. The eigenproblem is *global*. Thus if we wish to learn the value of an eigenfunction $\varphi_i$ at only one location $x \in \mathbb{X}$, we need an approximation of the transfer operator $\mathcal{T}_t$ that has to be accurate on all of $\mathbb{X}$. The computational effort to construct such an approximation grows exponential with $\dim(\mathbb{X})$, this is the *curse of dimensionality*. There have been attempts to mitigate this [191, 88, 192], but we aim to circumvent this problem entirely. Given two points $x, y \in \mathbb{X}$, we will decide whether $\xi(x)$ is close to $\xi(y)$ or not by using only local computations around $x$ and $y$ (i.e. samples from the transition densities $p^t(x, \cdot)$ and $p^t(y, \cdot)$ for moderate $t$).

2. The number of dominant eigenfunctions $(d + 1)$ equals the number of metastable states, and this number can be much larger than the dimension of the transition manifold. This fact is illustrated in Example A.4.1 below.

**Example A.4.1.** Let us consider a diffusion process of the form (A.2) with the circular multi-well potential shown in Figure A.2. Choosing a temperature that is not high

enough for the central potential barrier to be overcome easily, transitions between the wells typically happen in the vicinity of a one-dimensional reaction pathway, the unit circle. The number of dominant eigenfunctions, however, corresponds to the number of wells. Nevertheless, projecting the system onto the unit circle would retain the dominant time scales of the system, cf. Section A.5. △

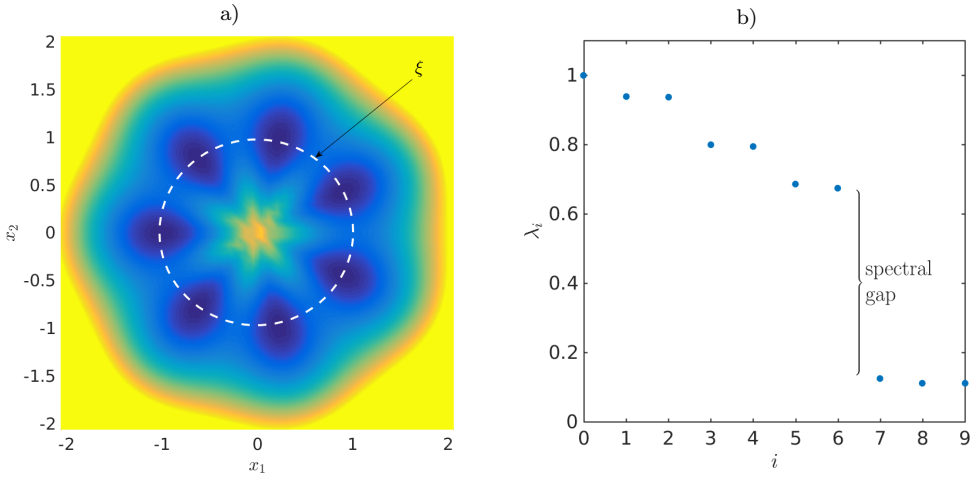



**Figure A.2:** a) Potential with seven wells and thus seven dominant eigenvalues, but only a one-dimensional reaction coordinate. The reaction pathway is marked by a dashed white line. b) Dominant eigenvalues of $\mathcal{T}^t$ for $t = 0.1$ and $\beta = 0.5$. The spectral gap is clearly visible.

### A.4.1 Parametrization of dominant eigenfunctions

If the $(d+1)$ dominant eigenfunctions do not depend fully on the phase space $\mathbb{X}$, a lower-dimensional and ultimately easier to find reaction coordinate suffices for keeping the eigenvalue approximation error (A.12) small. It is easy to see that if there exists a function $\xi\colon \mathbb{X} \to \mathbb{R}^k$ for some $k$ so that the eigenfunctions $\varphi$ are constant on the level sets of $\xi$, i.e., there exist functions $\tilde{\varphi}_i\colon \mathbb{R}^k \to \mathbb{R}$, $i = 1, \dots, d$ such that $\varphi_i = \tilde{\varphi}_i \circ \xi$, then the projection error $\|P_\xi^\perp \varphi_i\|_{L^2_\mu}$ is zero. A quantitative generalization of this is the statement that if the eigenfunctions $\varphi_i$ are *almost constant* on level sets of a $\xi$, then the projection error is small.

**Lemma A.4.2.** *Assume that there exists a function $\xi\colon \mathbb{X} \to \mathbb{R}^k$ for some $k$ and functions $\tilde{\varphi}_i\colon \mathbb{R}^k \to \mathbb{R}$, $i = 1, \dots, d$, with*

$$|\varphi_i(x) - \tilde{\varphi}_i(\xi(x))| \leq \varepsilon \quad \forall\, x \in \mathbb{X}. \tag{A.13}$$

*Then $\|P_\xi^\perp \varphi_i\|_{L^2_\mu} \leq 2\varepsilon$.*

*Proof.* Assuming (A.13) holds, there exists a function $c_i\colon \mathbb{R} \to \mathbb{R}$ with $c_i(x) \leq 1\ \forall x \in \mathbb{X}$ so that

$$\varphi_i(x) = \tilde{\varphi}_i(\xi(x)) + c_i(x)\varepsilon\,.$$

Thus, we have

$$\begin{aligned} P_\xi \varphi_i(x) &= \int_{\mathbb{L}_{\xi(x)}} \Big(\tilde{\varphi}_i\big(\xi(x')\big) + c_i(x')\varepsilon\Big) d\mu_{\xi(x)}(x') \\ &= \tilde{\varphi}_i\big(\xi(x)\big) + \varepsilon \int_{\mathbb{L}_{\xi(x)}} c_i(x') d\mu_{\xi(x)}(x'). \end{aligned}$$

For the projection error, we then obtain

$$\begin{aligned} \|P_\xi \varphi_i - \varphi_i\|_{L^2_\mu} &\leq \|P_\xi \varphi_i - \tilde{\varphi}_i \circ \xi\|_{L^2_\mu} + \|\tilde{\varphi}_i \circ \xi - \varphi_i\|_{L^2_\mu} \\ &\leq 2\varepsilon. \end{aligned}$$

□

**Remark A.4.3.** From the proof we see that the pointwise condition (A.13) can be replaced by the much weaker condition

$$\int_{\mathbb{L}_z} \big|\varphi_i(x') - \tilde{\varphi}_i(\xi(x'))\big|\, d\mu_z(x') \leq \varepsilon,$$

for all level sets $\mathbb{L}_z$ of $\xi$.

From here on, we address the following two central questions:

(Q1) *In which dynamical situations can we expect to find low-dimensional reaction coordinates?*

(Q2) *How can we computationally exploit the properties of the dynamics to obtain reaction coordinates?*

Let us start with the first question. We will address the second question in Section A.4.2 and Section A.4.3. Experience shows [53, 155, 54, 171] that transitions between metastable states tend to happen along so-called *reaction pathways*, which is the low-dimensional dynamical backbone in the high-dimensional state space, connecting the metastable states via saddle points of the potential $V$ [60].

From now on, we observe the system at an intermediate time scale $t_{\text{slow}} \gg t \gg t_{\text{fast}}$ (where $t_{\text{slow}}$ and $t_{\text{fast}}$ are the implied time scales $t_d$, $t_{d+1}$ from Section A.2.3) and thus assume that the process $\mathbf{X}_t$ has already left the transition region (if it started there), equilibrated to a quasi-stationary distribution inside some metastable wells, but has not had enough time to equilibrate *globally*. At this time scale, starting in some $x \in \mathbb{X}$, the transition density $p^t(x, \cdot)$ is observed to approximately depend only on progress along these reaction paths; see Figure A.3 for an illustration. This means that the density $p^t(x, \cdot)$ on the fiber perpendicular to the transition pathway is approximately the same as

$p^t(x^*, \cdot)$ for some $x^*$ *on* the transition pathway. As this pathway is low-dimensional, this means that the image $\overline{\mathcal{Q}}(\mathbb{X})$ of the map

$$\overline{\mathcal{Q}}(x) := p^t(x, \cdot)$$

is almost a low-dimensional manifold in $L^1(\mathbb{X})$.

The existence of this low-dimensional structure in the space of probability densities is exactly the assumption we need to ensure that the dominant eigenfunctions are low-dimensionally parametrizable, and thus that a low-dimensional reaction coordinate $\xi$ exists. This assumption is made precise in Definition A.4.4. To summarize, we will see that $\xi$ is a good reaction coordinate if $p^t(x, \cdot) \approx p^t(y, \cdot)$ for $\xi(x) = \xi(y)$.

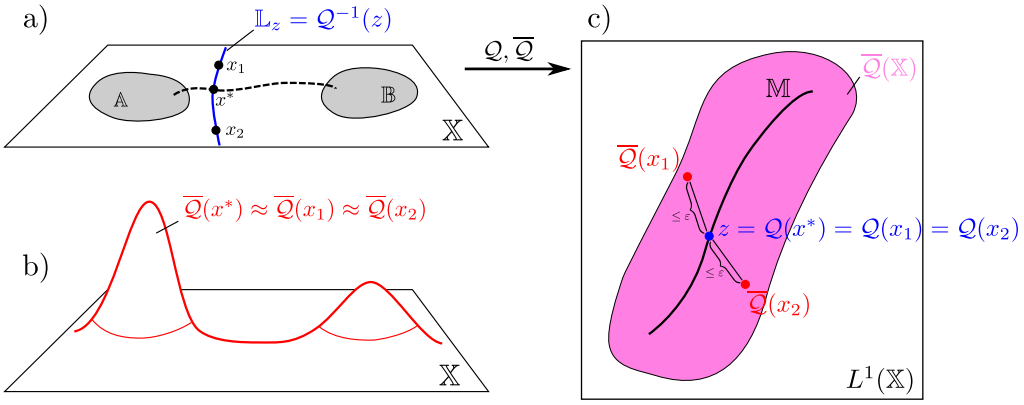



**Figure A.3:** a) and b) The transition densities $\overline{\mathcal{Q}}(x_1)$ and $\overline{\mathcal{Q}}(x_2)$ are "similar" to $\overline{\mathcal{Q}}(x^*)$ for some $x^*$ on the transition path (dashed line) that connects the metastable sets $\mathbb{A}$ and $\mathbb{B}$. c) The mapping $\mathcal{Q}$ can be thought of as mapping all points that are "similar" under $\overline{\mathcal{Q}}$ to the same point in $L^1(\mathbb{X})$. The image of $\mathcal{Q}$ thus forms a $r$-dimensional manifold in $L^1(\mathbb{X})$.

**Definition A.4.4** ($(\varepsilon, r)$-reducibility and transition manifold)**.** *We call the process* $\mathbf{X}_t$ $(\varepsilon, r)$-reducible*, if there exists a smooth closed* $r$*-dimensional manifold* $\mathbb{M} \subset L^2_{1/\mu} \subset L^1(\mathbb{X})$ *such that for* $t_{fast} \ll t \ll t_{slow}$ *and all* $x \in \mathbb{X}$

$$\min_{f \in \mathbb{M}} \|f - p^t(x, \cdot)\|_{L^2_{1/\mu}} \leq \varepsilon \tag{A.14}$$

*holds. We call* $\mathbb{M}$ *the* transition manifold *and the map* $\mathcal{Q}\colon \mathbb{X} \to \mathbb{M}$*,*

$$\mathcal{Q}(x) := \arg\min_{f \in \mathbb{M}} \|p^t(x, \cdot) - f\|_{L^2_{1/\mu}} \tag{A.15}$$

*the* mapping onto the transition manifold*. We can set* $\mathbb{M} = \mathrm{cl}(\mathcal{Q}(\mathbb{X}))$*, where* $\mathrm{cl}(\mathbb{Y})$ *denotes the closure of the set* $\mathbb{Y}$*.*[3]

[3] If it is necessary to break ties in (A.15), we can do so by taking any of the minimizers. The mapping $x \mapsto p^t(x, \cdot)$ can be shown to be smooth [9, Theorem C.1], hence $\mathcal{Q}(\mathbb{X})$ is a smooth manifold satisfying (A.14).

**Remark A.4.5.** While it is natural to motivate $(\varepsilon, r)$-reducibility by the existence of reaction pathways in phase space, it is not strictly necessary. There exist stochastic systems without low-dimensional reaction pathways whose densities still quickly converge to a transition manifold in $L^1$. Future work includes the identification of necessary and sufficient conditions for the existence of transition manifolds (see the first point in the conclusions). We also further elaborate on the connection between reaction pathways and transition manifolds in Appendix A.B.

