The Variational Approach to Conformational Dynamics

Dissertation

zur Erlangung des Grades eines Doktors der Naturwissenschaften

am Fachbereich Mathematik und Informatik der Freien Universität Berlin

vorgelegt von

Feliks Nüske

Berlin, 2017

Erstgutachter:
Prof. Dr. Frank Noé
Freie Universität Berlin
Fachbereich Mathematik und Informatik
Arnimallee 6, 14195 Berlin

Zweitgutachter:
Prof. Dr. Reinhold Schneider
Technische Universität Berlin
Institut für Mathematik
Straße des 17. Juni 136, 10623 Berlin

Tag der Disputation: 17. Februar 2017

Contents

Chapter 1

Introduction

This thesis is about the *variational approach to conformational dynamics* (VAC), a method to extract essential information about stationary and kinetic properties from simulations of high-dimensional stochastic dynamics. We will develop the theory for ergodic and reversible Markov processes, and we will use molecular dynamics (MD) simulations of biological macromolecules as the guiding application, but the method can be applied in a much broader context.

Molecular dynamics is a family of powerful simulation protocols to explore the thermodynamics and kinetics of biological macromolecules, such as protein or enzymes [1, 2]. Here, one or a few macromolecules of interest are simulated at atomistic resolution, that is, each atom is represented as a point particle. Often, the molecule is placed into a simulation box filled with water molecules which are also represented at atomistic resolution. The system's time evolution is modeled by a stochastic dynamical system, typically some Itô stochastic differential equation like Langevin dynamics, where the drift is determined by the gradient of an empirical potential energy [3]. The details vary for each individual application, but the important point is that MD simulations sample from a continuous time Markov process in a high-dimensional space. The dimension equals the total number of atomic coordinates and possibly momenta, which can be on the order of ten thousands or even hundred thousands degrees of freedom.

A key feature of molecular simulation data is *metastability*. The integration time step in MD simulations is limited to only a few femtoseconds (the standard value is two femtoseconds), which reflects the timescales of fastest vibrations within the molecule. However, it can often be observed that the overall geometrical shape of the molecule is preserved for long times in the simulation, during which the dynamics only samples small scale fluctuations. Only rarely, a major transition into a different geometrical shape can be observed. These long-lived geometrical shapes are called *metastable conformations*, and the process of exchanging between them is called conformational dynamics [4]. Metastable transitions often correspond directly to biologically relevant events, like protein folding or binding / unbinding of a ligand. The timescales of transitions between metastable conformations frequently exceed the integration time step by many orders of magnitude, and can be on the orders of microseconds or milliseconds or even beyond.

Two challenges arise from this observation. Firstly, MD simulations pro-

duce huge amounts of highly redundant data. Automatic analysis methods that can extract the essential information from the data are needed to make use of the simulations. Such a method should be able to detect metastable states, the timescales of transitions in between them, and provide a low-dimensional representation of the data that captures the essential dynamics. Secondly, statistically reliable sampling of all relevant metastable states and transitions by a single long equilibrium simulation is infeasible for large and complex biological systems, even with modern supercomputers. Therefore, methods are needed that can extract the essential information from a large ensemble of fairly short simulations, that can be produced efficiently on parallel or distributed computer architectures, but must be expected to sample from a non-equilibrium distribution.

Going back to the pioneering works of Davies [5, 6] as well as Dellnitz and Junge [7], it is known that metastability in Markov processes gives rise to eigenvalues of the associated transfer operator which are close to the Perron root $\lambda_1 = 1$. While these eigenvalues provide information about characteristic transition timescales between metastable states, the corresponding eigenfunctions can be used to identify metastable states in space. The eigenfunctions can be expected to be almost constant within each metastable state, but display changes of sign across the transition regions in between [8]. Dominant eigenfunctions of the transfer operator thus provide an optimal low-dimensional representation of the essential dynamics.

Over the past 15 years, *Markov state models (MSMs)* [9, 10, 11] have become a standard tool for the analysis of molecular dynamics simulation data. An MSM provides a simplified model of the original Markov process by a discrete time Markov chain on finitely many states. These states are defined by partitioning the continuous state space into finitely many disjoint sets. Time is discretized by choosing a discrete time step, called the *lag time*, and the full process is replaced by a snapshot process that only keeps track of the discrete state visited at the discrete time steps. Any time information in between and any spatial information within the discrete sets is discarded. This Markov chain is fully described by a stochastic transition matrix. MSMs provide an approximation to the transfer operator and its dominant eigenfunctions in terms of piecewise constant functions on the set partitioning [12], and therefore agree with what is known as Ulam's method in the study of dynamical systems [13]. MSMs are easily estimated; basically, only the number of transitions between all pairs of discrete states must be counted over all available simulation data. More advanced estimators have also been developed [14, 15], while extensive analysis tools can detect metastable states [16, 17], transition paths [18], uncertainties of derived quantities [19, 20, 15] and relations to experimental data [21, 22]. Moreover, it is often possible to estimate an MSM from an ensemble of many short trajectories [23], although we will discuss this in more detail in chapter 6. Recalling the challenges introduced above, MSMs are a powerful tool for the analysis of molecular simulation data. It is no surprise then that Markov state models have been successfully applied in numerous studies [24, 25, 14, 23, 26, 27].

One of the main difficulties for the construction of MSMs is the determination of a suitable state space partitioning. A widely used procedure is to first apply basic dimensionality reduction methods, like time-lagged independent component analysis (TICA) [28, 29, 30], to project the data onto a smaller

number of informative coordinates. In the second step, geometric clustering algorithms, such as kmeans, are used to cluster the projected data and thus determine the discrete states. The result is a Voronoi partition in high-dimensional space that often lacks a physical interpretation. Moreover, it is generally unclear how a clustering of the simulation data can be found that separates metastable states well. If we recall that an MSM approximates the dominant eigenfunctions from a basis of piecewise constant functions, we see that it is particularly important to separate metastable states and to resolve transition regions well by means of the discretization [12].

In this thesis, we present a general method that extracts the essential information from molecular simulation data by approximating the dominant eigenfunctions from a pre-selected library of basis functions. These basis functions can be chosen to carry a physical meaning. They can, for instance, encode basic configurational changes along simple molecular coordinates, like distances or angles, thus allowing for an interpretation of the results. We start by reviewing the theory of Markov processes and their associated transfer operators in chapter 2, as well as the relation between metastability and dominant spectral components. Next, in chapter 3, we introduce the variational theorem 3.1 from Ref. [31] and explain how it gives rise to the generalized eigenvalue problem Eq. (3.1.7) that must be solved in order to approximate dominant spectral components from a given subspace of basis functions. This problem is the centerpiece of our study. We show how the matrices appearing in Eq. (3.1.7) can be estimated from simulation data, and we also explain how both MSMs and TICA arise as special cases. In chapter 4, we illustrate the method using toy systems and small molecular systems, and provide the proof of concept that physically interpretable basis sets can indeed be advantageous compared to set discretizations. In chapter 5, we go on to discuss how basis functions defined on elementary molecular coordinates (we will call them one-coordinate functions) can be used to model complex dynamical processes, that must be expected to be non-linear functions of the molecular coordinates. We suggest to use tensor products of one-coordinate functions as a basis set. This formulation allows to use a large and very general library of basis functions, and thus removes the dependence of the method on pre-processing steps or a priori knowledge of the system under investigation. However, the tensor product formulation leads to an explosion of the basis set size - the curse of dimensionality. Consequently, we investigate the use of a low-rank tensor representation, the *tensor-train-(TT)-format* [32, 33], and its learning algorithm *alternating linear scheme (ALS)* [34], in order to circumvent the dimensionality problem. We formulate a new optimization problem that helps to apply the existing algorithm in our setting, and provide two benchmark applications as a proof of concept. In the final chapter 6, we discuss the use of short non-equilibrium simulations in order to address the second challenge mentioned above. Markov state models can often be constructed from such data [23]. The intuitive argument is that only conditional transition probabilities need to be estimated to build the transition matrix. It follows that only a local equilibrium within each state is required in order to obtain an unbiased estimate of transition probabilities [35, 36, 37, 38]. However, this local equilibrium can only be approximately achieved in practical simulations, and it has remained an open question how to verify the approximate local equilibrium in practice, or how to deal with deviations in an optimal way. We systematically study the error due to deviations

from equilibrium, and show how it depends on the initial conditions, the lag time, and the state definition. We proceed to explain how the framework of *observable operator models* (OOMs) [39, 40, 41] can be used to obtain unbiased estimates of transition probabilities from non-equilibrium simulations. The OOM formulation only relies on a finite-rank assumption on the transfer operator, that can often be assumed to be approximately fulfilled in practice. OOM estimation methods can also provide exact estimates of the slowest eigenvalues of the transfer operator, thus enabling us to assess the approximation quality of an MSM. We discuss several algorithmic details and present applications to model systems. Although we focus on MSMs in chapter 6, we conclude by explaining how the OOM-based estimation method can be generalized to arbitrary basis functions in the VAC. Application of this formulation will be the subject of future work.

Chapter 2

Markov Processes and Transfer Operators

In this chapter, we review some of the theory of Markov processes, transfer operators and their dominant spectrum.

2.1 Markov Processes

We are concerned with *Markov processes* X_t in continuous time and space. A Markov process is a random process where the future behaviour is only dependent on the present state. In order to introduce such a process formally, we start with a general stochastic process. We assume there is a state space S, which is the set of all possible states the process can visit. In this work, we will think of S being a subset of Euclidean space $\mathbb{R}^n$, but the theory remains valid in much more general situations. Let us denote the Borel σ-algebra on S by $\mathfrak{S}$, and let $(\Omega, \Sigma, \mathbb{P})$ be a probability space. A stochastic process is a family of random variables X_t , defined for each t in some index set I, that map Ω into S:

$$\omega \in \Omega \quad \rightarrow \quad X_t(\omega) \in S. \tag{2.1.1}$$

We will restrict ourselves to the situation where the index set is the non-negative real line, $I = \mathbb{R}_{\geq 0}$, such that the index t can be interpreted as time. In view of this, we call the map

$$t \quad \rightarrow \quad X_t(\omega), \tag{2.1.2}$$

for fixed ω, a *path*, a *trajectory* or a *realization* of the process X_t. The probability space Ω can be chosen to be the space of trajectories S^I, i.e. the space of mappings $\omega : \mathbb{R}_{\geq 0} \rightarrow S$, and the random variables X_t are the projections of a trajectory ω onto its value at time t. In this scenario, the probability measure $\mathbb{P}$ determines the probabilities of complete realizations.

Stochastic processes can be specified by their *finite-dimensional distributions*. Let $J = \{t_1, \ldots, t_l\} \subset I$, $t_1 \leq \ldots \leq t_l$ denote any finite collection of time indices. Assume that for every collection J, $\mathbb{P}_J$ is a probability measure on the

σ-algebra generated by sets of the form $A_1 \times A_2 \times \ldots \times A_l$, where the A_i are sets in $\mathfrak{S}$. Then, under mild conditions on the $\mathbb{P}_J$, the consistency theorem of Kolmogorov and Daniell [42, Cor. 35.4] guarantees the existence of a stochastic process with probability measure $\mathbb{P}$, such that

$$\mathbb{P}_J(A_1 \times \ldots \times A_l) \quad = \quad \mathbb{P}(X_{t_1} \in A_1, \ldots X_{t_l} \in A_l). \tag{2.1.3}$$

Thus, a stochastic process is determined by the probabilities of all finite observation sequences as on the right hand side of Eq. (2.1.3).

A Markov process is a stochastic process that does not remember its past, only its present state. This can be formalized by requiring that for all choices of $s_1 < \ldots < s_l < t$, we have

$$\mathbb{P}(X_t \in A | X_{s_1}, \ldots, X_{s_l}) \quad = \quad \mathbb{P}(X_t \in A | X_{s_l}), \tag{2.1.4}$$

see [42, Lem. 42.4]. Strictly speaking, the conditional probabilities in Eq. (2.1.4) are conditional expectations. The centerpiece for the definition of a Markov process is a stochastic transition kernel:

Definition 2.1. A family of maps $(p_t)_{t \geq 0}$, $p_t : S \times \mathfrak{S} \to [0, 1]$ is called a *semigroup* of stochastic transition kernels, or a *Markov semigroup*, if

1. For each $x \in S$, the map $p_t(x, \cdot)$ is a probability measure on $\mathfrak{S}$.
2. For each $A \in \mathfrak{S}$, the map $p_t(\cdot, A)$ is $\mathfrak{S}$-measurable.
3. The *Chapman-Kolmogorov equation* is fulfilled for all s, $t \geq 0$:

$$p_{s+t}(x, A) \quad = \quad \int_S p_s(x, \mathrm{d}y) p_t(y, A). \tag{2.1.5}$$

4. For $t = 0$, the measure $p_0(x, \cdot)$ is the Dirac measure concentrated at x for all x.

The family of stochastic transition kernels and an initial distribution determine a Markov process:

Definition 2.2. For a Markov semigroup p_t and a probability measure ν_0 on S, define a stochastic process via the finite-dimensional distributions

$$\mathbb{P}_J(A) \quad := \quad \int_S \ldots \int_S \chi_A(x_1, \ldots, x_l) p_{t_l - t_{l-1}}(x_{l-1}, \mathrm{d}x_1) \ldots p_{t_1}(x_0, \mathrm{d}x_1) \nu_0(\mathrm{d}x_0), \tag{2.1.6}$$

where J is the collection of time indices $t_1 < \ldots < t_l$, χ denotes the indicator function of a set, and $A = A_1 \times \ldots \times A_l$, $A_i \in \mathfrak{S}$.

Theorem 2.3. *The finite-dimensional distributions Eq. (2.1.6) define a stochastic process X_t which fulfills the Markov property Eq. (2.1.4).*

For a proof of this theorem, see [42, Thm. 36.4 and Thm. 42.3]. In order to emphasize that the probability measure $\mathbb{P}$ of the resulting stochastic process was generated by choosing the initial distribution ν_0, it can also be denoted by $\mathbb{P}_{\nu_0}$. If we consider the space $\Omega_{\Delta t}$ of discrete trajectories $(X_{k\Delta t})_{k=0}^{\infty}$, generated by sampling continuous time trajectories X_t at discrete time intervals $k\Delta t$, $\Delta t > 0$, then the measure $\mathbb{P}$ induces a probability measure $\mathbb{P}_{\Delta t}$ on the corresponding σ-algebra $\Sigma_{\Delta t}$ by restricting the finite-dimensional distributions:

$$\mathbb{P}_{\Delta t}(X_{k_1\Delta t} \in A_1, \ldots, X_{k_l\Delta t} \in A_l) \quad = \quad \mathbb{P}(X_{k_1\Delta t} \in A_1, \ldots, X_{k_l\Delta t} \in A_l). \tag{2.1.7}$$

2.2 Transfer Operators

Usually, one is interested in the dynamics of an ensemble of realizations, not of a single trajectory. If a large (ideally infinite) ensemble of trajectories is initially distributed according to ν_0, Equation (2.1.6) provides an expression for the distribution ν_t at time $t > 0$:

$$\nu_t(A) \quad = \quad \int_A \int_S p_t(x_0, \mathrm{dx_t})\nu_0(\mathrm{dx}_0) \tag{2.2.1}$$

$$\quad = \quad \int_S p_t(x_0, A)\nu_0(\mathrm{dx}_0). \tag{2.2.2}$$

We restrict our attention to initial distributions which are absolutely continuous w.r.t. some fixed measure μ on the state space S, that is, there is a density function ρ_0 such that

$$\nu_0(A) \quad = \quad \int_A \rho_0(x)\mu(\mathrm{dx}). \tag{2.2.3}$$

In this case, Eq. (2.2.2) transforms into

$$\nu_t(A) \quad = \quad \int_S p_t(x_0, A)\rho_0(x_0)\mu(\mathrm{dx}_0). \tag{2.2.4}$$

If the distribution ν_t also possesses a density ρ_t w.r.t. μ, then Eq. (2.2.4) reads:

$$\int_A \rho_t(x)\mu(\mathrm{dx}) \quad = \quad \int_S p_t(x, A)\rho_0(x)\mu(\mathrm{dx}),\ \forall A \in \mathfrak{S}. \tag{2.2.5}$$

Under mild conditions, such a density can be proven to exist. In fact, Eq. (2.2.5) determines a linear operator which can be extended from densities to functions of the space L^1_μ, that is, the space of absolutely integrable functions w.r.t. the measure μ. The proof mainly relies on the Radon-Nikodym-Theorem, see Ref. [43, Chap. 3.2.] for details:

Theorem 2.4. *If the transition kernel p_t satisfies that whenever $\mu(A) = 0$ for some $A \in \mathfrak{S}$, it follows that $p_t(x, A) = 0$ μ-a.e. in x, there is a linear operator $\mathcal{T}_t : L^1_\mu \to L^1_\mu$, satisfying for all $A \in \mathfrak{S}$:*

$$\int_A \left[\mathcal{T}_t f\right](x)\mu(\mathrm{dx}) \quad = \quad \int_S p_t(x,A)f(x)\mu(\mathrm{dx}). \tag{2.2.6}$$

There are different names for the hypothesis of Theorem 2.4, following Ref. [43], we will call it *non-singularity* of the transition kernel w.r.t. μ. The linear operator $\mathcal{T}_t$ is called the *transfer operator*, and it is a key ingredient for all that follows. We list some important properties [43]:

Proposition 2.5. *(i) The transfer operator is a Markov operator, that is, $\mathcal{T}_t f \geq 0$ if $f \geq 0$, and $\|\mathcal{T}_t f\|_1 = \|f\|_1$ if $f \geq 0$.*

(ii) The transfer operators for different time indices form a semigroup of operators, i.e.

$$\mathcal{T}_{s+t} \quad = \quad \mathcal{T}_t\mathcal{T}_s,\ s,t \geq 0. \tag{2.2.7}$$

In particular, $\mathcal{T}_0$ equals the identity operator.

Before proceeding, we introduce a closely related operator that also carries an intuitive meaning [43].

Definition 2.6. The adjoint operator $\mathcal{K}_t : L^\infty_\mu \to L^\infty_\mu$ of the transfer operator is called the *Koopman operator*.

Proposition 2.7. *The Koopman operator acts on functions $g \in L^\infty_\mu$ by*

$$\mathcal{K}_t g(x) \quad = \quad \int_S p_t(x,\mathrm{dy})g(y) \tag{2.2.8}$$

$$\quad = \quad \mathbb{E}_x\left[g(X_t)\right]. \tag{2.2.9}$$

where $\mathbb{E}_x$ denotes expectation with respect to the measure $\mathbb{P}_x$ generated by the initial distribution δ_x, i.e. the process is started deterministically at $X_0 = x$.

2.3 Invariant Measures and Reversibility

Invariant measures are of particular importance in the study of Markov processes. A measure ν is called *invariant* if it is unchanged under the dynamics, that is, if the initial probability distribution ν_0 equals ν, the measure ν_t in Eq. (2.2.2) is also equal to ν. If ν is absolutely continuous w.r.t. the measure μ defining the transfer operator $\mathcal{T}_t$, invariance can be formulated in terms of $\mathcal{T}_t$:

Lemma 2.8. *Let ρ be the density function of an invariant measure which is absolutely continuous w.r.t. μ. Then ρ satisfies*

$$\mathcal{T}_t\rho \quad = \quad \rho. \tag{2.3.1}$$

Proof. This follows directly from Eqs. (2.2.5-2.2.6). □

Henceforth, we will assume that there is a unique invariant measure of the process X_t. In molecular applications, this assumption is usually satisfied, and the invariant measure is defined by the Boltzmann distribution. We will also assume that the transfer operator is defined in terms of the unique invariant measure, i.e. μ is invariant from now on. It is easy to show that invariant measures are non-singular, and Lemma 2.8 implies that in this case, we have $\mathcal{T}_t 1 = 1$, where 1 is the constant function on S. In many cases, the invariant measure possesses a density with respect to the Lebesgue measure. We will always call this density π. We also use the notation

$$\langle f, g\rangle_\mu \quad = \quad \int_S f(x)g(x)\mu(\mathrm{dx}) \tag{2.3.2}$$

for the duality bracket between $f \in L^p_\mu$, $g \in L^q_\mu$, $\frac{1}{p} + \frac{1}{q} = 1$, or

$$\langle f, g\rangle_\pi \quad = \quad \int_S f(x)g(x)\pi(x)\mathrm{dx}, \tag{2.3.3}$$

if the invariant measure possesses a density π.

The transfer operators based on invariant measures are well-defined on all Lebesgue spaces L^p_μ, $1 \leq p \leq \infty$. As the proof of this statement can be hard to find in the literature, we repeat it appendix C.1, see also Ref. [44]. The restriction of the transfer operator to other Lebesgue spaces is particularly useful for *reversible* systems:

Definition 2.9. The transition kernel p_t is called reversible with respect to the measure μ if for all A, $B \in \mathfrak{S}$:

$$\int_A p_t(x, B)\mu(\mathrm{dx}) \quad = \quad \int_B p_t(x, A)\mu(\mathrm{dx}). \tag{2.3.4}$$

$$\mathbb{P}_\mu(X_0 \in A, X_t \in B) \quad = \quad \mathbb{P}_\mu(X_0 \in B, X_t \in A). \tag{2.3.5}$$

The second formulation of reversibility in Eq. (2.3.5) shows that there is no preferred pathway or direction in the system. For molecular systems, reversibility reflects that the system is in thermodynamic equilibrium. Choosing $B = \Omega$ in Eq. (2.3.5) implies that μ is automatically invariant if p_t is reversible w.r.t. μ. The following can be proven using Eq. (2.3.4) [45, Prop. 1.1.]:

Proposition 2.10. *The transition kernel p_t is reversible w.r.t. μ if and only if, for all $f, g \in L^2_\mu$,*

$$\langle \mathcal{T}_t f, g\rangle_\mu \quad = \quad \langle f, \mathcal{T}_t g\rangle_\mu. \tag{2.3.6}$$

For a reversible system, the transfer operator can be studied as a self-adjoint operator on the Hilbert space L^2_μ.

2.4 Spectral Properties of Transfer Operators

We focus on the transfer operator at some specific time step $\tau > 0$, called the *lag time*. Many interesting properties of transfer operators are directly linked to its spectrum, especially to its dominant spectrum, i.e. to eigenvalues close to one. The following two assumptions on the spectrum are typically made [45, 31]. Ref. [45] also provides conditions to guarantee these assumptions are satisfied.

1. The eigenvalue $\lambda = 1$ is simple, i.e. of multiplicity one, and dominant, that is, it is the only eigenvalue on the complex unit circle.
2. The essential spectral radius of the transfer operator is smaller than one, i.e. there is a ball of radius $r < 1$ such that any element of the spectrum outside this ball is a discrete eigenvalue.