**Remark A.4.6.** We recall from Section A.2 that the Perron–Frobenius operator $\mathcal{P}^t$ is also naturally defined on the space $L^2_{1/\mu}$ [162]. Further, with the Dirac distribution centered in $x \in \mathbb{X}$, denoted by $\delta_x$, we formally have $p^t(x,\cdot) = \mathcal{P}^t\delta_x$. Hence, the choice of norm in Definition A.4.4 is natural. It should also be noted that since $\mu$ is a probability measure, the Hölder inequality yields $\|f\|_{L^1_\mu} \leq \|f\|_{L^2_\mu}$. Using this we have

$$\|f\|_{L^1} = \|f/\varrho\|_{L^1_\mu} \leq \|f/\varrho\|_{L^2_\mu} = \|f\|_{L^2_{1/\mu}} ,$$

which shows that if $p^t(x,\cdot)$ and $p^t(y,\cdot)$ are close in the $L^2_{1/\mu}$ norm, they are also close in the $L^1$ norm. We require the closeness of the respective $p^t(x,\cdot)$ in the $L^2_{1/\mu}$ norm for our theoretical considerations below, but otherwise we will think of them as functions in $L^1$.

Note that we only need to evolve the system at hand for a moderate time $t \ll t_{\text{slow}}$, which has to be merely sufficiently large to damp out the fast fluctuations in the metastable states. This will be an important point later, allowing for numerical tractability.

Next, we show that $(\varepsilon, r)$-reducibility implies that dominant eigenfunctions are almost constant on the level sets of $\mathcal{Q}$.

**Lemma A.4.7.** *If $\mathbf{X}_t$ is $(\varepsilon, r)$-reducible, then for an eigenfunction $\varphi_i$ of $\mathcal{T}^t$ with $\|\varphi_i\|_{L^2_\mu} = 1$ and points $x, y \in \mathbb{X}$ with $\mathcal{Q}(x) = \mathcal{Q}(y)$ we have*

$$|\varphi_i(x) - \varphi_i(y)| \leq \frac{2\varepsilon}{|\lambda_i|} .$$

*Proof.* First note that for the transition densities $p^t(x,\cdot)$, $p^t(y,\cdot)$ it holds that

$$\begin{aligned} \|p^t(x,\cdot) - p^t(y,\cdot)\|_{L^2_{1/\mu}} &\leq \|p^t(x,\cdot) - \mathcal{Q}(x)\|_{L^2_{1/\mu}} + \|\mathcal{Q}(x) - p^t(y,\cdot)\|_{L^2_{1/\mu}} \\ &= \|p^t(x,\cdot) - \mathcal{Q}(x)\|_{L^2_{1/\mu}} + \|\mathcal{Q}(y) - p^t(y,\cdot)\|_{L^2_{1/\mu}} \leq 2\varepsilon . \end{aligned} \tag{A.16}$$

With this we can show the assertion:

$$\lambda_i\varphi_i(x) = \mathcal{T}^t\varphi_i(x) = \mathcal{K}^t\varphi_i(x) = \int_{\mathbb{X}} \varphi_i(x')p^t(x,x')\,dx'.$$

Applying (A.16), for some function $e \in L^2_{1/\mu}(\mathbb{X})$ with $\|e\|_{L^2_{1/\mu}} \leq 2\varepsilon$, we get

$$\lambda_i\varphi_i(x) = \int_{\mathbb{X}} \varphi_i(x')\big(p^t(y,x') + e(x')\big)\,dx'$$

$$= \int_{\mathbb{X}} \varphi_i(x') p^t(y, x') dx' + \int_{\mathbb{X}} \varphi_i(x') \frac{e(x')}{\varrho(x')} \, d\mu(x')$$
$$= \lambda_i \varphi_i(y) + \int_{\mathbb{X}} \varphi_i(x') \frac{e(x')}{\varrho(x')} \, d\mu(x'),$$

where in the last equation, we again used that due to reversibility $\mathcal{K}^t = \mathcal{T}^t$ and that $\varphi_i$ is an eigenfunction. Thus for the difference, we have

$$|\varphi(x) - \varphi(y)| = \frac{1}{|\lambda_i|} \Big| \int_{\mathbb{X}} \varphi_i(x') \frac{e(x')}{\varrho(x')} \, d\mu(x') \Big|$$
$$\leq \frac{1}{|\lambda_i|} \underbrace{\|\varphi_i\|_{L^2_\mu}}_{=1} \underbrace{\|e/\varrho\|_{L^2_\mu}}_{=\|e\|_{L^2_{1/\mu}}} \leq \frac{2\varepsilon}{|\lambda_i|} . \qquad \square$$

Assuming that the eigenfunctions are normalized (which we do from now on), i.e., $\|\varphi_i\|_{L^2_\mu} = 1$, and that $\varepsilon$ is sufficiently small, Lemma A.4.7 implies that the dominant eigenfunctions (i.e., $|\lambda_i| \approx 1$) are almost constant on the level sets of $\mathcal{Q}$. This can now be used to show that the $\varphi_i$ are not fully dependent on $\mathbb{X}$, but only on the level sets of $\mathcal{Q}$ (up to a small error), in a sense similar to Lemma A.4.2.

**Corollary A.4.8.** *Let $\mathbf{X}_t$ be $(\varepsilon, r)$-reducible. Then there exists a function $\tilde{\varphi}_i \colon \mathbb{M} \to \mathbb{R}$ such that*

$$\big|\varphi_i(x) - \tilde{\varphi}_i\big(\mathcal{Q}(x)\big)\big| \leq \frac{\varepsilon}{|\lambda_i|} .$$

*Proof.* Fix $x \in \mathbb{X}$, and let $z = \mathcal{Q}(x)$. Define the function $\tilde{\varphi}_i$ by

$$\tilde{\varphi}_i(\mathcal{Q}(x)) := \frac{1}{2} \left( \inf_{\mathcal{Q}(y)=z} \varphi_i(y) + \sup_{\mathcal{Q}(y)=z} \varphi_i(y) \right) .$$

Since by Lemma A.4.7 it holds that $|\varphi_i(x) - \varphi_i(y)| \leq \frac{2\varepsilon}{|\lambda_i|}$ if $\mathcal{Q}(x) = \mathcal{Q}(y)$, we have that

$$\left| \sup_{\mathcal{Q}(y)=z} \varphi_i(y) - \inf_{\mathcal{Q}(y)=z} \varphi_i(y) \right| \leq \frac{2\varepsilon}{|\lambda_i|} ,$$

thus our choice of $\tilde{\varphi}_i$ gives

$$|\varphi_i(x) - \tilde{\varphi}_i(\mathcal{Q}(x))| \leq \frac{\varepsilon}{|\lambda_i|}. \qquad \square$$

### A.4.2 Embedding the transition manifold

In light of Corollary A.4.8, one could say that $\mathcal{Q}$ is an "$\mathbb{M}$-valued reaction coordinate". However, as we have no access to $\mathbb{M}$ so far, and a $\mathbb{R}^k$-valued reaction coordinate is more intuitive, we aim to obtain a more useful representation of the transition manifold through *embedding* it into a finite, possibly low-dimensional Euclidean space.

We will see that we are very free in the choice of the embedding mapping, even though the manifold $\mathbb{M}$ is not known explicitly (we only assumed that it exists). To achieve this, we will use an infinite-dimensional variant of the *weak Whitney embedding theorem* [161, 194], which, roughly speaking, states that "almost every bounded linear map from $L^1(\mathbb{X})$ to $\mathbb{R}^{2r+1}$ will be one-to-one on $\mathbb{M}$ and its image". We first specify what we mean by "almost every" in the context of bounded linear maps, following the notions of Sauer et al. [161].

**Definition A.4.9** (Prevalence)**.** *A Borel subset $\mathbb{S}$ of a normed linear space $\mathbb{V}$ is called* prevalent *if there is a finite-dimensional subspace $\mathbb{E}$ of $\mathbb{V}$ such that for each $v \in \mathbb{V}$, $v + e$ belongs to $\mathbb{S}$ for (Lebesgue) almost every $e$ in $\mathbb{E}$.*

As the infinite-dimensional embedding theorem from Hunt et al. [84] is applicable not only to smooth manifolds, but to arbitrary subsets $\mathbb{A} \subset \mathbb{V}$ of fractal dimension, it uses the concepts of *box covering dimension* $\dim_B(\mathbb{A})$ and *thickness exponent* $\tau(\mathbb{A})$ from fractal geometry. Intuitively, $\dim_B(\mathbb{A})$ describes the exponent of the growth rate in the number of boxes of decreasing side length that are needed to cover $\mathbb{A}$, and $\tau(\mathbb{A})$ describes how well $\mathbb{A}$ can be approximated using only finite-dimensional linear subspaces of $\mathbb{V}$. As these concepts coincide with the traditional measure of dimensionality in our setting, we will not go into detail here and point to [84] for a precise definition.

The general infinite-dimensional embedding theorem reads:

**Theorem A.4.10** ([84, Theorem 3.9])**.** *Let $\mathbb{V}$ be a Banach space and $\mathbb{A} \subset \mathbb{V}$ be a compact set with box-counting dimension $d$ and thickness exponent $\tau$. Let $k > 2d$ be an integer, and let $\alpha$ be a real number with*

$$0 < \alpha < \frac{k - 2d}{k(1 + \tau)} .$$

*Then for almost every (in the sense of prevalence) bounded linear function $\mathcal{E} : \mathbb{V} \to \mathbb{R}^k$ there exists $C > 0$ such that for all $x, y \in \mathbb{A}$,*

$$C\|\mathcal{E}(x) - \mathcal{E}(y)\|_2^\alpha \geq \|x - y\|_2 , \tag{A.17}$$

*where $\|\cdot\|_2$ denotes the Euclidean 2-norm.*

Note that (A.17) implies Hölder continuity of $\mathcal{E}^{-1}$ on $\mathcal{E}(\mathbb{A})$ and in particular that $\mathcal{E}$ is one-to-one on $\mathbb{A}$ and its image. Using that the box counting dimension of a smooth $r$-dimensional manifold $\mathbb{K}$ is simply $r$ and that the thickness exponent is bounded from above by the box-counting dimension, thus $0 \leq \tau(\mathbb{K}) \leq r$, see [84], we get the following infinite-dimensional embedding theorem for smooth manifolds.

**Corollary A.4.11.** *Let $\mathbb{V}$ be a Banach space, let $\mathbb{K} \subset \mathbb{V}$ be a smooth manifold of dimension $r$ and let $k > 2r$. Then almost every (in the sense of prevalence) bounded linear function $\mathcal{E} : \mathbb{V} \to \mathbb{R}^k$ is one-to-one on $\mathbb{K}$ and its image in $\mathbb{R}^k$.*

Thus, since the transition manifold $\mathbb{M}$ is assumed to be a smooth $r$-dimensional manifold in $L^1(\mathbb{X})$, an arbitrarily chosen bounded linear map $\mathcal{E}\colon L^1(\mathbb{X}) \to \mathbb{R}^{2r+1}$ can be assumed to be one-to-one on $\mathbb{M}$ and its image. In particular, $\mathcal{E}(\mathbb{M})$ is again an $r$-dimensional manifold (although not necessarily smooth). With this insight, we can now construct a reaction coordinate in Euclidean space:

**Corollary A.4.12.** *Let $\mathbf{X}_t$ be $(\varepsilon, r)$-reducible and let $\mathcal{E}\colon L^1(\mathbb{X}) \to \mathbb{R}^{2r+1}$ be one-to-one on $\mathbb{M}$ and its image. Define $\xi\colon \mathbb{R}^n \to \mathbb{R}^{2r+1}$ by*

$$\xi(x) := \mathcal{E}\big(\mathcal{Q}(x)\big)\,. \tag{A.18}$$

*Then there exists a function $\hat{\varphi}_i\colon \mathbb{R}^{2r+1} \to \mathbb{R}$ so that*

$$|\varphi_i(x) - \hat{\varphi}_i(\xi(x))| \leq \frac{\varepsilon}{|\lambda_i|}\,. \tag{A.19}$$

*Proof.* As $\mathcal{E}$ is one-to-one on $\mathbb{M}$ and its image, it is invertible on $\mathcal{E}(\mathbb{M})$. With $\tilde{\varphi}_i$ chosen as in the proof of Corollary A.4.8, define $\hat{\varphi}_i\colon \mathcal{E}(\mathbb{M}) \to \mathbb{R}$ by

$$\hat{\varphi}_i(\hat{z}) := \tilde{\varphi}_i\big(\mathcal{E}^{-1}(\hat{z})\big)\,. \tag{A.20}$$

Then

$$|\varphi_i(x) - \hat{\varphi}_i(\xi(x))| = |\varphi_i(x) - \tilde{\varphi}_i(\mathcal{Q}(x))| \overset{\text{Cor. A.4.8}}{\leq} \frac{\varepsilon}{|\lambda_i|}\,. \qquad \square$$

Since $\hat{\mathbb{M}} := \mathcal{E}(\mathbb{M})$ is an $r$-dimensional manifold, $\xi$ is effectively an $r$-dimensional reaction coordinate. Thus, if the right-hand side of (A.19) is small, the $\varphi_i$ are "almost parametrizable" by the $r$-dimensional reaction coordinate $\xi$. Using Lemma A.4.2, we immediately see that this results in a small projection error $\|P_\xi^\perp \varphi_i\|$, and due to Corollary A.3.6 in a good transfer operator approximation; hence $\xi$ is a good reaction coordinate.