For a reversible transfer operator on L^2_μ, assumptions 1 and 2 imply that the spectrum is composed of a number of discrete eigenvalues $1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_M$, and the remaining spectrum is contained in an interval $[0, r]$, $r < \lambda_M$ [31]:

$$\sigma(\mathcal{T}_\tau) \quad \subset \quad [0, r] \cup \{\lambda_M, \ldots, \lambda_2, 1\}. \tag{2.4.1}$$

If we wish to emphasize the lag time dependence of the eigenvalues, we also write $\lambda_m(\tau)$. By self-adjointness of $\mathcal{T}_\tau$, the eigenvalues $\lambda_1, \ldots, \lambda_M$ come along with mutually orthonormal eigenfunctions $\psi_1, \ldots \psi_M$, where $\psi_1 \equiv 1$ is the constant function on S. If there is a stationary density π, we will sometimes also refer to weighted eigenfunctions

$$\varphi_m(x) \quad = \quad \pi(x)\psi_m(x). \tag{2.4.2}$$

In analogy to reversible Markov chains, the functions φ_m are sometimes called *left eigenfunctions* while ψ_m are called *right eigenfunctions*.

The first assumption expresses that there is a unique invariant measure and that every initial density converges to the invariant density by the action of the transfer operator. It is shown in Ref. [45, Cor. 4.22.] that assumption 1 is equivalent to

$$\|\mathcal{T}_{n\tau}\rho - 1\|_1 \quad \leq \quad Mq^n, \tag{2.4.3}$$

for some constants $M > 0$, $q < 1$ and any density $\rho \in L^1_\mu$.

The spectrum close to one is related to the concept of *metastability*, which is a key feature of Markov processes arising in molecular simulation. As outlined in the introduction, we expect to find regions of the state space such that transitions between two such regions over lag time τ are rare-events unless τ is very large. Introducing the transition probability from set A to B (A, $B \in \mathfrak{S}$) over time τ,

$$p^\tau(A, B) \quad := \quad \frac{1}{\mu(A)} \int_A p(x, B)\mu(\mathrm{d}x) \tag{2.4.4}$$

$$= \quad \frac{\langle \mathcal{T}_\tau \chi_A, \chi_B \rangle_\mu}{\langle \chi_A, \chi_A \rangle_\mu}, \tag{2.4.5}$$

a set A is called metastable (on the timescale τ) if $p^\tau(A, A) \approx 1$. Following Ref. [31], the metastability of a decomposition of state space S into M non-overlapping sets $S_1, \ldots, S_M$, $S = \bigcup_{m=1}^M S_m$, is defined as the sum of conditional self-transition probabilities:

$$\sum_{m=1}^M p^\tau(S_m, S_m) \quad = \quad \sum_{m=1}^M \frac{\langle \mathcal{T}_\tau \chi_{S_m}, \chi_{S_m} \rangle_\mu}{\langle \chi_{S_m}, \chi_{S_m} \rangle_\mu}. \tag{2.4.6}$$

The relation of metastability to eigenvalues close to one goes back to the work of Davies [5, 6]. It was studied in the context of dynamical systems by Dellnitz and Junge [7] and in the context of molecular dynamics by Deuflhard and Schütte [4, 8]. The following upper and lower bounds for the metastability of a decomposition have been established by [31, Cor. 3]:

Theorem 2.11. *For a reversible transfer operator $\mathcal{T}_\tau$ such that assumptions 1 and 2 are satisfied, i.e. the spectrum satisfies Eq. (2.4.1), the metastability of any decomposition $S_1, \ldots, S_M$ is bounded from above and from below by*

$$\sum_{m=1}^M c_m \lambda_m \quad \leq \quad \sum_{m=1}^M p^\tau(S_m, S_m) \tag{2.4.7}$$

$$\leq \quad \sum_{m=1}^M \lambda_m, \tag{2.4.8}$$

where $0 \leq c_m \leq 1$ are given by

$$c_m \quad = \quad \sum_{i=1}^M \frac{\langle \psi_m, \chi_{S_i} \rangle_\mu^2}{\langle \chi_{S_i}, \chi_{S_i} \rangle_\mu}. \tag{2.4.9}$$

If there exist M sets S_m which are metastable on the timescale τ, then

$$p^\tau(S_m, S_m) \quad \approx \quad 1 \tag{2.4.10}$$

for all m, and by the upper bound, there must be M eigenvalues close to one. The lower bound reflects another important observation: the lower bound gets sharper if all projections c_m are close to one. For a metastable decomposition, we can expect the eigenfunctions ψ_m to be almost constant on each of the sets S_m, see again Refs. [6, 8].

Chapter 3

The Variational Formulation

We have seen that in order to analyze the stationary and slow dynamical properties of reversible Markov processes, we need to approximate its dominant eigenvalues and eigenfunctions. In this chapter, we introduce a variational formulation for the leading eigenvalues of the transfer operator and derive an approximation procedure from a finite dimensional space of basis functions. We also show how the matrices arising in this approximation method can be estimated from simulations of the process. The results we state were originally presented in Ref. [46].

3.1 The Variational Theorem

In order to approximate the dominant eigenfunctions ψ_m and eigenvalues λ_m of the transfer operator $\mathcal{T}_\tau$ numerically, we follow one of the standard approaches in numerical mathematics by fixing some finite dimensional subspace $D \subset L^2_\mu$, and try to approximate the eigenfunctions optimally within the subspace. The following result shows what optimality means in this context [47, 31]:

Theorem 3.1. *Let $1 = \lambda_1 > \lambda_2 \geq \ldots \geq \lambda_M$ be the dominant eigenvalues of the reversible transfer operator on L^2_μ. Then*

$$\sum_{m=1}^{M} \lambda_m = \sup \sum_{m=1}^{M} \langle \mathcal{T}_\tau f_m, f_m \rangle_\mu, \tag{3.1.1}$$

$$\langle f_m, f_{m'} \rangle_\mu = \delta_{m,m'}. \tag{3.1.2}$$

The sum of the first M eigenvalues maximizes the Rayleigh trace, which is the sum on the right hand side of Eq. (3.1.1) over all selections of M orthonormal functions f_m. The maximum is attained for the first M eigenfunctions $\psi_1, \ldots, \psi_M$.

Proof. The M-dimensional space D spanned by the functions f_m must contain an element g_M which is orthonormal to the first $M-1$ eigenfunctions ψ_j, i.e. $\langle g_M, \psi_j \rangle_\mu = 0$, $j = 1, \ldots, M-1$. By the standard Rayleigh principle for self-adjoint operators [48]

$$\langle \mathcal{T}_\tau g_M, g_M \rangle_\mu \quad \leq \quad \lambda_M. \tag{3.1.3}$$

Next, determine a normalized element g_{M-1} of the orthogonal complement of g_M in D, s.t. $\langle g_{M-1}, \psi_j \rangle_\mu = 0$, $j = 1, \ldots, M-2$. Again, we can invoke the Rayleigh principle to find

$$\langle \mathcal{T}_\tau g_{M-1}, g_{M-1} \rangle_\mu \quad \leq \quad \lambda_{M-1}. \tag{3.1.4}$$

Repeating this argument another $M-2$ times provides an orthonormal basis $g_1, \ldots, g_M$ of the space D such that

$$\sum_{m=1}^{M} \langle \mathcal{T}_\tau g_m, g_m \rangle_\mu \quad \leq \quad \sum_{m=1}^{M} \lambda_m. \tag{3.1.5}$$

As the Rayleigh trace is independent of the choice of orthonormal basis for the subspace D, and the space itself was arbitrary, this proves Eqs. (3.1.1-3.1.2). Clearly the maximum is attained for the first M eigenfunctions. □

In order to find the optimal approximations from within a given N-dimensional subspace D, where $N \geq M$, we restrict the Rayleigh trace to the space D. We need to find M orthonormal linear combinations from the subspace D such that the Rayleigh trace is maximal. If the space D is spanned by N linearly independent functions $f_1, \ldots, f_N$, and $\mathbf{a} \in \mathbb{R}^N$ is the coefficient vector of a function in D, we will call this function $f_\mathbf{a}$:

$$f_\mathbf{a} \quad = \quad \sum_{i=1}^{N} \mathbf{a}(i) f_i. \tag{3.1.6}$$

The restricted optimization problem is solved as described by the following proposition, see e.g. [49]:

Proposition 3.2. *Let D be a space of N linearly independent ansatz functions f_i, $i = 1, \ldots, N$. The set of $M \leq N$ mutually orthonormal functions $f_{\mathbf{a}_m}$, $m = 1, \ldots, M$ which maximize the Rayleigh trace restricted to D, is given by the first M eigenvectors of the generalized eigenvalue problem*

$$\mathbf{C}^\tau \mathbf{a}_m \quad = \quad \hat{\lambda}_m \mathbf{C}^0 \mathbf{a}_m, \tag{3.1.7}$$

where the matrices $\mathbf{C}^\tau$, $\mathbf{C}^0$ are given by

$$\mathbf{C}^\tau(i,j) \quad = \quad \langle \mathcal{T}_\tau f_i, f_j \rangle_\mu \tag{3.1.8}$$

$$\mathbf{C}^0(i,j) \quad = \quad \langle f_i, f_j \rangle_\mu. \tag{3.1.9}$$

Proof. First, note that for any functions $f_{\mathbf{a}}$ and $f_{\mathbf{b}}$, we have that

$$\langle \mathcal{T}_\tau f_{\mathbf{a}}, f_{\mathbf{b}} \rangle_\mu = \mathbf{a}^T \mathbf{C}^\tau \mathbf{b}, \tag{3.1.10}$$

$$\langle f_{\mathbf{a}}, f_{\mathbf{b}} \rangle_\mu = \mathbf{a}^T \mathbf{C}^0 \mathbf{b}. \tag{3.1.11}$$

Let us assume that the ansatz functions are mutually orthonormal, i.e. $\mathbf{C}^0 = \mathbf{Id}$. Then, maximization of the Rayleigh trace is equivalent to finding M vectors $\mathbf{a}_m$, such that $\mathbf{a}_m^T \mathbf{a}_{m'} = \delta_{m,m'}$ and

$$\sum_{m=1}^{M} \mathbf{a}_m^T \mathbf{C}^\tau \mathbf{a}_m = \sum_{m=1}^{M} \langle \mathbf{C}^\tau \mathbf{a}_m, \mathbf{a}_m \rangle_{\mathbb{R}^N} \tag{3.1.12}$$

is maximal (the expression on the right hand side is the Euclidean scalar product on $\mathbb{R}^N$). By Theorem 3.1 applied to the operator $\mathbf{C}^\tau$ on $\mathbb{R}^N$, the optimal vectors $\mathbf{a}_m$ are given by the first M eigenvectors of the symmetric matrix $\mathbf{C}^\tau$. In the general case, transform the basis functions into a set of mutually orthonormal functions $\tilde{f}_i$ via

$$\tilde{f}_i = \sum_{j=1}^{N} \left(\mathbf{C}^0\right)^{-1/2} (j,i) f_j. \tag{3.1.13}$$

The square root of the matrix $\mathbf{C}^0$ is guaranteed to exist because the basis functions were assumed to be linearly independent. For the transformed basis, we need to compute the eigenvectors $\tilde{\mathbf{a}}_m$ of

$$\left(\mathbf{C}^0\right)^{-1/2} \mathbf{C}^\tau \left(\mathbf{C}^0\right)^{-1/2} \tilde{\mathbf{a}}_m = \hat{\lambda}_m \tilde{\mathbf{a}}_m. \tag{3.1.14}$$

This is equivalent to the generalized eigenvalue problem Eq. (3.1.7), the relation between the eigenvectors is given by

$$\mathbf{a}_m = \left(\mathbf{C}^0\right)^{-1/2} \tilde{\mathbf{a}}_m. \tag{3.1.15}$$