The reaction coordinate $\xi$ remains an "ideal" case, because we have no access to the map $\mathcal{Q}$ and hence to $\mathbb{M}$, only to $\overline{\mathcal{Q}}(x) = p^t(x, \cdot) \approx \mathcal{Q}(x)$. We summarize the construction of the ideal reaction coordinate $\xi$ in Figure A.4.

**Remark A.4.13.** The recent work of Dellnitz et al. [41] uses similar embedding techniques to identify finite-dimensional objects in the state space of infinite-dimensional dynamical systems. They utilize the infinite-dimensional delay-embedding theorem of Robinson [156], a generalization of the well-known *Takens embedding theorem* [178], to compute finite-dimensional attractors of delay differential equations by established subdivision techniques [39].

### A.4.3 Numerical approximation of the reaction coordinate

**Approximate embedding of the transition manifold.** We now elaborate how to construct a good reaction coordinate $\overline{\xi}$ numerically. To use the central definition (A.18) in practice, two points have to be addressed:

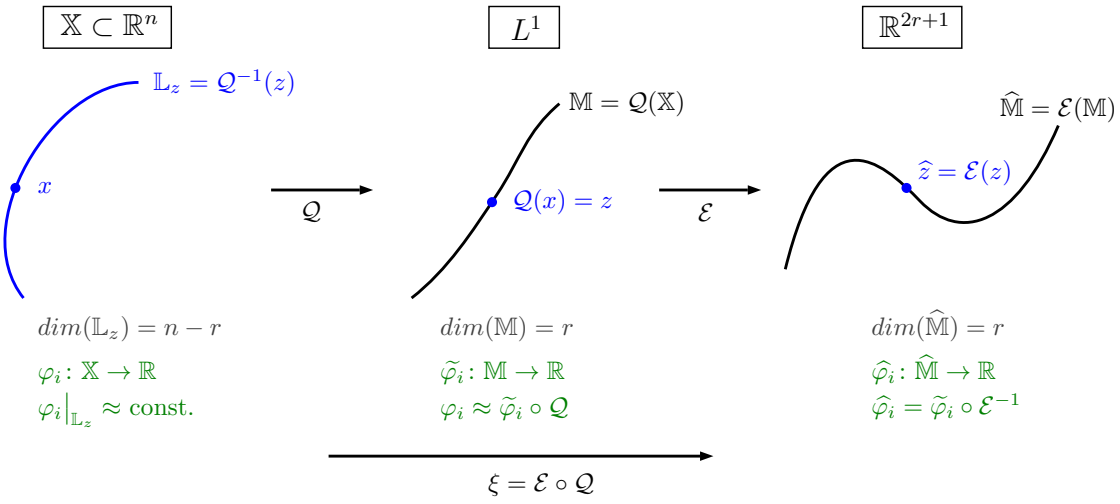



**Figure A.4:** Summary of the construction of the ideal reaction coordinate $\xi$.

1. How to choose the embedding $\mathcal{E}$?
2. How to deal with the fact that we do not know $\mathcal{Q}$?

For the choice of $\mathcal{E}$, we restrict ourselves to linear maps of the form

$$\mathcal{E}(f) := \begin{pmatrix} \langle f,\, \eta_1 \rangle \\ \vdots \\ \langle f,\, \eta_{2r+1} \rangle \end{pmatrix}, \tag{A.21}$$

with arbitrarily chosen linearly independent functions $\eta_i \in L^\infty(\mathbb{X})$, where $\langle f, \eta_i \rangle = \int f \eta_i$. In practice, we will choose the $\eta_i : \mathbb{X} \to \mathbb{R}$ as linear functions themselves, i.e. $\eta_i(x) = a_i^\intercal x$ for some, usually randomly drawn, $a_i \in \mathbb{R}^n$. Note that then $\eta_i \notin L^\infty$, but this is not a problem because we will embed the functions $f = p^t(x, \cdot)$, and $p^t(x, y)$ can be shown to decay exponentially as $\|y\|_2 \to \infty$, cf. [9, Theorem C.1]. Thus, $\langle f, \eta_i \rangle$ will exist. For linearly independent $\eta_i$, these maps are still generic in the sense of the Whitney embedding theorem, and thus still embed the transition manifold $\mathbb{M}$.

A natural choice for the approximation of the unknown map $\mathcal{Q}$ is the mapping to the transition probability density,

$$\overline{\mathcal{Q}} : x \mapsto p^t(x, \cdot)\,, \tag{A.22}$$

as $\|\mathcal{Q}(x) - p^t(x,\cdot)\|_{L^2_{1/\mu}} \leq \varepsilon$. With this, we consider

$$\mathcal{E}\big(\overline{\mathcal{Q}}(x)\big) = \mathcal{E}\big(p^t(x,\cdot)\big) = \begin{pmatrix} \langle p^t(x,\cdot),\, \eta_1 \rangle \\ \vdots \\ \langle p^t(x,\cdot),\, \eta_{2r+1} \rangle \end{pmatrix} \overset{(A.1)}{=} \begin{pmatrix} \mathcal{K}^t \eta_1(x) \\ \vdots \\ \mathcal{K}^t \eta_{2r+1}(x) \end{pmatrix}. \tag{A.23}$$

The values on the right-hand side can in turn be approximated by a Monte Carlo quadrature, using only short-time trajectories of the original dynamics:

$$\mathcal{K}^t \eta_i(x) = \mathsf{E}\big[\eta_i(\mathbf{X}_t) \mid \mathbf{X}_0 = x\big] \approx \frac{1}{M} \sum_{m=1}^{M} \eta_i\big(\mathbf{\Phi}_t^{(m)}(x)\big), \tag{A.24}$$

where the $\mathbf{\Phi}_t^{(m)}(x)$ are independent realizations of $\mathbf{X}_t$ with starting point $\mathbf{X}_0 = x$, in practice realized by a stochastic integrator (e.g. Euler–Maruyama).

**The computationally infeasible reaction coordinate $\xi$.** Note that $\mathcal{E} \circ \overline{\mathcal{Q}}$ is not yet an $r$-dimensional reaction coordinate, since $\overline{\mathcal{Q}}(\mathbb{X})$ is only approximately an $r$-dimensional manifold; more precisely, it lies in the $\varepsilon$-neighborhood of an $r$-dimensional submanifold $\mathbb{M}$ of $L^1$. Hence, also $\mathcal{E}(\overline{\mathcal{Q}}(\mathbb{X}))$ is only approximately an $r$-dimensional manifold; see the magenta regions in Figure A.5.

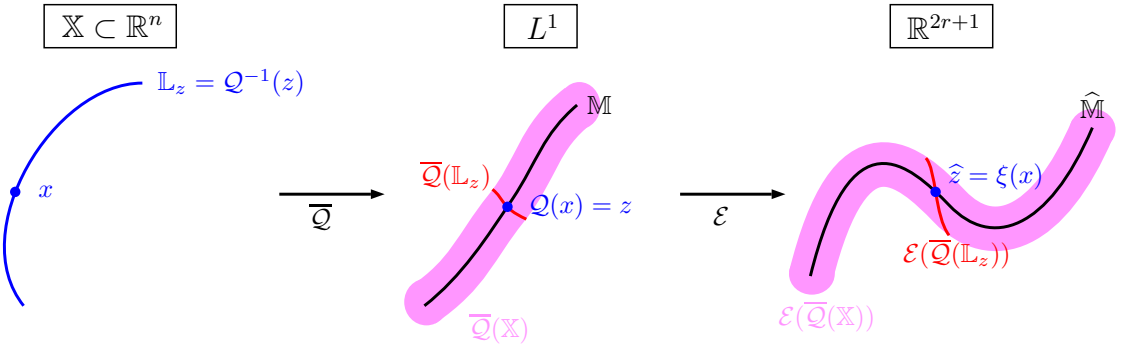



**Figure A.5:** How to make a good $r$-dimensional reaction coordinate out of $\mathcal{E} \circ \overline{\mathcal{Q}}$? Given the analysis from the previous section, we would like to make the level sets $\mathbb{L}_z$ of $\mathcal{Q}$ also the level sets of $\overline{\xi}$ (red line segment). Unfortunately, we have no access to these.

The question now is how we can reduce $\mathcal{E} \circ \overline{\mathcal{Q}}$ to an $r$-dimensional *good* reaction coordinate. Since we know from above that $\xi = \mathcal{E} \circ \mathcal{Q}$ is a good reaction coordinate, let us see what would be needed to construct it.

The property of $\xi$ that we want is that it is constant along level sets $\mathbb{L}_z$ of $\mathcal{Q}$, i.e., $\xi|_{\mathbb{L}_z} = \text{const}$ (because this implies that it is a good reaction coordinate, cf. Corollary A.4.12). Hence, if we could identify $\hat{\mathbb{M}}$ as an $r$-dimensional manifold in $\mathbb{R}^{2r+1}$, we would project $\mathcal{E}(\overline{\mathcal{Q}}(x))$ along $\mathcal{E}(\overline{\mathcal{Q}}(\mathbb{L}_z))$ onto $\hat{\mathbb{M}}$ — assuming that $\hat{\mathbb{M}}$ and $\mathcal{E}(\overline{\mathcal{Q}}(\mathbb{L}_z))$ intersect in $\mathbb{R}^{2r+1}$ — to obtain $\xi(x)$ as the resulting point (see Figure A.5, where we would project along the red line on the right). Unfortunately, we have no access to $\mathcal{Q}$ (not to mention that $\hat{\mathbb{M}}$ and $\mathcal{E}(\overline{\mathcal{Q}}(\mathbb{L}_z))$ need not intersect in $\mathbb{R}^{2r+1}$) and hence to its level sets $\mathbb{L}_z$. Thus, this strategy seems infeasible.

**A computationally feasible reaction coordinate.** What helps us at this point is that there is a certain amount of arbitrariness in the definition of $\mathcal{Q}$. Recalling Definition A.4.4,

what we are given is $\overline{\mathcal{Q}}$, and we construct $\mathcal{Q}(x)$ as a projection of $\overline{\mathcal{Q}}(x)$ onto the $r$-dimensional manifold $\mathbb{M}$ by the closest-point projection $\mathcal{Q}'$; i.e., $\mathcal{Q} = \mathcal{Q}' \circ \overline{\mathcal{Q}}$. This choice of $\mathcal{Q}'$ is convenient, because we can show

$$\|\overline{\mathcal{Q}}(x) - \overline{\mathcal{Q}}(y)\|_{L^2_{1/\mu}} \leq 2\varepsilon \quad \text{for every } \mathcal{Q}(x) = \mathcal{Q}(y) \text{ (i.e., on level sets of } \mathcal{Q}'), \tag{A.25}$$

which is used in Lemma A.4.7. Other choices of $\mathcal{Q}'$ could, however, yield a similarly practicable $\mathcal{O}(\varepsilon)$-bound in (A.25). Our strategy will be to choose a specific $r$-dimensional reaction coordinate $\overline{\xi}$ and to show that in general it can be expected to be a good reaction coordinate.

Let us recall that, by assumption, the set $\overline{\mathcal{Q}}(\mathbb{X})$ is contained in the $\varepsilon$-neighborhood of an unknown smooth $r$-dimensional manifold $\mathbb{M} \subset L^1(\mathbb{X})$. Thus, a generic smooth map $\mathcal{E}\colon L^1(\mathbb{X}) \to \mathbb{R}^{2r+1}$ will embed $\mathbb{M}$ into $\mathbb{R}^{2r+1}$, forming a diffeomorphism from $\mathbb{M}$ to $\hat{\mathbb{M}}$. Thus, $\mathcal{E}$ is going to map $\overline{\mathcal{Q}}(\mathbb{X})$ to an $\mathcal{O}(\varepsilon)$-neighborhood of $\hat{\mathbb{M}}$. This means, the $r$-dimensional manifold structure of $\hat{\mathbb{M}}$ should still be detectable and can be identified with standard manifold learning tools. We use the diffusion maps algorithm (see Section A.4.4 below), which gives us a map $\Psi : \mathbb{R}^{2r+1} \to \mathbb{R}^r$ (the diffusion map). Then we define $\overline{\xi}$ as

$$\overline{\xi} := \Psi \circ \mathcal{E} \circ \overline{\mathcal{Q}}. \tag{A.26}$$

This is depicted on the right-hand side of Figure A.6, where the red dashed line shows the level set $\hat{\mathbb{L}}_{\hat{z}} = \{z \in \mathbb{R}^{2r+1} : \Psi(z) = \Psi(\hat{z})\}$.