□

The generalized eigenvalue problem Eq. (3.1.7) is the cornerstone of this thesis. It allows us to compute an approximation to the leading transfer operator eigenfunctions from any finite-dimensional subspace D. Approximations from different subspaces can be compared based on the Rayleigh trace, which is easily computable from the solution [49]:

Lemma 3.3. *Let $f_{\mathbf{a}_m}$, $m = 1, \ldots, M$ denote the solutions of the generalized eigenvalue problem Eq. (3.1.7). The functions $f_{\mathbf{a}_m}$ are mutually orthonormal in L^2_μ and the associated Rayleigh trace is given by*

$$\sum_{m=1}^{M} \langle \mathcal{T}_\tau f_{\mathbf{a}_m}, f_{\mathbf{a}_m} \rangle_\mu = \sum_{m=1}^{M} \hat{\lambda}_m. \tag{3.1.16}$$

Also, each individual eigenvalue $\hat{\lambda}_m$ underestimates the true eigenvalue λ_m,

$$\begin{aligned} \hat{\lambda}_m &= \langle \mathcal{T}_\tau f_{\mathbf{a}_m}, f_{\mathbf{a}_m} \rangle_\mu && (3.1.17) \\ &\leq \lambda_m. && (3.1.18) \end{aligned}$$

Proof. The orthonormality of solutions follows from the orthonormality of generalized eigenvectors with respect to the inner product on $\mathbb{R}^N$ induced by the symmetric positive-definite matrix $\mathbf{C}^0$:

$$\begin{aligned} \delta_{m,m'} &= \mathbf{a}_m^T \mathbf{C}^0 \mathbf{a}_{m'} && (3.1.19) \\ &= \langle f_{\mathbf{a}_m}, f_{\mathbf{a}_{m'}} \rangle_\mu. && (3.1.20) \end{aligned}$$

The second equality follows from Eq. (3.1.11). Likewise, the statement about the Rayleigh trace follows from Eq. (3.1.10) via

$$\begin{aligned} \langle \mathcal{T}_\tau f_{\mathbf{a}_m}, f_{\mathbf{a}_m} \rangle_\mu &= \mathbf{a}_m^T \mathbf{C}^\tau \mathbf{a}_m && (3.1.21) \\ &= \hat{\lambda}_m \mathbf{a}_m^T \mathbf{C}^0 \mathbf{a}_m && (3.1.22) \\ &= \hat{\lambda}_m. && (3.1.23) \end{aligned}$$

It remains to prove inequality (3.1.18). First, we note that repeating the calculation in Eqs. (3.1.21-3.1.23) using functions $f_{\mathbf{a}_m}$, $f_{\mathbf{a}_{m'}}$, it follows that

$$\langle \mathcal{T}_\tau f_{\mathbf{a}_m}, f_{\mathbf{a}_{m'}} \rangle_\mu = \hat{\lambda}_{m'} \delta_{m,m'}. \quad (3.1.24)$$

Now, let D_m denote the space spanned by the first m solutions of Eq. (3.1.7), i.e. $D_m = \text{span}\,\{f_{\mathbf{a}_r},\, r = 1, \ldots m\}$. For any normalized function $f = \sum_{r=1}^m c_r f_{\mathbf{a}_r} \in D_m$, we conclude from the normalization and from Eq. (3.1.24):

$$\begin{aligned} 1 &= \langle f, f \rangle_\mu && (3.1.25) \\ &= \sum_{r=1}^m c_r^2, && (3.1.26) \\ \langle \mathcal{T}_\tau f, f \rangle_\mu &= \sum_{r=1}^m \hat{\lambda}_r c_r^2. && (3.1.27) \end{aligned}$$

The last expression is bounded from below by $\hat{\lambda}_m$. Moreover, we can find a normalized function $g \in D_m$ such that g is orthogonal to the first $m-1$ solutions $f_{\mathbf{a}_1}, \ldots, f_{\mathbf{a}_{m-1}}$. Using the Rayleigh variational principle again, we conclude that

$$\begin{aligned} \hat{\lambda}_m &\leq \langle \mathcal{T}_\tau g, g \rangle_\mu && (3.1.28) \\ &\leq \lambda_m. && (3.1.29) \end{aligned}$$

□

3.2 Estimation from Data

The generalized eigenvalue problem Eq. (3.1.7) requires the computation of the matrices $\mathbf{C}^\tau$, $\mathbf{C}^0$, the elements of which contain overlap integrals between basis functions (and operators) on the state space S. The computation of these integrals cannot be performed by standard quadrature methods for realistic systems, because of the high-dimensionality of S, and because of the lack of a closed-form expression for the transfer operator $\mathcal{T}_\tau$ or its transition kernel p_τ. However, the matrix entries also correspond to spatial correlations that can be estimated from a sufficiently long realization of the process. Using the space $\Omega_{\Delta t}$ with measure $\mathbb{P}_{\Delta t}$ from Eq. (2.1.7), it can be shown that (see appendix C.2 and Ref. [50, appendix B]):

Theorem 3.4. *Let $\tau = L\Delta t$ be an integer multiple of the discrete time step. If the Markov process is initially distributed according to the unique invariant measure μ, then for $\mathbb{P}_{\Delta t}$-a.s. all trajectories $(X_{k\Delta t})_{k=0}^{\infty}$, we have:*

$$\mathbf{C}^\tau(i,j) = \lim_{K\to\infty} \frac{1}{K-L} \sum_{k=0}^{K-L-1} f_i(X_{k\Delta t}) f_j(X_{(k+L)\Delta t}), \tag{3.2.1}$$

$$\mathbf{C}^0(i,j) = \lim_{K\to\infty} \frac{1}{K-L} \sum_{k=0}^{K-L-1} f_i(X_{k\Delta t}) f_j(X_{k\Delta t}). \tag{3.2.2}$$

Theorem 3.4 is the real strength of the variational formulation from the previous chapter. The matrices required to solve the eigenvalue problem Eq. (3.1.7) can be approximated by computing matrices of instantaneous and time-lagged correlations between the basis functions from a long equilibrium simulation of the process.

3.3 Special Cases: Markov State Models and TICA

We would like to highlight two special cases of the approximation procedure from Prop. 3.2. The first is obtained by partitioning the state space S into N disjoint sets S_i, $i = 1, \ldots, N$, and choosing the space D as the span of the sets' indicator functions $f_i = \chi_{S_i}$. In this case, we find

$$\mathbf{C}^0(i,j) = \int_S \chi_{S_i}(x)\chi_{S_j}(x)\mu(\mathrm{dx}) \tag{3.3.1}$$

$$= \delta_{ij}\mathbb{P}(X_0 \in S_i), \tag{3.3.2}$$

$$\mathbf{C}^\tau(i,j) = \int_{S_i} p(x, S_j)\mu(\mathrm{dx}) \tag{3.3.3}$$

$$= \mathbb{P}(X_0 \in S_i, X_\tau \in S_j). \tag{3.3.4}$$

The matrix $\mathbf{C}^0$ is a diagonal matrix of the stationary probabilities of all sets S_i, while the matrix $\mathbf{C}^\tau$ contains the joint probabilities of observing the process in S_i first and in S_j after a time step τ. Multiplying the generalized eigenvalue problem Eq. (3.1.7) by the inverse of $\mathbf{C}^0$ from the left, and recalling Eq. (2.4.5), we end up with a standard eigenvalue problem of the form

$$\mathbf{T}^\tau \mathbf{a}_m = \hat{\lambda}_m \mathbf{a}_m, \tag{3.3.5}$$
$$\mathbf{T}^\tau(i,j) = p^\tau(S_i, S_j). \tag{3.3.6}$$

The matrix $\mathbf{T}^\tau$ contains conditional jump probabilities between the sets S_i and is therefore called a *transition matrix*, the resulting model is called a *Markov state model (MSM)* [9, 10, 11]. MSMs have many appealing properties and have been used very successfully in recent years.

The second special case is obtained as follows: assume that $x_1, \ldots x_N$ are the coordinates of the state space S or of some subspace of S. Then, choose the basis functions as the mean-free coordinates, i.e.

$$f_i = x_i - \langle x_i \rangle_\mu \tag{3.3.7}$$
$$= x_i - \int_S x_i \mu(\mathrm{dx}). \tag{3.3.8}$$

In this case, the generalized eigenvalue problem Eq. (3.1.7) is equivalent to *time-lagged independent component analysis (TICA)*, which has been known for a long time in statistics [28] under various names, and has been introduced in molecular dynamics by Refs. [29, 30].

3.4 Implied Timescales

In many cases, the semigroup property from Prop. 2.5 (ii) implies that dynamical eigenvalues $\lambda_m(\tau)$, $m > 1$ decay exponentially as a function of the lag time. More precisely, the *infinitesimal generator* of the semigroup of transfer operators $\mathcal{T}_\tau$ is defined by

$$\mathcal{A}f = \lim_{t \to 0} \frac{\mathcal{T}_t f - f}{t}. \tag{3.4.1}$$

The operator $\mathcal{A}$ is defined for all functions f s.t. the limit in Eq. (3.4.1) exists. Typically, these functions only form a subspace of the transfer operator's domain, and the resulting generator is unbounded. For a reversible process, the generator is self-adjoint and possesses real eigenvalues. Under the additional assumption of *strong continuity*, which is fulfilled for many types of stochastic differential equations [51], the following connection between the spectra of $\mathcal{A}$ and $\mathcal{T}_\tau$ can be made [51, Thm. 2.2.4]:

Theorem 3.5. *Let the semigroup of transfer operators $\mathcal{T}_\tau$ be strongly continuous, that is*

$$\lim_{t \to 0} \mathcal{T}_t f - f = 0 \tag{3.4.2}$$

holds for all f in the domain of $\mathcal{T}_\tau$. Let $\mathcal{A}$ be the infinitesimal generator. Then, the spectra of $\mathcal{A}$ and $\mathcal{T}_\tau$ are connected via

$$e^{\tau\sigma_p(\mathcal{A})} \quad \subset \quad \sigma_p(\mathcal{T}_\tau) \subset e^{\tau\sigma_p(\mathcal{A})} \cup \{0\}\,, \tag{3.4.3}$$

where σ_p denotes the point spectrum of an operator. The corresponding eigenfunctions of the generator and transfer operator are identical.

It follows that eigenvalues κ_m of the generator are non-positive, and the largest eigenvalue $\kappa_1 = 0$ is non-degenerate. The corresponding eigenfunction is $\psi_1 \equiv 1$ and corresponds to the stationary process. For every negative eigenvalue κ_m, $m > 1$ of the generator, the corresponding eigenvalue $\lambda_m(\tau)$ decays exponentially with rate κ_m:

$$\lambda_m(\tau) \quad = \quad e^{\kappa_m \tau}. \tag{3.4.4}$$

In what follows, we will refer to the eigenvalues of the generator as rates. Metastability of the transfer operator is then reflected in the presence of rates close to zero.