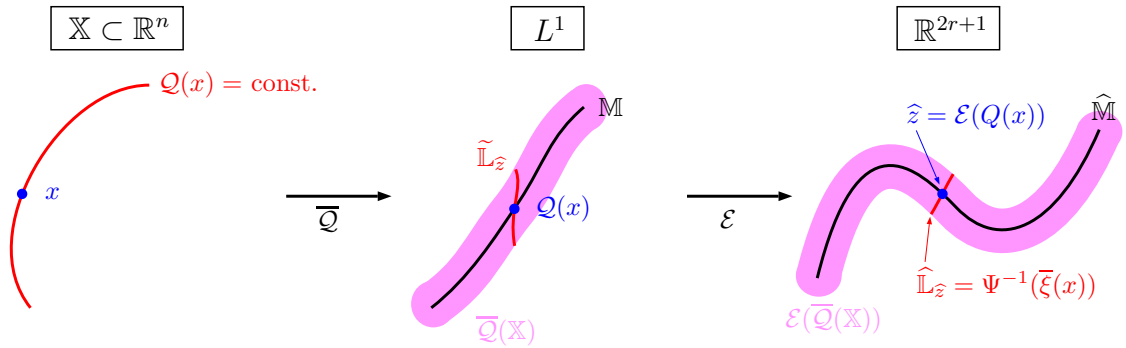



**Figure A.6:** The realized reaction coordinate $\overline{\xi}$.

Next, we consider $\tilde{\mathbb{L}}_{\hat{z}} := \mathcal{E}^{-1}(\hat{\mathbb{L}}_{\hat{z}}) \cap \overline{\mathcal{Q}}(\mathbb{X})$. It holds that $\tilde{\mathbb{L}}_{\hat{z}} = \big\{\overline{\mathcal{Q}}(x) \,\big|\, \overline{\xi}(x) = \Psi(\hat{z})\big\}$. Recall that $\mathcal{E}\colon \mathbb{M} \to \hat{\mathbb{M}}$ is one-to-one, thus $\tilde{\mathbb{L}}_{\hat{z}}$ intersects $\mathbb{M}$ in exactly one point. We define this one point as $\mathcal{Q}(x)$, and thus $\mathcal{Q}'$ is the projection onto $\mathbb{M}$ along $\tilde{\mathbb{L}}_{\hat{z}}$. We see that $\mathcal{Q}$ is well-defined and that $\mathcal{Q}(x) = \mathcal{Q}(y) \Leftrightarrow \overline{\xi}(x) = \overline{\xi}(y)$.

At this point we assume that $\mathcal{E}^{-1}$ is sufficiently well-behaved in a neighborhood of $\hat{\mathbb{M}}$, it does not "distort transversality" of intersections, such that the diameter of $\tilde{\mathbb{L}}_{\hat{z}}$ is $\mathcal{O}(\varepsilon)$ with a moderate constant in $\mathcal{O}(\cdot)$. We will investigate a formal justification of this fact in a future work, here we assume it holds true, and we will see in the numerical experiments

that the assumption is justified. This assumption implies that $\|\overline{\mathcal{Q}}(x) - \overline{\mathcal{Q}}(y)\|_{L^2_{1/\mu}} = \mathcal{O}(\varepsilon)$ for $\mathcal{Q}(x) = \mathcal{Q}(y)$, i.e. for $\overline{\xi}(x) = \overline{\xi}(y)$. Now, however, Lemma A.4.7 implies that $\varphi_i$ is almost constant (up to an error $\mathcal{O}(\varepsilon)$) on level sets of $\overline{\xi}$, which, in turn, by Lemma A.4.2 and Corollary A.3.6 shows that $\overline{\xi}$ is a good reaction coordinate.

### A.4.4 Identification of $\hat{\mathbb{M}}$ through Manifold Learning

In this section, we describe how to identify $\hat{\mathbb{M}}$ numerically. The task is as follows: Given that we have computed $\mathcal{E}(\overline{\mathcal{Q}}(x_i)) = \hat{z}_i \in \mathbb{R}^{2r+1}$ for a number of sample points $\{x_i\}_{i=1}^{\ell} \subset \mathbb{X}$, we would like to identify the $r$-dimensional manifold $\hat{\mathbb{M}}$, noting the points $\mathcal{E}(\overline{\mathcal{Q}}(x_i))$ are in a $\mathcal{O}(\varepsilon)$-neighborhood of $\hat{\mathbb{M}}$ (see Section A.4.3). Additionally, we would like an $r$-dimensional coordinate function $\Psi : \mathbb{R}^{2r+1} \to \mathbb{R}^r$ that parameterizes $\hat{\mathbb{M}}$ (so that the level sets of $\Psi$ are transversal to $\hat{\mathbb{M}}$).

This is a default setting for which manifold learning algorithms can be applied. Many standard methods exist; we name multidimensional scaling [102, 101], Isomap [181], and diffusion maps [33] as a few of the most prominent examples. Because of its favorable properties, we choose the diffusion maps algorithm here and summarize it briefly for our setting in what follows. For details, the reader is referred to [33, 132, 32, 176].

Given sample points $\{\hat{z}_i\}_{i=1}^{\ell} \subset \mathbb{R}^{2r+1}$, diffusion maps proceeds by constructing a similarity matrix $W \in \mathbb{R}^{\ell \times \ell}$ with

$$W_{ij} = h\left(\frac{\|\hat{z}_i - \hat{z}_j\|_2^2}{\sigma}\right),$$

where $\|\cdot\|_2$ is the Euclidean norm in $\mathbb{R}^{2r+1}$, $\sigma > 0$ is a scale factor, and $h : \mathbb{R} \to \mathbb{R}_+$ is a kernel function which is most commonly chosen as $h(x) = \exp(-x)1_{x \leq R}$ with a suitably chosen cutoff $R$ that sparsifies $W$ and ensures that only local distances enter the construction. With $D$ being the diagonal matrix containing the row sums of $W$, the similarity matrix is then normalized to give $\tilde{W} = D^{-1}WD^{-1}$. Finally, the stochastic matrix $P = \tilde{D}^{-1}\tilde{W}$ is constructed, where $\tilde{D}$ is the diagonal matrix containing the row sums of $\tilde{W}$. $P$ is similar to the symmetric matrix $\tilde{D}^{-1/2}\tilde{W}\tilde{D}^{-1/2}$, thus it has an orthonormal basis of eigenvectors $\{\psi_i\}_{i=0}^{\ell-1}$ with real eigenvalues $\gamma_i$. Since $P$ is also stochastic, $|\gamma_i| \leq 1$. The diffusion map is then given by

$$\Psi : \mathbb{R}^{2r+1} \to \mathbb{R}^r, \quad \Psi(\hat{z}) = (\gamma_1\psi_1(\hat{z}), \ldots, \gamma_r\psi_r(\hat{z}))^\intercal . \tag{A.27}$$

Using properties of the Laplacian eigenproblem on $\hat{\mathbb{M}}$, one can show that $\Psi$ indeed parameterizes the $r$-dimensional manifold $\hat{\mathbb{M}}$ for suitably chosen $\sigma$ [33].

**Remark A.4.14.** The diffusion maps algorithm will only reliably identify $\hat{\mathbb{M}}$ based on the neighborhood relations between the embedded sample points $z_i$, if the points cover all parts of $\hat{\mathbb{M}}$ sufficiently well. In particular, as $p^t(x, \cdot)$ and thus $\big(\mathcal{E} \circ \overline{\mathcal{Q}}\big)(x)$ vary strongly with $x$ traversing the transition regions, a good coverage of those regions is required.

For the various low-dimensional academical examples Section A.5, this is ensured by choosing the $x_i$ to be a dense grid of points in $\mathbb{X}$. For the high-dimensional example in

Section A.5.2, the evaluation points are generated as a subsample from a long equilibrated trajectory, essentially sampling $\mu$. Both of these ad-hoc methods are likely to be unapplicable in realistic high-dimensional systems with very long equilibration times. However, as we mentioned in the introduction, there exist multiple statistical and dynamical approaches to this common problem of quickly sampling the relevant parts of phase space, including the transition regions. Each of these sampling methods can be easily integrated into our proposed algorithm as a pre-processing step.

Fundamentally though, the central idea of our method does not depend crucially on the applicability of diffusion maps. Rather, the latter can be considered an optional post-processing step. Using the $2r+1$-dimensional reaction coordinate

$$\overline{\overline{\xi}} := \mathcal{E} \circ \overline{\mathcal{Q}} \,,$$

i.e. (A.26) without the manifold learning step, may in practice already represent a sufficient dimensionality reduction.

In addition, situations may occur where the a priori generation of evaluation points is not possible or desired. One of the final goals and currently work in progress is the construction of an accelerated integration scheme that generates significant evaluation points and their reaction coordinate value "on the fly". This is related to the effective dynamics mentioned in fifth point of the conclusion. However, this also requires us to be able to evaluate the reaction coordinate at isolated points, independent of each other, and thus also necessitates the use of the above $\overline{\overline{\xi}}$ instead of $\overline{\xi}$.

## A.5 Numerical Examples

Based on the results from the previous sections, we propose the following algorithm to compute reaction coordinates numerically:

1. Let $x_i$, $i = 1, \ldots, \ell$, be the points for which we would like to evaluate $\overline{\xi}$. Here, we assume the points satisfy the requirements addressed in Remark A.4.14.

2. Choose linearly independent functions $\eta_j \in L^\infty(\mathbb{X})$, $j = 1, \ldots, 2r+1$. The essential boundedness of the $\eta_j$ is not necessary, but $|\eta_j(x)|$ should not grow faster than a polynomial as $\|x\|_2 \to \infty$.

3. In each point $x_i$, start $M$ simulations of length $t$ and estimate $\mathcal{E}_j\big(\overline{\mathcal{Q}}(x_i)\big)$ using (A.23) and (A.24), to obtain the point $\hat{z}_i \in \mathbb{R}^{2r+1}$. We discuss the appropriate choice of $M$ and $t$ in Section A.5.1.

4. Apply the diffusion maps technique from Section A.4.4 for the point cloud $\{\hat{z}_i\}_{i=1}^{\ell}$, and obtain $\Psi : \mathbb{R}^{2r+1} \to \mathbb{R}^r$, a parametrization of the point in its $r$ essential directions of variation.

5. By (A.27), we define the reaction coordinate as

$$\overline{\xi} : \, x_i \mapsto \Psi(\hat{z}_i) \,. \tag{A.28}$$

The numerical effort of this algorithm depends strongly on the third step. Given $\ell$ evaluation points, and a choice of $M$ trajectories per point, the cost is mainly given by $M \cdot \ell \cdot c(t)$, where $c(t)$ is the effort of a single numerical realization of the dynamics up to time $t$. The high-dimensional phase space only enters the algorithm as the domain of the observables $\eta_j$. The cost of evaluating those typically very simple functions[4] at the $M \cdot \ell$ end points of the trajectory is negligible. The cost of the method is thus essentially independent of $n$.

In order to demonstrate the efficacy of our method, we compute the reaction coordinates for three representative problems, namely a simple curved double-well potential, a multi-well potential defined on a circle, both in low and high dimensions, and two slightly different quadruple-well potentials stressing the difference between a one- and a two-dimensional reaction coordinate.

### A.5.1 Curved double-well potential

As a first verification, we consider a system with an analytically known reaction coordinate that is then used for comparison. Consider the two-dimensional drift-diffusion process (A.2) with potential

$$V(x_1, x_2) = (x_1^2 - 1)^2 + 2(x_1^2 + x_2 - 1)^2$$

and inverse temperature $\beta = 0.5$. This potential already served as a motivational example for the nature of reaction coordinates in the introduction and is shown in Figure A.1. The system possesses two metastable sets around the minima $(-1, 0)^\intercal$ and $(1, 0)^\intercal$, which are connected by the transition path $\{x \in \mathbb{R}^2 \mid x_2 = 1 - x_1^2\}$. The implied time scales, defined in (A.3), can be computed from the eigenvalues using a standard Ulam-type Galerkin discretization [92, 93] of the transfer operator $\mathcal{T}^t$ and are shown in Figure A.7a[5]. We observe a significant gap between $t_1$ and $t_2$ and thus identify $t_1$ as the last slow and $t_2$ as the first fast time scale. Choosing the lag time $t = 2$ then satisfies $t_{\text{slow}} > t > t_{\text{fast}}$. A visual inspection of a typical trajectory of length $t$ starting in one of the two metastable sets as shown in Figure A.7b confirms that the respective set is sampled, yet a transition to the other set is a rare event.