Based on the exponential decay of dynamical eigenvalues, so-called *implied timescales* have become an important concept. For $m > 1$, the m-th implied timescale is defined as the negative inverse rate:

$$t_m \quad = \quad -\frac{1}{\kappa_m} \tag{3.4.5}$$

$$\quad = \quad -\frac{\tau}{\log(\lambda_m(\tau))}. \tag{3.4.6}$$

The implied timescales are characteristic quantities of the system. If $\lambda_m(\tau)$ corresponds to a metastable transition, then for lag times much larger than t_m, $\lambda_m(\tau)$ has practically decayed to zero, thus the relaxation process between the corresponding metastable sets has equilibrated. It follows that implied timescales can be connected to experimentally observed relaxation timescales. Moreover, if the eigenvalues $\lambda_m(\tau)$ are known or can be estimated without error, the right hand side of Eq. (3.4.6) does not depend on the lag time τ. If, however, they are estimated from an approximation $\hat{\lambda}_m(\tau)$ using Eq. (3.4.6), it follows from the variational principle Thm. 3.1 that the timescales are underestimated. It has been shown in Ref. [52] that the estimation error $|\hat{\lambda}_m(\tau) - \lambda_m(\tau)|$ decays to zero with increasing lag time. Therefore, it can be expected that timescale estimates based on Eq. (3.4.6) become constant for large enough τ. This convergence is at least a necessary condition for a good approximation of the eigenvalues $\lambda_m(\tau)$. The following *implied timescale test* has become a standard validation procedure in Markov model construction [9]: For a series of lag times $\tau_1 < \ldots < \tau_l$, a model is estimated at each lag time τ_i. The leading implied timescales are estimated using Eq. (3.4.6) and these estimates are plotted as a function of the lag time. By visual inspection of the plot, a lag time $\tau*$ is selected such that all relevant timescales are approximately constant for $\tau \geq \tau*$, and the model at $\tau*$ is analysed further. If such a model cannot be determined, it is recommended to modify the model inputs, like the discretization or the data. We will also use the implied timescale test as a model validation tool in the ensuing numerical examples.

3.5 Alternative Derivation

Only recently, it has been found [53] that the algorithm of the VAC is identical to *extended dynamic mode decomposition* (EDMD) [54] for reversible dynamics, although the theoretical derivation of EDMD is different. In the limit of infinite sampling, EDMD converges to a Galerkin approximation of the Koopman operator, even for non-reversible dynamics. As a consequence, the VAC algorithm can also be used to obtain approximate eigenfunctions of the Koopman operator for non-reversible dynamics. It should be noted that the interpretation of these eigenfunctions may be different in this case.

Chapter 4

Model Applications

In this chapter, we apply the VAC to four model systems: Two one-dimensional toy systems, and two standard examples of small molecular systems. We confirm that these systems can be analyzed by smaller basis sets compared to standard MSM protocols if physically motivated basis functions are used. Let us introduce a new notation which we will use frequently in the remainder of the text. If $\mathbf{a}_m$ are the solutions of the generalized eigenvalue problem Eq. (3.1.7), the corresponding functions $f_{\mathbf{a}_m}$ (see Eq. (3.1.6)) serve as approximations to the true eigenfunctions ψ_m. Therefore, we also denote them by $\hat{\psi}_m$ now:

$$\hat{\psi}_m = \sum_{i=1}^{N} \mathbf{a}_m(i) f_i. \tag{4.0.1}$$

The text is adapted from Ref. [55].

4.1 Systems and Models

4.1.1 One-dimensional Diffusion Models

Simulations. We first consider two examples of one-dimensional diffusion processes X_t governed by Brownian dynamics. The process is then described by the stochastic differential equation

$$dX_t = -\nabla v(X_t)dt + \sqrt{2D}\, dB_t \tag{4.1.1}$$

where v is the reduced potential energy (measured in units of $k_B T$, where k_B is the Boltzmann constant and T is the temperature), D is the diffusion constant, and dB_t denotes the differential of Brownian motion. For simplicity, we set all of the above constants equal to one. The potential function is given by the harmonic potential

$$v(x) = 0.5x^2, \quad x \in \mathbb{R} \tag{4.1.2}$$

in the first case, and by the periodic double-well potential

$$v(x) = 1 + \cos(2x), \quad x \in [-\pi, \pi), \tag{4.1.3}$$

in the second case. In order to apply our method, we first produced finite simulation trajectories for both potentials. To this end, we picked an (also artificial) time step $\Delta t = 10^{-3}$, and then used the Euler-Maruyama method, where position x_{k+1} is computed from position x_k as

$$x_{k+1} = x_k - \Delta t \nabla v(x_k) + \sqrt{2D\Delta t} y_t \tag{4.1.4}$$

$$y_t \sim \mathcal{N}(0,1). \tag{4.1.5}$$

In this way, we produced simulations of $5 \cdot 10^6$ time steps for the harmonic potential and 10^7 time steps for the double-well potential.

Gaussian model. We apply our method with Gaussian basis functions to both problems. To this end, $N = 2, 3, \ldots, 10$ centers are chosen at uniform distance between $x = -4$ and $x = 4$ for the harmonic potential and between $x = -\pi$ and $x = \pi$ for the double-well potential. In the latter case, the basis functions are modified to be periodic on $[-\pi, \pi)$. Subsequently, an "optimal" width of the Gaussians is picked by simply trying out several choices for the standard deviations between 0.4 and 1.0 and using the one which yields the highest second eigenvalue. From this choice, the matrices $\mathbf{C}^\tau$ and $\mathbf{C}^0$ are estimated and the eigenvalues, -functions and implied timescales are computed.

Markov models. As a reference for our methods, we also compute Markov state models for both processes. To this end, the simulation data is clustered into $N = 2, 3, \ldots, 10$ disjoint clusters using the kmeans algorithm. Subsequently, the EMMA software package [56] is used to estimate the MSM transition matrices and to compute eigenvalues and timescales.

4.1.2 Alanine Dipeptide

MD simulations. We performed 20 simulations of 200 ns of all-atom explicit solvent molecular dynamics of alanine dipeptide using the AMBER ff-99SB-ILDN force field [57]. The detailed simulation setup is found in appendix A.1.

Gaussian model. Similar to the previous example, we use periodic Gaussian functions which only depend on one of the two significant dihedral angles of the system (see sec. 4.2.2) to apply our method. For both dihedrals, we separately perform a pre-selection of the Gaussian trial functions. To this end, we first project the data onto the dihedral, then we solve the projected generalized eigenvalue problem Eq. (3.1.7) for all possible choices of centers and widths, and then pick the ones yielding the highest eigenvalues. In every step of the optimization, we select three out of four equidistributed centers between $-\pi$ and π, and one of eleven standard deviations between 0.04π and 0.4π. In this way, we obtain three Gaussian trial functions per coordinate, resulting in a full basis set of six functions. Having determined the parameters for both angles, we use the resulting trial functions to apply our method as before. A bootstrapping procedure is used to estimate the statistical uncertainty of the implied timescales.

Note that the variations of basis functions described here to find a “good” basis set could be conducted once for each amino acid (or short sequences of amino acids) for a given force field, and then be reused.

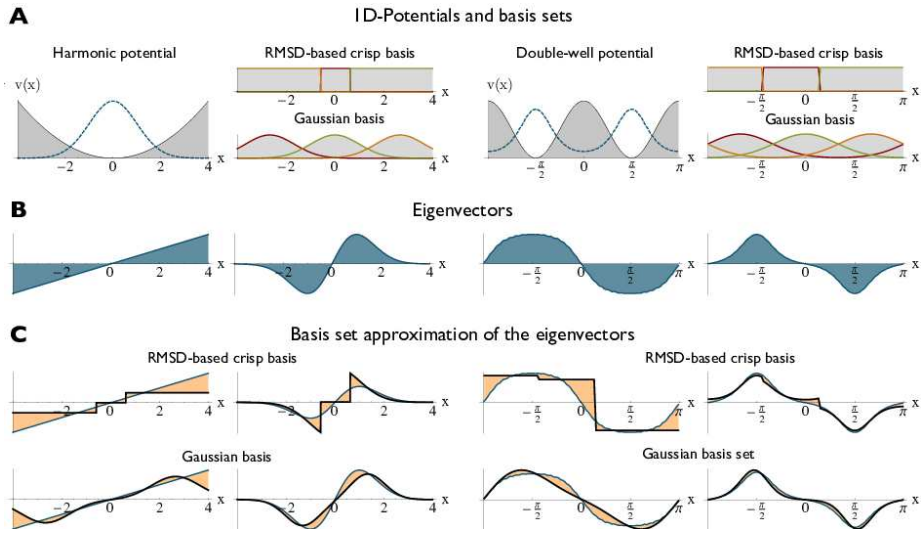

Figure 4.1: Illustration of the method with two one-dimensional potentials, the harmonic potential in the left half and a periodic double-well potential in the right half of the figure. Panel A shows the potential v together with its invariant distribution π (shaded) next to two possible choices of basis functions: A three-element crisp basis and a set of three Gaussian functions. Panel B shows the exact right and left second eigenfunctions, ψ_2 and φ_2. In Panel C, the approximation results for these second eigenfunctions obtained from the basis sets shown above are displayed. This figure has been re-used with permission from Nüske, Keller *et al.*, J. Chem. Theory Comput. 10(4), 1739-1752 (2014) [Fig. 2]. Copyright 2014 American Chemical Society.

Markov models. This time, we cluster the data into $N = 5, 6, 10, 15, 20, 30, 50$ clusters, again using the kmeans algorithm. From these clustercenters, we build Markov models and estimate the eigenvalues and eigenvectors using the EMMA software.

4.1.3 Deca Alanine.

MD simulations. We performed six 500 ns all-atom explicit solvent molecular dynamics simulations of deca alanine using the Amber03 force field, see appendix A.3 for the detailed simulation setup.

Gaussian model. As before, we use Gaussian basis functions which depend on the backbone dihedral angles of the peptide, which means that we now have a total of 18 internal coordinates. A pre-selection of the trial functions is performed for every coordinate independently, similar to the alanine dipeptide example. In order to keep the number of basis functions acceptably small, we select two trial functions per coordinate. As before, their centers are chosen from four equidistributed centers along the coordinate, and their standard deviations are chosen from eleven different values between 0.04π and 0.4π. We also build a second Gaussian model using five functions per coordinate, with

equidistributed centers and standard deviations optimized from the same values as in the first model. Having determined the trial functions, we estimate the matrices $\mathbf{C}^\tau$ and $\mathbf{C}^0$ and compute the eigenvalues and eigenvectors, and again use bootstrapping to estimate uncertainties.

Markov models. We construct two different Markov models from the dihedral angle data. The first is built using kmeans clustering with 1000 cluster-centers on the full data set, whereas for the second, we divide the $\phi - \psi$ plane of every dihedral pair along the chain into three regions corresponding to the α-helix, β-sheet and left-handed α-helix conformation, see section 4.2.2. Thus, we have three discretization boxes for all dihedral pairs, which yields a total of 8^3 discrete states to which the trajectory points are assigned.

4.2 Results

We now turn to the results obtained for the four systems presented in the previous section.

4.2.1 One-dimensional Potentials

The two one-dimensional systems are toy examples where all important properties are either analytically known or can be computed arbitrarily well from approximations. For the harmonic potential, the stationary distribution is just a Gaussian function

$$\pi(x) \quad = \quad \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2}). \tag{4.2.1}$$

The exact eigenvalues $\lambda_m(\tau)$, as a function of τ, are given by

$$\lambda_m(\tau) \quad = \quad \exp(-(m-1)\tau), \tag{4.2.2}$$

and the associated right eigenfunction ψ_m is given by the $(m-1)$-th normalized Hermite polynomial

$$\psi_m(x) = H_{m-1}(x) \quad \sim \quad (-1)^{m-1} \exp(\frac{x^2}{2}) \frac{d^{m-1}}{dx^{m-1}} \exp(-\frac{x^2}{2}). \tag{4.2.3}$$

The left halves of Figs. 4.1A and 4.1B show the harmonic potential and its stationary distribution, as well as the second right and left eigenfunction. The sign change of φ_2 indicates the oscillation around the potential minimum, which is the slowest equilibration process. Note, however, that there is no energy barrier in the system, i.e. this process is not metastable. On the right hand sides of Figs. 4.1A and 4.1B, we see the same for the periodic double-well potential. The invariant density is equal to the Boltzmann distribution, where the normalization constant was computed numerically. The second eigenfunction was computed by a very fine finite-element approximation of the corresponding Fokker-Planck equation, using 1000 linear elements. The slowest transition in the system is the crossing of the barrier between the left and right minimum. This is reflected in the characteristic sign change of the second eigenfunction.