The low dimension of the system allows us to compute the reaction coordinate on a full regular grid over the phase space. We choose a $40 \times 30$ grid in the rectangular region $[-2, 2] \times [-1, 2]$ and denote the set of grid points by $\overline{\mathbb{X}}$. For this system, we expect a one-dimensional transition path and thus a one-dimensional reaction coordinate $\xi$. That is, $r = 1$ and $2r + 1 = 3$. Thus, we choose three linear observables in our embedding

[4] In our examples, we used linear functions with great success.

[5] In realistic, high-dimensional systems, the computation of the dominant eigenvalues using grid-based methods is likely infeasible. In these situations, the implied time scales have to be estimated, for example using standard Markov State Model techniques [16].

a)

| | |
|---|---|
| $t_0$ | $\infty$ |
| $t_1$ | 6.1823 |
| $t_2$ | 0.9066 |
| $t_3$ | 0.6098 |
| $t_4$ | 0.3976 |

b)

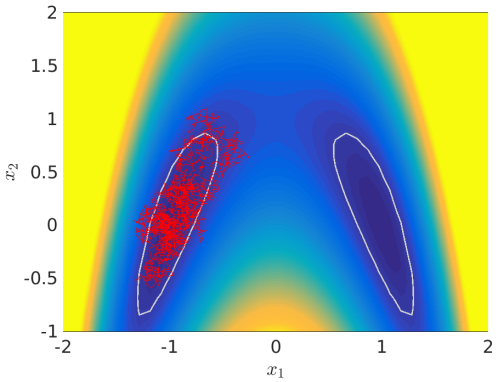



**Figure A.7:** a) Implied time scales of the double-well system. b) Trajectory of length $t = 1$.

function (A.21), e.g.,

$$\begin{aligned} \eta_1(x_1, x_2) &= -0.2630\, x_1 - 0.3186\, x_2, \\ \eta_2(x_1, x_2) &= -0.2246\, x_1 + 0.0969\, x_2, \\ \eta_3(x_1, x_2) &= \quad 0.1564\, x_1 + 0.0783\, x_2, \end{aligned} \tag{A.29}$$

whose coefficients were drawn uniformly from $[-1, 1]$. The expectation value in (A.23) is approximated by a Monte Carlo quadrature using $M = 10^5$ sample trajectories for each grid point, cf. (A.24). The parameter $M$ was chosen such that the error in (A.24), commonly defined as the variance of the Monte Carlo sum, is sufficiently low. The resulting embedding of the grid points $x$ into $\mathbb{R}^3$ is shown in Figure A.8. The transition path seems to be already parametrized well by the individual components of $\mathcal{E} \circ \overline{Q}$.

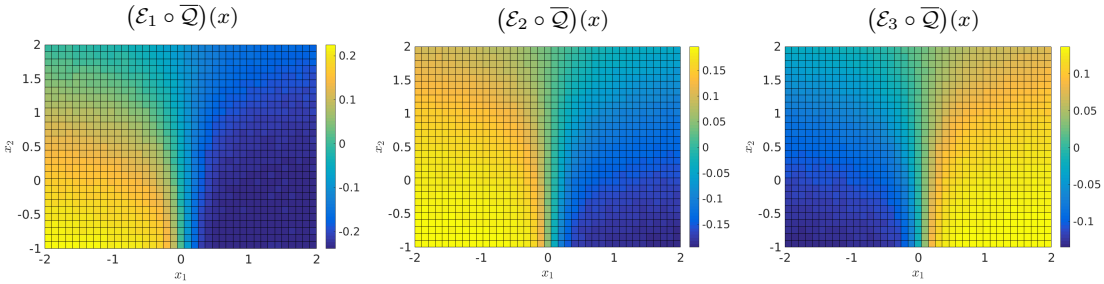



**Figure A.8:** The individual components of the embedding $\mathcal{E} \circ \overline{\mathcal{Q}}$ on the grid points $x \in \overline{\mathbb{X}}$.

For this example, the image of $\mathbb{X}$ under $\mathcal{E} \circ \overline{\mathcal{Q}}$ should form a compact neighborhood of the one-dimensional manifold $\mathcal{E}(\mathbb{M})$, as described in Section A.4.3. The one-dimensional structure in $\mathcal{E}\big(\overline{\mathcal{Q}}(\overline{\mathbb{X}})\big)$ is clearly visible, see Figure A.9a. To identify the one-dimensional coordinate along this set the diffusion map algorithm is used. Let $\Psi_1 : \big(\mathcal{E} \circ \overline{\mathcal{Q}}\big)(\overline{\mathbb{X}}) \to \mathbb{R}$ denote the first diffusion map coordinate on the embedded grid points, also visualized in

Figure A.9a. The final reaction coordinate, shown in Figure A.9b, is then given by

$$\overline{\xi}(x) := \Psi_1\big((\mathcal{E} \circ \overline{\mathcal{Q}})(x)\big), \quad x \in \overline{\mathbb{X}}.$$

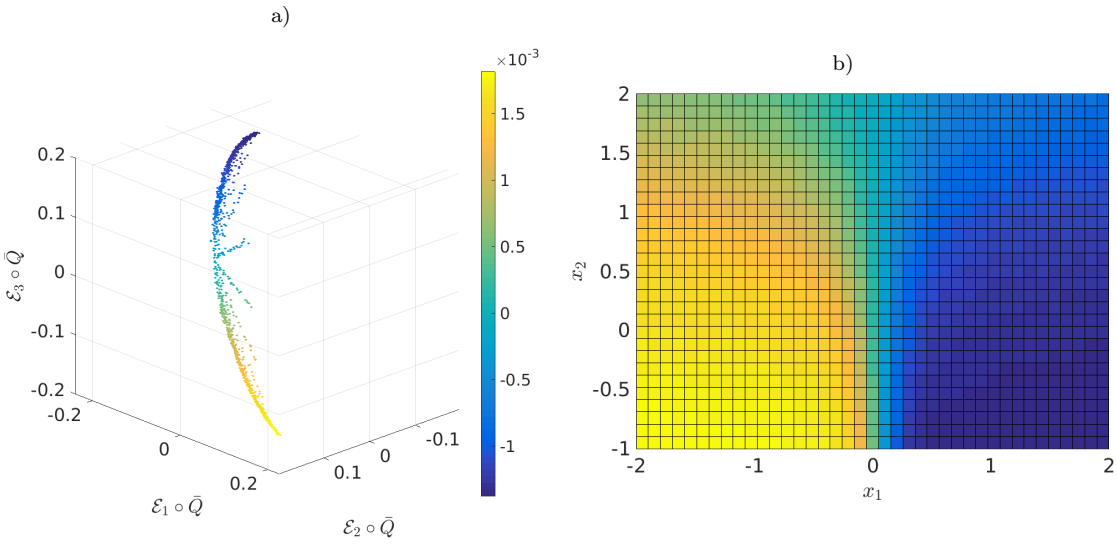



**Figure A.9:** a) The embedded grid points colored according to the first diffusion map coordinate. b) The final reaction coordinate $\overline{\xi}$.

Legoll and Lelièvre [109] show that the effective dynamics based on the reaction coordinate

$$\xi^*(x) = x_1 \exp(-2x_2)$$

accurately reproduces the long-time dynamics of the full process — although they do not use dominant eigenvalues of the transfer operator in their argumentation. It is easy to verify that the level sets of $\xi^*$ traverse the transition path orthogonally. Figure A.10 compares the level sets of $\overline{\xi}$ and $\xi^*$. While the two reaction coordinates have different absolute values, their contour lines coincide well. As the projection operator $P_\xi$ only depends on the level sets of $\xi$, the projected transfer operators $\mathcal{T}_{\overline{\xi}}^t$ and $\mathcal{T}_{\xi^*}^t$ should be similar as well.

**Projected eigenvalue error.** To conclude this example, we compute the dominant spectrum of the projected transfer operator and compare it to the spectrum of the full transfer operator. To discretize $\mathcal{T}_{\overline{\xi}}^t$, we use a simple Ulam-type discretization scheme based on a long equilibrated trajectory of the full dynamics. Recall from Section A.3.2 that, although $\mathcal{T}_\xi^t$ formally acts as an operator on functions over $\mathbb{X}$, it is constant along level sets of $\overline{\xi}$, and thus can be treated as an operator on functions over $\mathbb{R}^r$. For completeness, we state the rough outline of an algorithm that we used to approximate $\mathcal{T}_{\overline{\xi}}^t$. An introduction

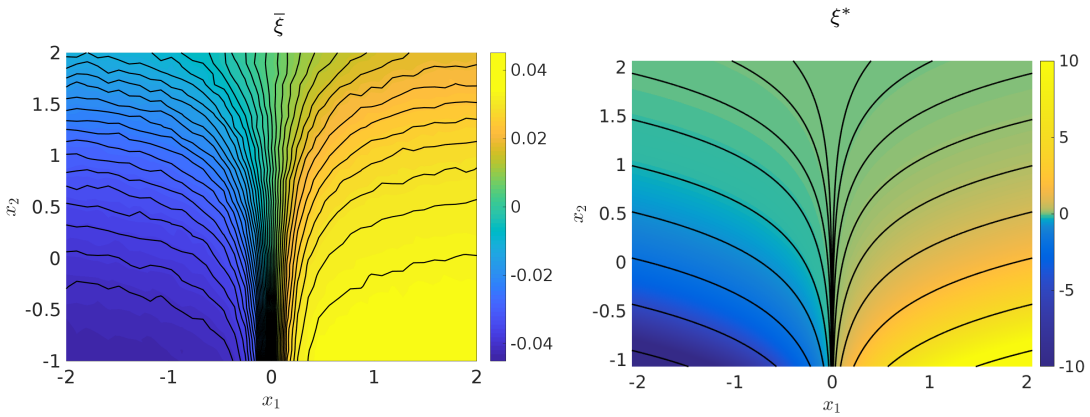



**Figure A.10:** Selected contour lines (black) of the newly identified reaction coordinate $\bar{\xi}$ and the reference reaction coordinate $\xi^*$.

to Ulam- and other Galerkin-type discretization schemes for transfer operators can be found, e.g., in [92].

1. Compute points $\overline{\mathbb{X}} := \{\mathbf{\Phi}_{(k\tau)}x_0 \mid k = 1, \dots, N\}$, a discrete trajectory with step size $\tau$ of the full phase space dynamics that adequately samples the invariant density $\varrho$.

2. Compute the reaction coordinate $\bar{\xi}$ on the points $\overline{\mathbb{X}}$.

3. Divide the neighborhood of $\bar{\xi}(\overline{\mathbb{X}})$ into boxes or other suitable discretization elements $\{\mathbb{A}_1, \dots, \mathbb{A}_N\}$ and sample the boxes from the trajectory, i.e. compute

$$\overline{\mathbb{X}}_i := \{x \in \overline{\mathbb{X}} \mid \bar{\xi}(x) \in \mathbb{A}_i\} \ .$$

4. Count the time-$t$-transitions within $\overline{\mathbb{X}}$ between the boxes (where $t$ is a multiple of $\tau$), i.e. compute the matrix

$$\left(T^t_{\bar{\xi}}\right)_{ij} := \#\left\{x \in \overline{\mathbb{X}}_i \mid \mathbf{\Phi}_t x \in \overline{\mathbb{X}}_j\right\} \ .$$

5. After row-normalization, the eigenvalues of $T^t_{\bar{\xi}}$ approximate the point spectrum of $\mathcal{T}^t_{\bar{\xi}}$.

**Remark A.5.1.** Note that the equilibrated trajectory $\overline{\mathbb{X}}$ is typically unavailable for more complex systems. In practice, one would replace steps 1 and 2 by directly computing a reduced trajectory $\overline{\mathbb{Z}} = \{z_1, \dots, z_N\} \subset \mathbb{R}^r$ whose statistics approximate that of $\xi\left(\overline{\mathbb{X}}\right)$. The formulation of a reduced numerical integration scheme to realize this is currently work in progress (see the fifth point in the conclusions).

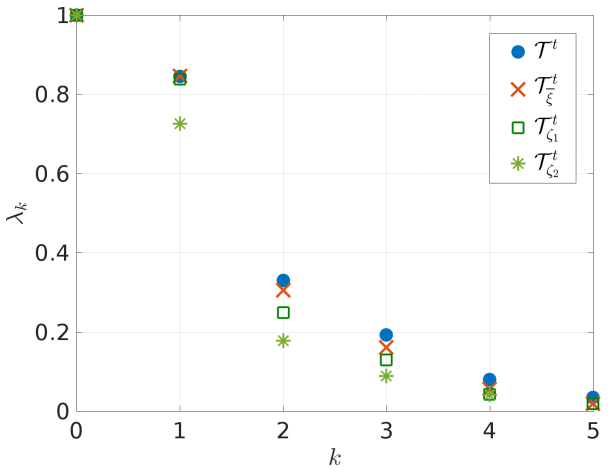
For our example system, we compute $\overline{\mathbb{X}}$ as a $N = 10^6$ step trajectory with step size $\tau = 10^{-2}$ using the Euler-Maruyama scheme. However, to reduce the numerical effort, $\overline{\xi}$ is computed only on a subsample of $\overline{\mathbb{X}}$ ($10^4$ points) and extended to $\overline{\mathbb{X}}$ by nearest-neighbor interpolation. On $\overline{\mathbb{X}}$, the image of the $\overline{\xi}$ is contained in the interval $[-0.04, 0.04]$, which we discretize into $M = 40$ subintervals of equal length. The spectrum of the full transfer operator $\mathcal{T}^t$ was computed using the standard Ulam method over a $40 \times 30$ uniform box discretization of the domain $[-2, 2] \times [-1, 2]$. With the choice $t = 1$ for the lag time, the spectral gap is clearly visible.

We observe in Figure A.11 that the eigenvalues of $\mathcal{T}^t_{\overline{\xi}}$ and $\mathcal{T}^t$ are in excellent agreement. Not only the dominant eigenvalues $\lambda_0, \lambda_1$ are approximated well (as predicted by Lemma A.3.5), but also the further subdominant eigenvalues that are not covered by our theory. In particular, the reaction coordinate $\overline{\xi}$ provides a better approximation to the spectrum of $\mathcal{T}^t$ than other, manually chosen reaction coordinates: Figure A.11 also shows the eigenvalues of the projected transfer operator associated with the reaction coordinates

$$\zeta_1(x) := x_1 \quad \text{and} \quad \zeta_2(x) := x_1 + x_2.$$

We see that these are consistently outperformed by the computed reaction coordinate $\overline{\xi}$ (although it appears that $\zeta_1$ already is quite a good reaction coordinate).

a)




b)

| | $\lambda_1$ |
|---|---|
| $\mathcal{T}^t$ | 0.8503 |
| $\mathcal{T}^t_{\overline{\xi}}$ | 0.8461 |
| $\mathcal{T}^t_{\zeta_1}$ | 0.8377 |
| $\mathcal{T}^t_{\zeta_2}$ | 0.7252 |

**Figure A.11:** a) Comparison of the two dominant and first four non-dominant eigenvalues of the full transfer operator $\mathcal{T}^t$ and the projected transfer operators $\mathcal{T}^t_{\overline{\xi}}, \mathcal{T}^t_{\zeta_1}, \mathcal{T}^t_{\zeta_2}$. b) Detailed comparison of the second eigenvalue of the various transfer operators.

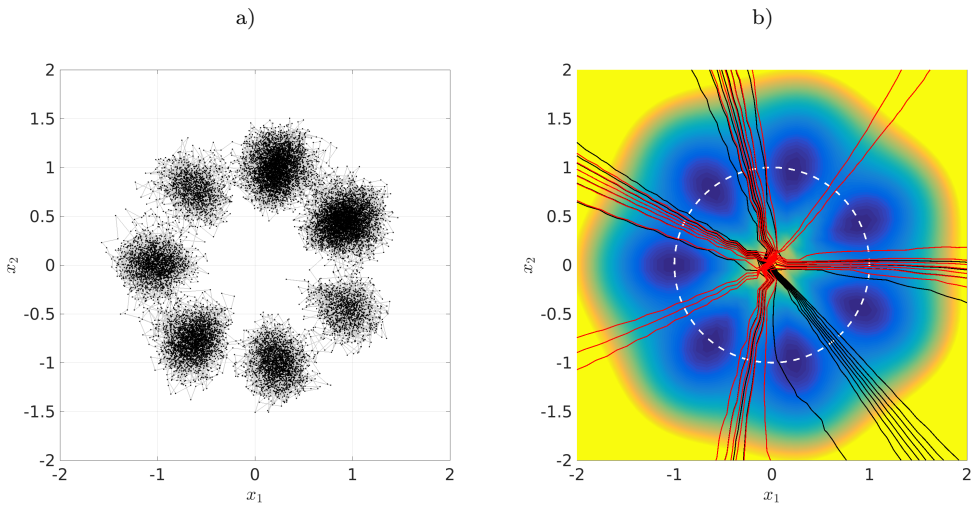



**Figure A.12:** a) Longtime trajectory of the diffusion process with the circular seven-well potential. b) The contour lines of $\overline{\xi}_1$ (black) and $\overline{\xi}_2$ (red) show that $\overline{\xi}$ is almost constant on the metastable sets, but resolves the transition regions well.

### A.5.2 Circular potential

Let us now compute the reaction coordinates for the multi-well diffusion process described in Example A.4.1. The corresponding $k$-well potential is defined as

$$V(x) = \cos\left(k \arctan(x_2, x_1)\right) + 10\left(\sqrt{x_1^2 + x_2^2} - 1\right)^2.$$

We use $k = 7$, for which the potential is shown in Figure A.2a. The potential as well as the dominant eigenvalues of the corresponding transfer operator clearly indicate the existence of seven metastable sets, yet a typical longtime trajectory, shown in Figure A.12a, suggests a one-dimensional transition path, the unit circle $\mathbb{B}_1$. We demonstrate that with our method, a reaction coordinate of minimal dimension can be computed.

We again choose the inverse temperature $\beta = 0.5$ and perform the same analysis as in the previous subsection. For this system, a time scale gap between $t_6 \approx 1.53$ and $t_7 \approx 0.05$ can be found. We thus choose the intermediate time scale $t = 0.1$. Since we again expect a one-dimensional transition path, the three observables (A.29) are used for the embedding of $\mathbb{M}$. We use the grid points of a $40 \times 40$ grid, denoted again by $\overline{\mathbb{X}}$, over the region $[-2, 2] \times [-2, 2]$ as our test points.

The individual components of the embedding $\mathcal{E} \circ \overline{\mathcal{Q}}$ are shown in Figure A.13. The embedded grid points, seen as the individual points in Figure A.14a, seem to concentrate around a one-dimensional circular manifold and thus reveal the one-dimensional nature of the reaction coordinate. Although slightly unintuitive, the diffusion maps algorithm now identifies *two* significant diffusion map components, as shown in Figure A.14a. The reason is that the circular manifold cannot be embedded into $\mathbb{R}^1$, so that a two-component

coordinate is necessary to parametrize it. Figure A.12b shows some contour lines (of equidistant values) of the two components of $\overline{\xi}$. We see that $\overline{\xi}$ is almost constant on the seven metastable sets, but resolves the transition regions well.

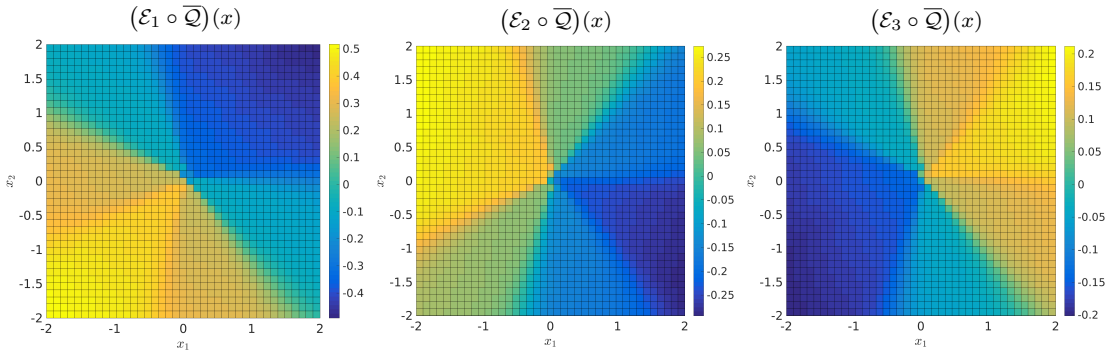



**Figure A.13:** The individual components of the embedding $\mathcal{E} \circ \overline{\mathcal{Q}}$ on the grid points $x \in \overline{\mathbb{X}}$.

**Parametrization of the dominant eigenfunctions.** Next, we experimentally investigate how well the dominant eigenfunctions $\varphi_i$ of $\mathcal{T}^t$ can be parametrized by the numerically computed reaction coordinate $\overline{\xi}$. If the eigenfunctions are almost functions of $\overline{\xi}$, then by Lemma A.4.2 and Corollary A.3.6 the reaction coordinate is suitable to reproduce *all the dominant time scales*. To this end, we compute the dominant eigenfunctions $\varphi_j$, $j = 0, \ldots, d$ by the Ulam-type Galerkin method (as in the previous example), and plot $\varphi_j(x_i)$ against $\overline{\xi}(x_i)$. Note that due to the reasons discussed above, the range of $\overline{\xi}$ is a one-dimensional manifold in $\mathbb{R}^2$. If $\varphi_j$ can be parametrized by $\overline{\xi}$, we expect that $\varphi_j(x_{i_1}) \approx \varphi_j(x_{i_2})$, whenever $\overline{\xi}(x_{i_1}) \approx \overline{\xi}(x_{i_2})$. The result is shown in Figure A.15. We clearly see the functional dependency of the first seven (i.e., the dominant) eigenfunctions on the reaction coordinate.

**Circular potential in higher dimensions.** The identification of reaction coordinates is not limited to two dimensions. To show that our method can effectively find the reaction coordinates in high-dimensional systems, we extend the 7-well potential to ten dimensions by adding a quadratic term in $x_3, \ldots, x_{10}$:

$$V(x) = \cos\left(7 \arctan(x_2, x_1)\right) + 10\left(\sqrt{x_1^2 + x_2^2} - 1\right)^2 + 10 \sum_{j=3}^{10} x_j^2 \,.$$

We expect the one-dimensional circle $\{x \in \mathbb{R}^{10} \mid x_1^2 + x_2^2 = 1,\ x_j = 0,\ j = 3, \ldots, 10\}$ to be the transition path and accordingly choose a three-dimensional linear observable $\eta(x) = A \cdot x$, $A \in \mathbb{R}^{3\times 10}$, where the coefficients $A_{ij}$ were again drawn uniformly from $[-1, 1]$.

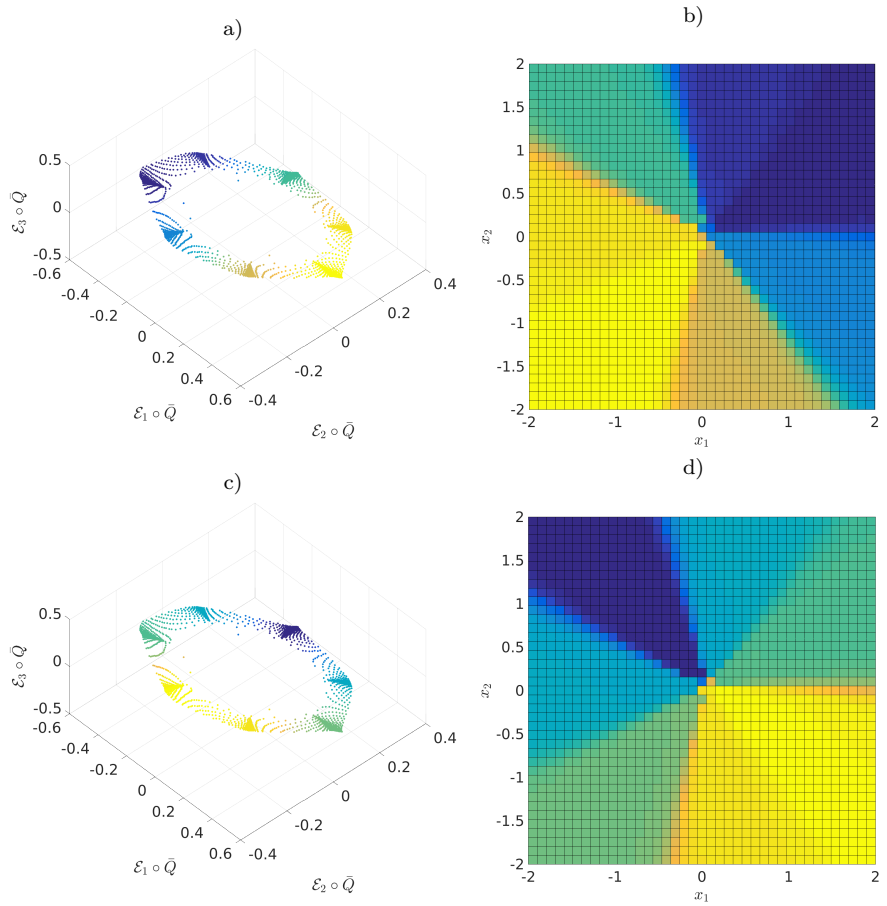



**Figure A.14:** Left column: The embedded grid points $\mathcal{E}(\overline{\mathcal{Q}}(\overline{\mathbb{X}}))$. The coloring shows the a) first and c) second significant diffusion map on the points. Right column: The b) first and d) second components of the final reaction coordinate $\overline{\xi}$.