Figures 4.1A and 4.1B also show two choices of basis sets which can be used to approximate these eigenfunctions: A three element Gaussian basis set and a three state crisp set. The resulting estimates of the right and left eigenfunctions are displayed in Fig. 4.1C. Already with these small basis sets, a good approximation is achieved.

Let us analyze the approximation quality of both methods in more detail. To this end, we first compute the L^2_μ-approximation error between the estimated second eigenfunction $\hat{\psi}_2$ and the exact solution ψ_2, i.e. the integral

$$\delta \quad = \quad \int_S (\psi_2(x) - \hat{\psi}_2(x))^2 \pi(x) \mathrm{dx}. \tag{4.2.4}$$

We expect this error to decay if the basis sets grow. Indeed, this is the case, as can be seen in the upper graphics of Figs. 4.2A and 4.2B, but the error produced by the Gaussian basis sets decays faster. Even for the ten state MSM, we still have a significant approximation error. Another important indicator is the implied timescale t_m. It was defined in sec. 3.4 and corresponds to the equilibration time of the associated slow transition. The exact value of t_m is independent of the lag time τ. But if we estimate the timescale from the approximate eigenvalues, the estimate will be too small due to the variational principle. However, with increasing lag time, the error is expected to decay, as the approximation error also decays with the lag time. The faster this decay occurs, the better the approximation will be. In the lower graphics of Figs. 4.2A and 4.2B, we see the lag time dependence of the second timescale t_2 for growing crisp and Gaussian basis sets. We observe that it takes only four to five Gaussian basis functions to achieve much faster convergence compared even to a ten state Markov model. For 7 or more Gaussian basis functions, we achieve precise estimates even for very short lag times, which cannot be achieved with Markov models with a reasonable number of states.

4.2.2 Alanine Dipeptide

Alanine dipeptide (Ac-Ala-NHMe, i.e. an alanine linked at either end to a protection group) is designed to mimic the dynamics of the amino acid alanine in a peptide chain. Unlike the previous examples, the eigenfunctions and eigenvalues of alanine dipeptide cannot be calculated directly from its potential energy function, but have to be estimated from simulations of its conformational dynamics. However, alanine dipeptide is a thoroughly studied system, many important properties are well-known, though their estimated values depend on the precise potential energy function (force field) used in the simulations. Most importantly, it is known that the dynamical behaviour can be essentially understood in terms of the two backbone dihedral angles ϕ and ψ: Fig. 4.3A shows the free energy landscape obtained from population inversion of the simulation, where white regions correspond to non-populated states. We find the three characteristic minima in the upper left, central left, and central right part of the plane, which correspond to the β-sheet, α-helix and left-handed α-helix conformation of the amino acid. The two slowest transitions occur between the left half and the left handed α-helix, and from β-sheet to α-helix within the main well on the left, respectively.

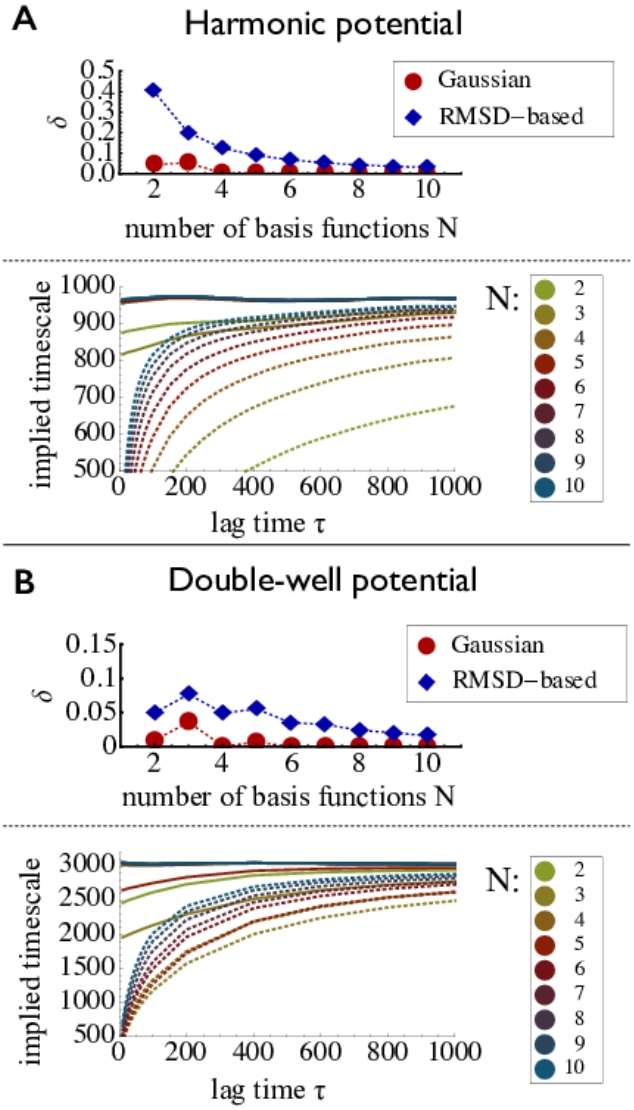

Figure 4.2: Analysis of the discretization error for both 1D-potentials. In the upper figure of both panels, we show the L^2_μ-approximation error of the second eigenfunction from both crisp basis functions and Gaussian basis functions, dependent on the size of the basis set. The lower figures show the convergence of the second implied timescale t_2 dependent on the lag time τ. Dotted lines represent the crips basis sets and solid lines the Gaussian basis sets. The colours indicate the size of the basis. This figure has been re-used with permission from Nüske, Keller *et al.*, J. Chem. Theory Comput. 10(4), 1739-1752 (2014) [Fig. 3]. Copyright 2014 American Chemical Society.

Figure 4.3B shows the weighted second and third eigenfunctions. They are obtained by applying our method with a total of six basis functions (3 for each dihedral), and from an MSM constructed from 30 clustercenters. The resulting

estimates of ψ_2 and ψ_3 are then weighted with the population estimated from the trajectory, in order to emphasize the regions of phase space which are related to the structural transitions. Almost identical results are achieved, and the sign pattern of both approximations clearly indicates the aforementioned processes.

Lastly, in Fig. 4.3C, we again investigate the convergence of the slowest implied timescales. Different MSMs with a growing number of crisp basis functions (clustercenters) were used and compared to the six basis function Gaussian model. The colors indicate the number of basis functions used, the thinner lines correspond to the Markov models, whereas the thick solid line is obtained from the Gaussian model. In agreement with the previous results, we find that thirty or more crisp basis functions are needed to reproduce a similar approximation quality like a six-Gaussian basis set.

4.2.3 Deca Alanine

As a third and last example, we study deca alanine, a small peptide which is about five times the size of alanine dipeptide. A sketch of the peptide is displayed in Fig. 4.4A.

The slow structural processes of deca alanine are less obvious compared to alanine dipeptide. The Amber03 force field used in our simulation produces a relatively fast transition between the elongated and the helical state of the system, with an associated timescale of 5 to 10 nanoseconds. As we can see in Fig. 4.4B, we are able to recover this slowest timescale with our method, t_2 converges to roughly 6.5 ns for both models. Comparing this to the two Markov models constructed from the same simulation data, we see that both yield slightly higher timescales: The kmeans based MSM returns a value of about 8 ns and the finely discretized one ends up with 8.5 ns. Note that the underestimate of the present Gaussian basis set is systematic, and likely due to the fact that all basis functions were constructed as a function of single dihedral angles only, thereby neglecting the coupling between multiple dihedrals.

Despite this approximation, we are able to determine the correct structural transition. In order to analyse this, we evaluate the second eigenfunction $\hat{\psi}_2$, obtained from the smaller model, for all trajectory points, and plot a histogram of these values as displayed in Fig. 4.4C. We then select all frames which are within close distance of the peaks of that histogram, and produce overlays of these frames as shown underneath. Clearly, large negative values of the second eigenfunction indicate that the peptide is elongated, whereas large positive values indicate that the helical conformation is attained. This is in accord with a similar analysis of the second right Markov model eigenvector: In Fig. 4.4D, we show overlays of structures taken from states with the most negative and most positive values of the second eigenvector, and we find that the same transition is indicated, although the most negative values correspond to a slightly more bent arrangement of the system.

In summary, it is possible to use a comparatively small basis of 36 Gaussian functions to achieve results about the slowest structural transition which are comparable to those of MSMs constructed from about 1000 and 6500 discrete states, respectively. However, the differences in the timescales point to a weakness of the method: The fact that increasing the number of basis functions does not alter the computed timescale indicates that coordinate correlation cannot

be appropriately captured using sums of one-coordinate basis functions. In order to use the method for larger systems, we will have to study ways to overcome this problem.

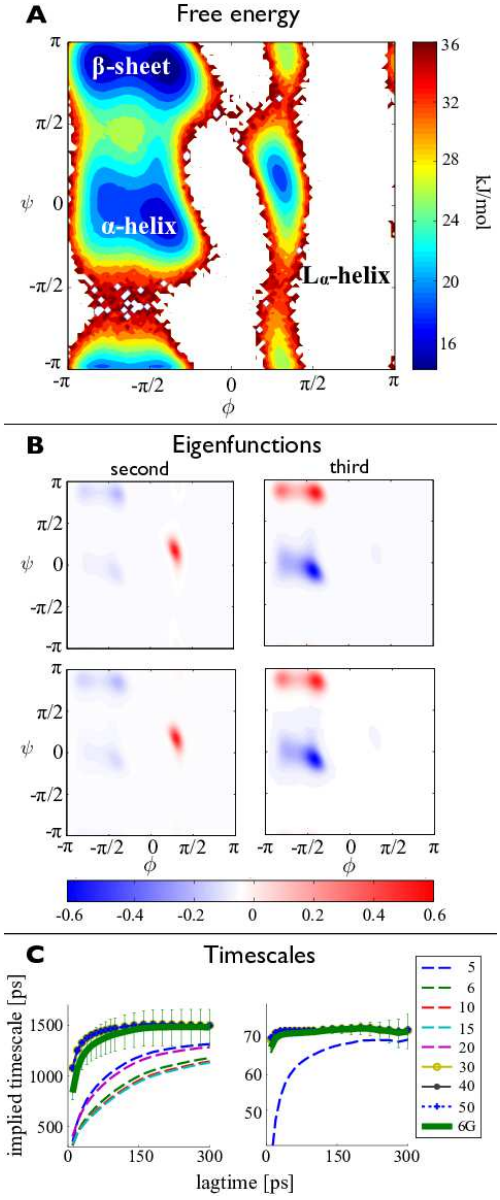

Figure 4.3: Illustration of the method for alanine dipeptide data. A) Free energy landscape from histogramming $\phi - \psi$ dihedral data. B1 and B2) Contour plots of approximate models for the eigenfunctions ψ_2 and ψ_3 from a Gaussian basis set with six functions, weighted by the estimated stationary distribution from A). C1 and C2) The same if ψ_2 and ψ_3 are approximated by a Markov state model with 30 clustercenters. D1 and D2) Convergence of implied timescales t_m (in picoseconds) corresponding to the second and third eigenfunction, as obtained from Markov models using $N = 5, 6, 10, 15, 20, 30, 50$ clustercenters (thin lines), compared to the timescales obtained from the Gaussian model with a total of six basis functions (thick green line). Thin vertical bars indicate the error estimated by a bootstrapping procedure. This figure has been re-used with permission from Nüske, Keller *et al.*, J. Chem. Theory Comput. 10(4), 1739-1752 (2014) [Fig. 4]. Copyright 2014 American Chemical Society.

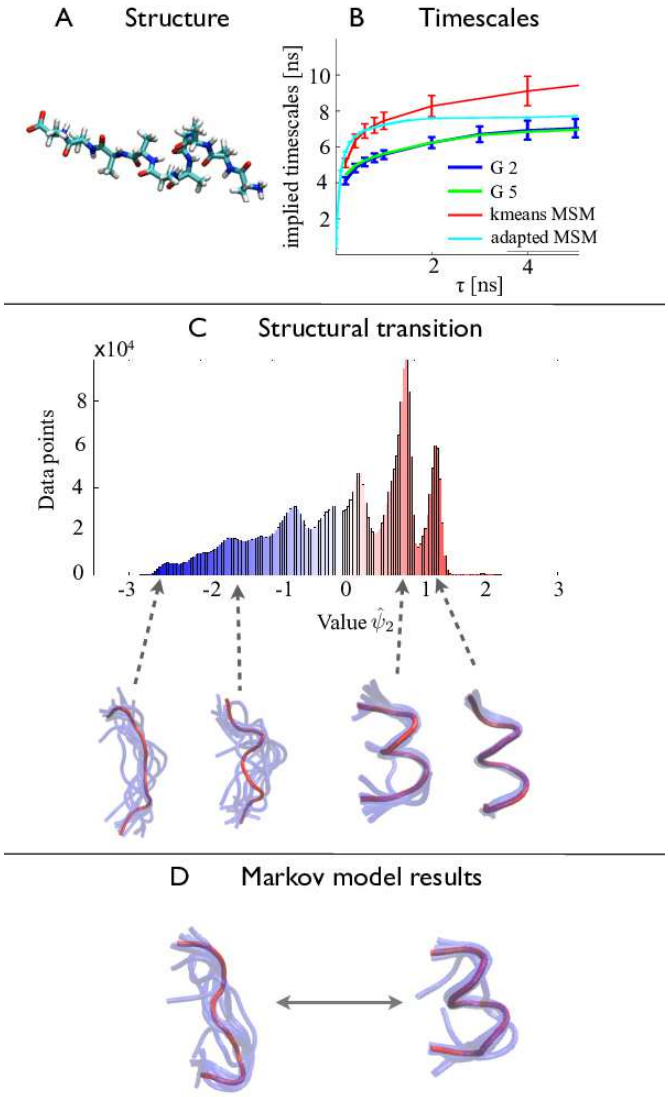

Figure 4.4: Illustration of the method using dihedral angle coordinates of the deca alanine molecule. A) Graphical representation of the system. B) Convergence of the estimated second implied timescale (in nanoseconds) depending on the lag time. We show the results of both Gaussian models and of both the kmeans based MSM and the adapted MSM. Thin vertical bars indicate the error estimated by a bootstrapping procedure. C) Assignment of representative structures for the second slowest process: The histogram shows how the values of the second estimated eigenfunction $\hat{\psi}_2$ of the smaller model are distributed over all simulation trajectories. Underneath, we show an overlay of structures taken at random from the vicinity of the peaks at -2.7, -1.6, 0.7 and 1.3. D) Overlays of structures corresponding to the most negative (left) and most positive (right) values of the second Markov model eigenvector, taken from the kmeans MSM. This figure has been re-used with permission from Nüske, Keller *et al.*, J. Chem. Theory Comput. 10(4), 1739-1752 (2014) [Fig. 5]. Copyright 2014 American Chemical Society.

Chapter 5

Tensor Approach

Motivated by the last example in the preceding chapter, we now study the application of the VAC in conjunction with a basis set comprised of tensor products of one-coordinate basis functions. The text is adapted from Ref. [58].

5.1 Tensor Product Bases

In order to correctly model the dynamics of large systems where coordinates are coupled, we should use basis functions that are products of one-coordinate basis functions. Let us assume there are d input coordinates labeled

$$x_1, x_2, \ldots, x_d, \tag{5.1.1}$$

and for each coordinate x_p we have n one-coordinate basis functions $f_{i_p}^p(x_p)$ ($p = 1, \ldots, d$ and $i_p = 1, \ldots n$), only dependent on x_p. All of the following remains valid if n varies with p, but for simplicity of notation we assume a constant n here. For practical reasons, we assume that the first one-coordinate basis function is the constant, $f_1^p(x_p) \equiv 1$, although this is not needed for most of the theory. Then we try to approximate each eigenfunction ψ_m, $m = 1, 2, \ldots$ by a function $\hat{\psi}_m$ that is a linear combination of all possible products of the $f_{i_p}^p$:

$$\hat{\psi}_m(x_1, \ldots, x_d) \quad = \quad \sum_{i_1,\ldots,i_d} \mathbf{A}_m(i_1, \ldots, i_d)\, f_{i_1}^1(x_1) \ldots f_{i_d}^d(x_d). \tag{5.1.2}$$

The basis functions for the variational approach are the products

$$f_{i_1,\ldots,i_d}(x_1, \ldots, x_d) \quad = \quad f_{i_1}^1(x_1) \ldots f_{i_d}^d(x_d), \tag{5.1.3}$$

and $\mathbf{A}_m$ is a d-dimensional array (tensor) containing the expansion coefficients of all these products. The tensor $\mathbf{A}_m$ corresponds to the vector $\mathbf{a}_m$ from Eq. (3.1.7), but due to the product structure of the basis Eq. (5.1.3), we can treat it as a d-dimensional object here. As we can immediately see, the number of basis functions used in Eq. (5.1.2) is n^d. This number becomes impossible to

cope with even for small n and moderate d, not to mention the evaluation of the correlation matrices in Eqs. (3.2.1-3.2.2) using long trajectories. However, our experience and the high degree of redundancy contained in intramolecular coordinates suggest that a small selection of these product functions should be sufficient to produce essentially the same results. Let us illustrate this by another example, a capped dimer of the two amino acids valine and alanine (Ac-Val-Ala-NHMe). Here, we have two pairs of dihedral angles, the dimension thus becomes $d = 4$. Since the coordinates are periodic angles, we used the real Fourier basis of sine and cosine functions. Setting $n = 5$, the full product basis comprises $5^4 = 625$ functions. In Figure 5.1A, we check the accuracy of the model by comparing the two slowest implied timescales t_2, t_3 to those obtained from a reference Markov model. This model is obtained by discretizing the dihedral plane of every residue into three states which were chosen according to known dynamics of the monomers, resulting in a total of $3^2 = 9$ states, see Ref. [59]. Both models perform comparably well. Clearly the Markov model is much more efficient, but its construction requires a priori knowledge of the peptide dynamics that is not easily transferred to larger systems. Figure 5.1B) shows the cumulative sum of the squared coefficients of the estimated second eigenfunction $\hat{\psi}_2$ from the product basis. The coefficients were computed after transforming the product basis into an orthonormal basis with respect to the π-weighted inner product Eq. (2.3.3). We observe that only a small part of the 625 basis functions contribute with a high coefficient compared to all others. We conclude that it should be possible to find a much smaller subspace of the full product space and end up with essentially the same result. The efficient search for this subspace is the topic of the next section.

5.2 Tensor Product Approximations

5.2.1 Tensor-Train-Format

The problem of finding a computationally feasible approximation to a high-dimensional representation like Eq. (5.1.2) occurs across many fields, and significant progress has been made in recent years. Out of all the different approaches that have been suggested, we choose to present and use the tensor-train-format (TT-format), which has been introduced in [32, 33].
A function in TT-format still possesses a high-dimensional representation like Eq. (5.1.2), but the coefficient array $\mathbf{A}_m$ has a special structure as in Eq. (5.2.2) below [33]:

$$\hat{\psi}_m = \sum_{i_1,\ldots,i_d} \mathbf{A}_m(i_1,\ldots,i_d)\, f_{i_1}^1(x_1)\ldots f_{i_d}^d(x_d) \tag{5.2.1}$$

$$= \sum_{i_1,\ldots,i_d} \left[\sum_{k_1=1}^{r_1} \ldots \sum_{k_{d-1}=1}^{r_{d-1}} \mathbf{U}_1(i_1,k_1)\mathbf{U}_2(k_1,i_2,k_2)\ldots\mathbf{U}_d(k_{d-1},i_d) \right] \cdot f_{i_1}^1(x_1)\ldots f_{i_d}^d(x_d). \tag{5.2.2}$$

Here, $\mathbf{U}_1 \in \mathbb{R}^{n\times r_1}$, $\mathbf{U}_d \in \mathbb{R}^{r_{d-1}\times n}$ are matrices and $\mathbf{U}_p \in \mathbb{R}^{r_{p-1}\times n\times r_p}$, $p =$

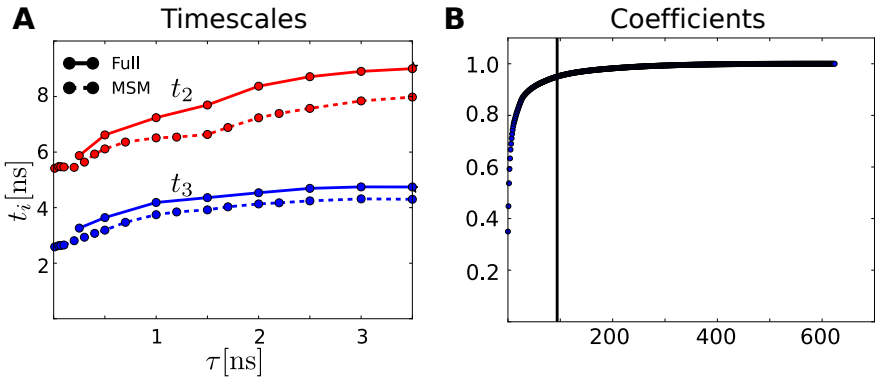

Figure 5.1: Illustration of low-dimensional subspaces carrying the relevant information for the dimer Ac-Val-Ala-NHMe: Panel A shows the two slowest implied timescales t_2, t_3, in red and blue, estimated by two different models: A reference MSM (dashed line, 9 states) and the full product expansion Eq. (5.1.2) (solid line, $n = 5$, 625 basis functions). Both models perform comparably well. In panel B, we present a cumulative plot of the squared expansion coefficients of the second eigenfunction $\hat{\psi}_2$, as estimated by the full product approach, expressed in an orthonormal basis w.r.t. the weighted inner product Eq. (2.3.3). It takes about 90 basis functions to reproduce 95 percent of the norm, as indicated by the black vertical line. This figure has been re-used with permission from Nüske *et al.*, J. Chem. Phys. 144, 054105 (2016) [Fig. 2]. Copyright 2016 AIP Publishing LLC.