In ten dimensions, the computation of the reaction coordinate on all points of a regular grid is no longer possible due to the curse of dimensionality, and neither is the visualization of this grid. Instead, we compute $\overline{\xi}$ on $10^5$ points sampled from the invariant measure and plot only the first three coordinates. Let this point cloud be called $\overline{\mathbb{X}}$.

Performing the standard procedure, i.e. embedding $\overline{\mathbb{X}}$ into $\mathbb{R}^3$ and identifying the one-dimensional core using diffusion maps, a two-component reaction coordinate is identified. Coloring the first three dimensions of $\overline{\mathbb{X}}$ by $\overline{\xi}$ (Figure A.16a,b), we see that the expected reaction pathway is indeed parametrized. This pathway as well as the seven metastable states can also be recognized in a plot of the components of $\overline{\xi}(\overline{\mathbb{X}})$ plotted against each other, indicating that the information about the dominant eigenfunctions, thus the long-time jump process, is indeed retained by $\overline{\xi}$.

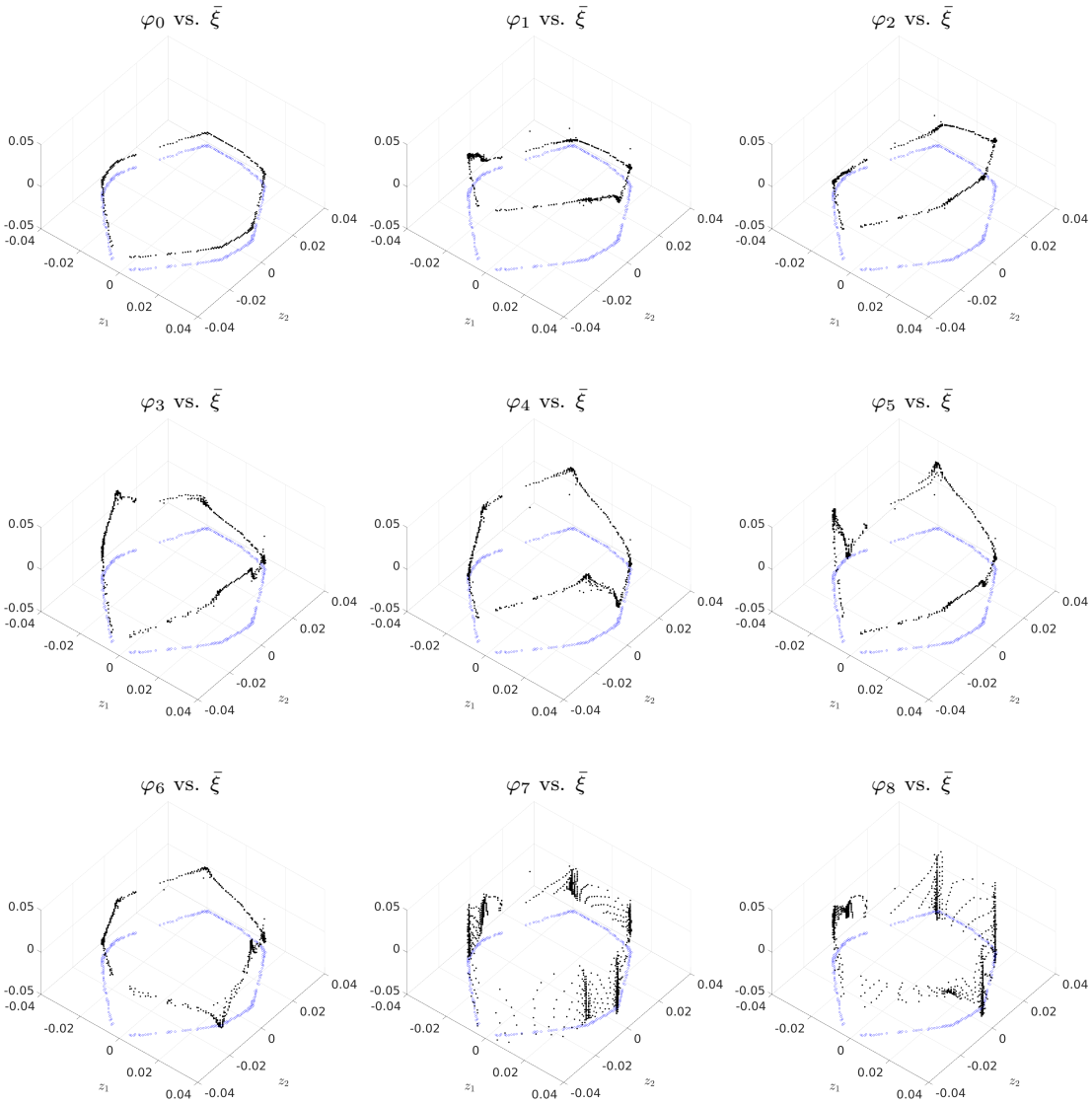



**Figure A.15:** Black dots: The values of the first nine eigenfunctions of $\mathcal{T}^t$ plotted against $\overline{\xi}(x_i)$, $x_i \in \overline{\mathbb{X}}$. The blue markers indicate the $\overline{\xi}(x_i)$ in the bottom plane. The seven dominant eigenfunctions ($\varphi_0$ to $\varphi_6$) seem to have a smooth dependency on $\overline{\xi}$. In contrast, the values of the non-dominant $\varphi_7$ and $\varphi_8$ vary substantially over individual level sets of $\overline{\xi}$.

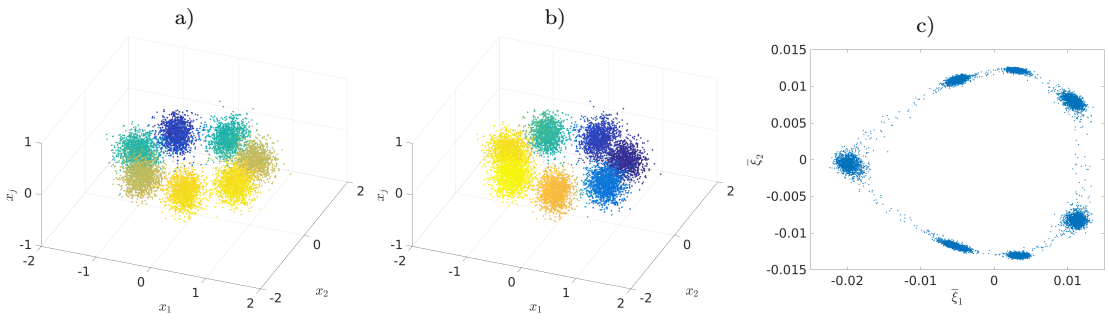



**Figure A.16:** a) & b) The two components $\overline{\xi}_1$ and $\overline{\xi}_2$ on the sampling points $\overline{\mathbb{X}}$. The picture shows the first three dimensions of $x$, but is qualitatively the same when replacing $x_3$ by $x_j$ , $j = 4, \ldots, 10$. c) The values of $\overline{\xi}_1$ and $\overline{\xi}_2$ on $\overline{\mathbb{X}}$ plotted against each other.

### A.5.3 Two quadruple well potentials

Our theory is based on the existence of an $r$-dimensional transition manifold $\mathbb{M}$ in $L^1(\mathbb{X})$ around which the transition probability functions concentrate. In Appendix A.B, we argued that the existence of an $r$-dimensional transition path suffices to ensure the existence of $\mathbb{M}$. Here we illustrate how the existence of the transition path is reflected in the embedding procedure.

For this we consider the "hilly" and "flat" quadruple well potentials

$$V_1(x) = (x_1^2 - 1)^2 + (x_2^2 - 1)^2 + 5\exp(-5(x_1^2 + x_2^2))$$

and

$$\begin{aligned} V_2(x) = 1 - \exp\big(-10\big((x_1-1)^2 + (x_2-1)^2\big)^2\big) - \exp\big(-10\big((x_1-1)^2 + (x_2+1)^2\big)^2\big) \\ - \exp\big(-10\big((x_1+1)^2 + (x_2+1)^2\big)^2\big) - \exp\big(-10\big((x_1+1)^2 + (x_2-1)^2\big)^2\big)\,. \end{aligned}$$

Both systems possess metastable sets around the four minima $(\pm 1, \pm 1)$, but $V_1$ confines its dynamics outside of the metastable sets onto a one-dimensional transition path, whereas $V_2$ does not impose such restrictions on the dynamics (see Figure A.17). For both potentials the time $t = 1$ lies inside the slow-fast time scale gap. Assuming a one-dimensional transition manifold (wrongfully for $V_2$), we use the three linear observables (A.29). A $40 \times 40$ grid on $[-2, 2] \times [-2, 2]$ is used as evaluation points for $\overline{\xi}$. The embedding of these points by $\mathcal{E} \circ \overline{\mathcal{Q}}$ can be seen in Figure A.18. We observe a one-dimensional structure in the case of the "hilly" potential $V_1$, whereas the embedding points of the "flat" potential $V_2$ lie on a seemingly two-dimensional manifold. As these embeddings are approximately one-to-one with the respective transition manifolds $\mathbb{M}$, we conclude that in the case of $V_1$ the manifold $\mathbb{M}$ must be one-dimensional, whereas for $V_2$ it is two-dimensional.

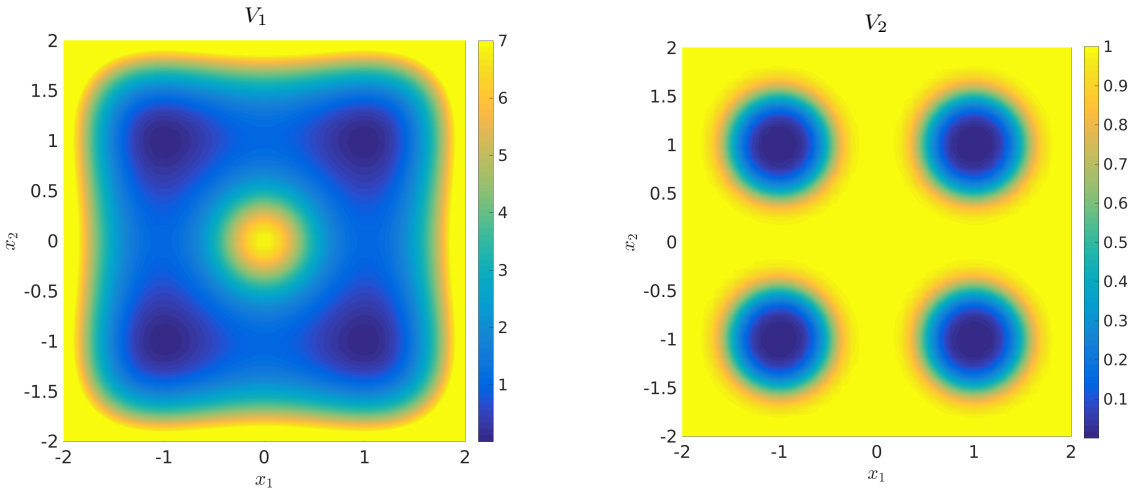



**Figure A.17:** The two quad-well potentials $V_1$ and $V_2$ possess qualitatively different transition regions.

## A.6 Conclusion

Our main contributions in this paper are:

(a) We developed a mathematical framework to characterize good reaction coordinates for stochastic dynamical systems showing metastable behavior but no local separa-

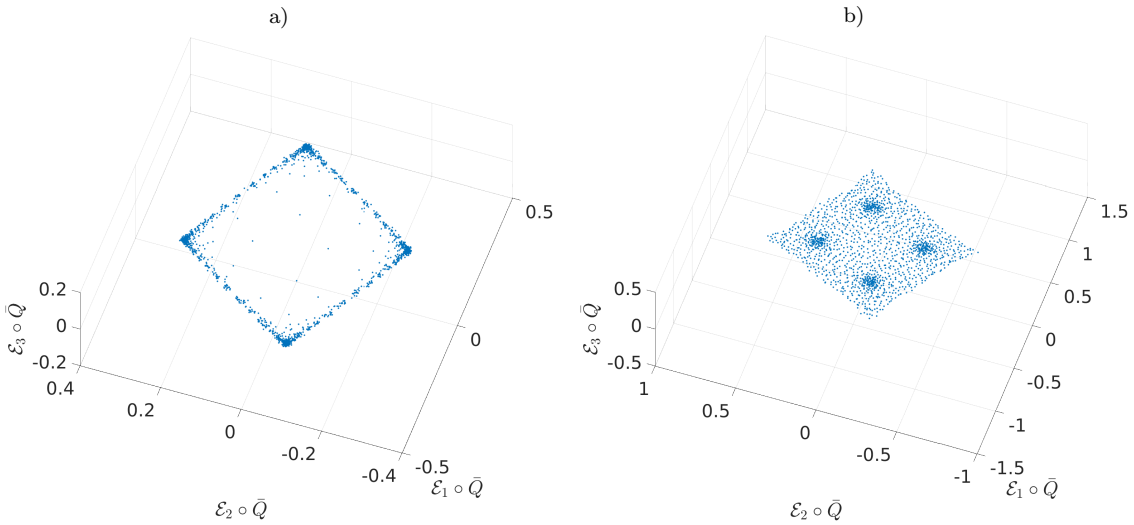



**Figure A.18:** Embedding of the grid points for the a) "hilly" and b) "flat" four well potential. A one-dimensional structure is only visible in a), i.e. in the presence of a one-dimensional transition path.

tion of fast and slow time scales.

(b) We showed the existence of good low-dimensional reaction coordinates under certain *dynamical* assumptions on the system.

(c) We proposed an algorithmic approach to numerically identify good reaction coordinates and the associated low-dimension transition manifold based on local evaluation of short trajectories of the system only.

Our numerical examples show how the procedure works, that it can be used in higher dimensions, and the examples give further evidence that the dynamical assumptions from (b) are valid in many realistic cases. The application of our approach to relevant biomolecular problems, e.g. in protein folding, is ongoing work.

Apart from the application to actual molecular systems, there are several open questions and challenges, which we will address in the future:

- A rigorous mathematical justification for the dynamical assumption in Definition A.4.4 in terms of the potential $V$ and the noise intensity $\beta^{-1}$ in (A.2) would be desirable. This seems to be a demanding task, as the interplay between potential landscape and the thermal forcing is nontrivial. For $\beta^{-1} \to 0$ the problem can be handled by large deviation approaches; however, understanding increasing $\beta^{-1}$ is challenging: the strength of noise increases, and additional transitions between metastable sets become more probable, as the barriers in the potential landscape become less significant, and thus the reaction coordinate may increase in dimension.

- Also related to the previous point, the choice of the correct lag time $t$ is crucial. Choosing the time too small, the concentration of the transition densities near a low-dimensional manifold in $L^1$ may not have happened yet, but a too large lag time has severe consequences for the numerical expenses. If no expert knowledge of a proper lag time $t$ is available, it has to be identified in a pre-processing step, for example using Markov State Model techniques [16].

- As discussed in the last part of Section A.4.3 and in Figure A.6, we need the embedding $\mathcal{E}$ not to distort transversality close to the transition manifold $\mathbb{M}$ too much, such that the realized reaction coordinate $\overline{\xi}$ is indeed a good one. Theoretical bounds shall be developed. This problem seems to be coupled with the problem of how to control the condition number of the embedding and its numerical realization.

- The dimension $r$ of the reaction coordinate may not be known in advance, hence we need an algorithmic strategy to identify this on the fly. Fortunately, once the sampling has been made, the evaluation of the embedding mapping $\mathcal{E}$, and finding intrinsic coordinates on the set of data points embedded in $\mathbb{R}^k$ has a negligible numerical effort, hence different embedding dimensions $k$ can be probed via (A.21). Theorem A.4.10 suggests that if the identified dimension of the reaction coordinate is smaller than $k/2$, then a reaction coordinate of sufficient dimension has been found.

- To benefit from the dimensionality reduction of the reaction coordinate $\xi$, the dynamics that generates the reduced transfer operator $\mathcal{T}_\xi^t$ has to be described in closed form. We are planing to employ techniques based on the Kramers–Moyal extension [199] to again receive an SDE for a stochastic process on $\mathbb{R}^r$.
- The embedding mapping $\mathcal{E}$ is evaluated by Monte Carlo quadrature (A.24). Although Monte Carlo quadrature is known to have a convergence rate independent of the underlying dimension $n$ of $\mathbb{X}$, there is still an impact of the dimension on the practical accuracy. This we shall investigate as well.

## Acknowledgements

This research has been partially funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114 "Scaling Cascades in Complex Systems", Project B03 "Multilevel coarse graining of multi-scale problems", and by the Einstein Foundation Berlin (Einstein Center ECMath).

## A.A Properties of $P_\xi$

*Proof of Proposition A.3.4.* (a) This property has been shown by Zhang [199] as well, we include the short reasoning for completeness. The linearity of $P_\xi$ is obvious. The property $P_\xi^2 = P_\xi$ follows from (A.8) by noting that $P_\xi f$ is constant on $\mathbb{L}_z$ and that $\mu_z$ is a probability measure for every $z$.

(b) From (A.10) we have for $f, g \in L^2_\mu(\mathbb{X})$ that

$$\begin{aligned} \langle P_\xi f, g\rangle_\mu &= \int_{\mathbb{X}} P_\xi f(x) g(x)\, d\mu(x) \\ &\overset{(A.10)}{=} \int_{\xi(\mathbb{X})} \Gamma(z) \widehat{P_\xi(g P_\xi f)}(z)\, dz \\ &\overset{(*)}{=} \int_{\xi(\mathbb{X})} \Gamma(z) \widehat{P_\xi f}(z) \widehat{P_\xi g}(z)\, dz\,, \end{aligned} \tag{A.30}$$

where $(*)$ follows from the linearity of $P_\xi$, and the fact that $P_\xi f|_{\mathbb{L}_{\xi(x)}} = \text{const}$, thus $\widehat{P_\xi(gP_\xi f)}(z) = \widehat{P_\xi f}(z)\widehat{P_\xi g}(z)$. Expression (A.30) is symmetric in $f$ and $g$, hence it follows that $\langle P_\xi f, g\rangle_\mu = \langle f, P_\xi g\rangle_\mu$.

(c) We first prove that $P_\xi$ is an orthogonal projection:

$$\langle P_\xi f, f - P_\xi f\rangle_\mu \overset{(b)}{=} \langle f, P_\xi f - P_\xi^2 f\rangle_\mu \overset{(a)}{=} \langle f, P_\xi f - P_\xi f\rangle_\mu = 0\,.$$

Thus,

$$\|f\|^2_{L^2_\mu} = \|f - P_\xi f\|^2_{L^2_\mu} + \|P_\xi f\|^2_{L^2_\mu} \geq \|P_\xi f\|^2_{L^2_\mu}\,,$$

and the claim follows.

□

## A.B On the existence of reaction coordinates

To motivate the existence of low-dimensional reaction coordinates, let us assume that the dynamics of consideration has $d+1$ metastable regions $\mathbb{C}_0, \dots, \mathbb{C}_d \subset \mathbb{X}$. Let $\mathbb{C} = \bigcup_i \mathbb{C}_i$. For a selected lag time $t > 0$ we make the following two assumptions:

1. Fast local equilibration: If $x$ is in (or close to) $\mathbb{C}_i$ then we have

   $$\mathcal{P}^t \delta_x \approx \varrho_i^{qs}$$

   where $\varrho_i^{qs}$ is the quasi-stationary density of the metastable core $\mathbb{C}_i$:

   $$\lim_{s\to\infty} \mathsf{P}\left[\mathbf{X}_s = y \,\middle|\, \boldsymbol{\tau}_{\mathbb{C}_i} > s\right] = \varrho_i^{qs}(y)dy$$

   with $\boldsymbol{\tau}_{\mathbb{C}_i}$ being the (random) exit time from the set $\mathbb{C}_i$.

2. Slow transitions: The typical transition time to reach $\mathbb{C} \setminus \mathbb{C}_i$ when starting in $\mathbb{C}_i$ is larger than $t$. In other words, $t$ is such, that if the process $\mathbf{X}_s$ transitions from $x$ to some $\mathbb{C}_i$, it did not transition through some other $\mathbb{C}_j$ with high probability.

These two assumptions essentially say that $t$ is much larger than the fast time scales of the system, but smaller than the dominant time scales. It follows that, for any $x \in \mathbb{X}$,

$$\mathcal{P}^t \delta_x \approx \sum_{i=0}^{d} q_i(x) \varrho_i^{qs}, \quad \sum_{i=0}^{d} q_i(x) = 1\,,$$

where by assumption 2) the coefficients $q_i(x)$ are given by the committor functions

$$q_i(x) = \mathsf{P}\left[\mathbf{X}_t \text{ reaches } \mathbb{C}_i \text{ before } \mathbb{C}\setminus\mathbb{C}_i \,\middle|\, \mathbf{X}_0 = x\right].$$

We say that $\mathcal{P}^t\delta_x$ is an $r$-dimensional structure in $L^1(\mathbb{X})$ if there is a function $\xi\colon \mathbb{X} \to \mathbb{R}^r$ that jointly parametrizes all the committor functions, i.e., $q_i = \tilde{q}_i \circ \xi$ with $\tilde{q}_i\colon \mathbb{R}^r \to \mathbb{R}$. If this is the case, then

$$\overline{\mathcal{Q}}(x) = \mathcal{P}^t\delta_x \approx \sum_{i=0}^{d} \tilde{q}_i(\xi(x))\varrho_i^{qs} =: \mathcal{Q}(x)$$

and clearly $\dim(\mathcal{Q}(\mathbb{X})) \le r$ since $\dim(\xi(\mathbb{X})) = r$. Moreover, $r \le d$ since we can explicitly construct $\xi\colon \mathbb{X} \to \mathbb{R}^d$ as $\xi = (q_1, \dots, q_d)$. This obviously parameterizes $q_1, \dots, q_d$, and it also parameterizes $q_0$ since $q_0 = 1 - \sum_{i=1}^{d} q_i$.

However, $r$ may also be smaller than $d$. As an example, consider the potential with 4 minima shown in Figure A.19 on the left. At low temperatures, the "hilly" potential energy landscape confines all transitions between the minima $\mathbb{C}_0, \dots, \mathbb{C}_3$ to a narrow region close to the red square connecting the four minima. Figure A.19 shows the level sets of $q_0$, the level sets of the other committors are given by the rotational symmetry

of the problem. All four committors can be jointly parameterized by a single coordinate $\xi$ which describes clockwise movement along the red square and is constant orthogonal to it. Therefore, $r = 1$. Figure A.19 on the right shows the situation with a "flat" energy landscape. Transition paths between the minima are no longer confined to a one-dimensional structure, and the committor level sets are more complicated. We can no longer parameterize all four committors with a single coordinate $\xi$, so $r > 1$. On the other hand, $\dim(\mathbb{X}) = 2$ so $r = 2$.

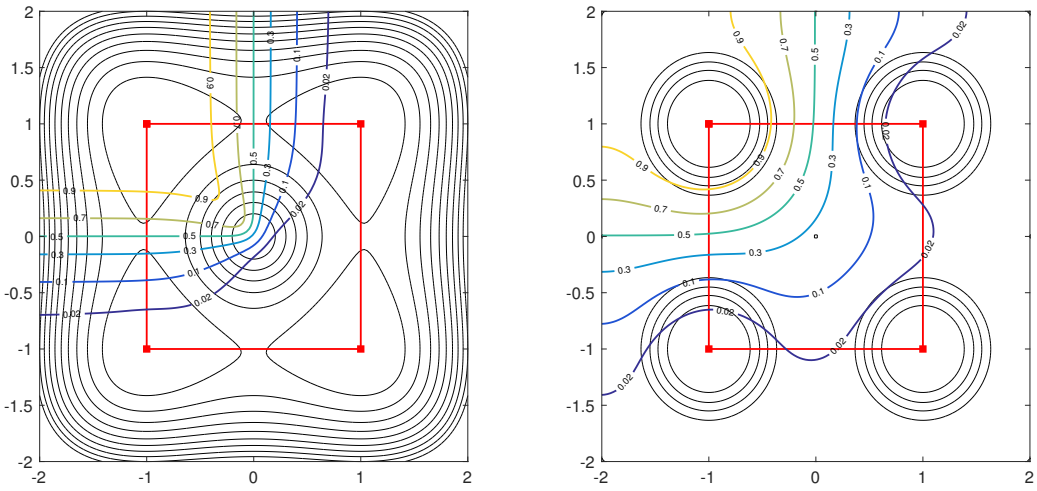


**Figure A.19:** A potential energy landscape with four minima (black contours) and level sets of $q_1$ (colored contours). Left: The "hilly" landscape structure confines transition pathways to a narrow region close to the red square connecting the four minima. As a result, all committor level sets are orthogonal to this main transition path. Right: "Flat" landscape structure with more complicated level sets of the committors.

This structural difference of the potentials can also be seen when applying our algorithm to construct the reaction coordinate $\overline{\xi}$, see Figure A.18 and Section A.5.3.