$2, \ldots, d-1$, are three-dimensional arrays. Consequently, for every choice of $i_1, \ldots, i_d$, the arrays $\mathbf{U}_1$ and $\mathbf{U}_d$ turn into vectors $\mathbf{U}_1(i_1)$, $\mathbf{U}_d(i_d)$, whereas all other arrays $\mathbf{U}_2, \ldots, \mathbf{U}_{d-1}$ become matrices $\mathbf{U}_2(i_2), \ldots, \mathbf{U}_{d-1}(i_{d-1})$, and the coefficient $\mathbf{A}_m(i_1, \ldots, i_d)$ can be computed by a repeated matrix-vector multiplication:

$$\mathbf{A}_m(i_1, \ldots, i_d) \quad = \quad \mathbf{U}_1(i_1)\mathbf{U}_2(i_2)\ldots\mathbf{U}_d(i_d). \tag{5.2.3}$$

Thus, only the arrays $\mathbf{U}_1, \ldots, \mathbf{U}_d$ need to be stored, and the number of parameters in these arrays is linear in the dimension d, see again Ref. [33].
The intuition behind this representation is that only limited information is passed on from one variable to the next in the sequence $x_1, \ldots, x_d$. To see this, consider the case $d = 4$, and re-order Eq. (5.2.2) as follows:

$$\begin{aligned} \hat{\psi}_m \quad &= \quad \sum_{k_1,i_2,k_2} \mathbf{U}_2(k_1^*, i_2, k_2^*)\, f_{i_2}^2(x_2) \cdot \qquad (5.2.4) \\ &\qquad \left[\sum_{i_1} \mathbf{U}_1(i_1, k_1^*) f_{i_1}^1(x_1)\right] \cdot \\ &\qquad \left[\sum_{i_3,i_4} \sum_{k_3} \mathbf{U}_3(k_2^*, i_3, k_3)\mathbf{U}_4(k_3, i_4)\, f_{i_3}^3(x_3) \cdot f_{i_4}^4(x_4)\right] \\ &= \quad \sum_{k_1,i_2,k_2} \mathbf{U}_2(k_1^*, i_2, k_2^*)\, f_{i_2}^2(x_2) \cdot g_{k_1^*}^2(x_1) \cdot h_{k_2^*}^2(x_3, x_4). \qquad (5.2.5) \end{aligned}$$

The expressions shown in brackets in Eq. (5.2.4) contain exactly one free index k_1 and k_2, respectively, indicated by the stars (these are not complex conjugates, they only serve to highlight the corresponding indices). Thus, it makes sense to define functions $g^2_{k_1}$, $h^2_{k_2}$ by these expressions, which leads us to the representation in Eq. (5.2.5). The meaning of Eq. (5.2.5) is that the function $\hat{\psi}_m$ is represented by a linear combination of basis functions which can be separated into three parts: each basis function is a product of a function $f^2_{i_2}$ depending on the variable x_2, a function $g^2_{k_1}$ which depends on all variables up to x_2, and another function $h^2_{k_2}$ which depends on all unknowns following x_2. Thus, the information about all coordinates up to x_2 is encoded into a limited number of functions, and so is the information about all coordinates following x_2. The representation in Eq. (5.2.4) corresponds to panel B in Fig. 5.2. However, this is not the only way to re-order Eq. (5.2.2), as there are d equivalent ways to do so. All of these different re-orderings for the case $d = 4$ are displayed in the remaining parts of Fig. 5.2. In the general case, the re-ordering centered around coordinate x_p is given by

$$\begin{aligned}
\hat{\psi}_m \;=\; & \sum_{k_{p-1},i_p,k_p} \mathbf{U}_p(k^*_{p-1},i_p,k^*_p)\, f^p_{i_p}(x_p)\cdot \qquad (5.2.6)\\
& \Bigg[\sum_{i_1,\ldots,i_{p-1}} \sum_{k_1,\ldots,k_{p-2}} \mathbf{U}_1(i_1,k_1)\ldots\mathbf{U}_{p-1}(k_{p-2},i_{p-1},k^*_{p-1})\\
& f^1_{i_1}(x_1)\ldots f^{p-1}_{i_{p-1}}(x_{p-1})\Bigg]\cdot\\
& \Bigg[\sum_{i_{p+1},\ldots,i_d} \sum_{k_{p+1},\ldots,k_{d-1}} \mathbf{U}_{p+1}(k^*_p,i_{p+1},k_{p+1})\ldots\mathbf{U}_d(k_{d-1},i_d)\\
& f^{p+1}_{i_{p+1}}(x_{p+1})\ldots f^d_{i_d}(x_d)\Bigg]\\
=\; & \sum_{k_{p-1},i_p,k_p} \mathbf{U}_p(k^*_{p-1},i_p,k^*_p) \qquad (5.2.7)\\
& f^p_{i_p}(x_p)\cdot g^p_{k^*_{p-1}}(x_1,\ldots,x_{p-1})\cdot h^p_{k^*_p}(x_{p+1},\ldots,x_d).
\end{aligned}$$

The underlying principle is the same: The information about the variables $x_1,\ldots,x_{p-1}$ is encoded into r_{p-1} functions $g^p_{k_{p-1}}$, which we call the left *interfaces* at position p. Also, the information about the variables $x_{p+1},\ldots,x_d$ is contained in r_p functions $h^p_{k_p}$, called right interfaces at p. The numbers $r_1,\ldots,r_{d-1}$ are called the *ranks* of the tensor-train. Furthermore, we note for later use that the interfaces satisfy the recursive relations

$$g^{p+1}_{k_p} \;=\; \sum_{k_{p-1},i_p} \mathbf{U}_p(k_{p-1},i_p,k_p)\, g^p_{k_{p-1}} f^p_{i_p}, \qquad (5.2.8)$$

$$h^{p-1}_{k_{p-1}} \;=\; \sum_{i_p,k_p} \mathbf{U}_p(k_{p-1},i_p,k_p)\, f^p_{i_p} h^p_{k_p}. \qquad (5.2.9)$$

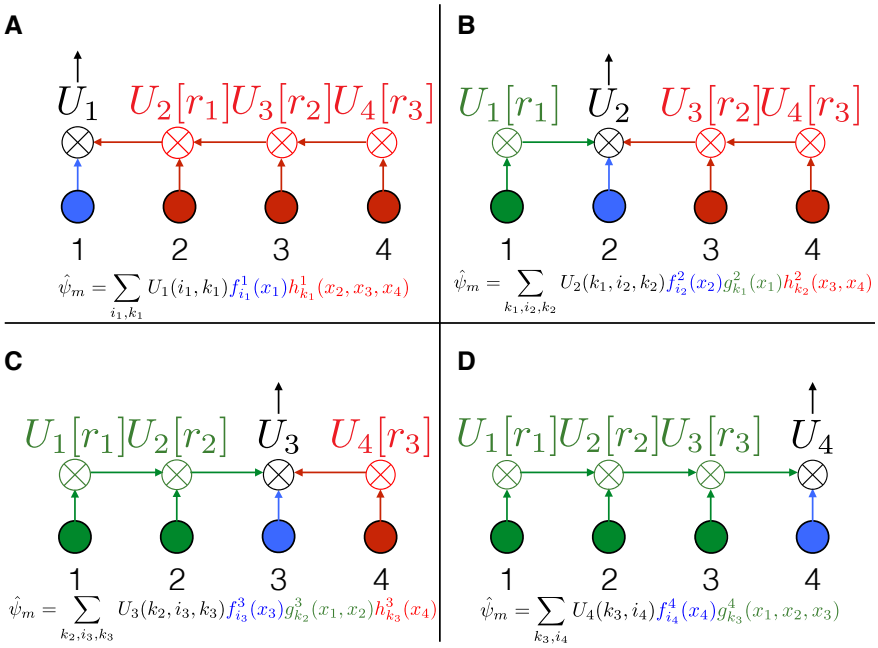

Figure 5.2: Illustration of a function $\hat{\psi}_m$ of $d = 4$ variables in tensor-train-format. The solid dots at the bottom represent the sets of one-coordinate basis functions $f^p_{i_p}$. Dots with the tensor product symbol $\otimes$ contain all products of the incoming bases, indicated by the arrows. The arrays denoted by $\mathbf{U}_p[r_p]$ select r_p linear combinations of the products, to form a new basis. We see that there are d equivalent representations of the function as a linear combination of a reduced and structured basis. If we center the representation around coordinate x_p, then the arrays $\mathbf{U}_1, \ldots, \mathbf{U}_{p-1}$ encode the information about the variables $x_1, \ldots, x_{p-1}$ into r_{p-1} functions. This process is shown in the green part of each panel. The arrays $\mathbf{U}_{p+1}, \ldots, \mathbf{U}_d$ encode the information about the variables $x_{p+1}, \ldots, x_d$ into r_p functions, which is shown in the red part of each panel. Both basis sets are combined with the one-coordinate functions $f^p_{i_p}$ (shown in blue), and a linear combination of these products is selected by $\mathbf{U}_p$, which is the final representation of $\hat{\psi}_m$. This figure has been re-used with permission from Nüske *et al.*, J. Chem. Phys. 144, 054105 (2016) [Fig. 3]. Copyright 2016 AIP Publishing LLC.

5.2.2 Alternating Linear Scheme

In order to make use of the tensor-train-format in practice, we need a method to determine the optimal components $\mathbf{U}_p$, and a way to parametrize multiple eigenfunctions $\hat{\psi}_m$. To this end, we build on two major developments in the field of tensor-trains: first, the alternating linear scheme (ALS), which is an iterative learning algorithm that arises naturally from the TT-format, see Ref. [34]. Second, the block-TT-format from Refs. [60, 61], which is a modification of tensor-trains allowing for the simultaneous approximation of multiple func-

tions using almost the same number of parameters. These concepts have led us to the algorithm outlined below. We present our optimization procedure as we have used it in the applications, and comment on its relation to the standard methods in the literature in Appendix B.1.
The idea of alternating optimization is that in each iteration step, we attempt to update one component $\mathbf{U}_p$ while keeping all others fixed. Starting from some initial guess for all $\mathbf{U}_p$, the method will first update $\mathbf{U}_1$ while $\mathbf{U}_2, \dots, \mathbf{U}_d$ are fixed, then it will update $\mathbf{U}_2$ with $\mathbf{U}_1, \mathbf{U}_3, \dots, \mathbf{U}_d$ fixed, and so on, until $\mathbf{U}_d$ is optimized. After completing this so called *forward sweep,* it will proceed backwards along the sequence of variables, which is called the *backward sweep.* This can be repeated until some convergence criterion is satisfied.
As outlined in the previous section, the component $\mathbf{U}_p$ can be read in two different ways: Either it is meant to optimally encode the information about all coordinates up to position p into r_p left interfaces $g^{p+1}_{k_p}$, or to encode the information about all coordinates $x_p, \dots, x_d$ into r_{p-1} right interfaces $h^{p-1}_{k_{p-1}}$. We will focus on the first reading during the forward sweep of the optimization, and on the second during the backward sweep. Consider the forward sweep case and assume that we attempt to optimize component $\mathbf{U}_p$ while all others are fixed. By Thm. 3.1 and recalling the recursive definition Eq. (5.2.8), the optimal left interfaces $g^{p+1}_{k_p}$ would be the linear combinations

$$g^{p+1}_{k_p}(\mathbf{U}_p) \quad = \quad \sum_{k_{p-1}, i_p} \mathbf{U}_p(k_{p-1}, i_p, k_p) g^p_{k_{p-1}} f^p_{i_p}, \tag{5.2.10}$$

that maximize the eigenvalue sum

$$L_p(\mathbf{U}_p) \quad = \quad \sum_{m=1}^{M} \hat{\lambda}_m(\mathbf{U}_p) \tag{5.2.11}$$

resulting from the generalized eigenvalue problem Eq. (3.1.7) for the basis

$$g^{p+1}_{k_p}(\mathbf{U}_p) f^{p+1}_{i_{p+1}} f^{p+2}_{i_{p+2}} \dots f^d_{i_d}, \tag{5.2.12}$$

as it combines limited information about the first p coordinates with all possible basis functions of the remaining ones. As this problem is not tractable, we use the information we have already computed, and determine the interfaces $g^{p+1}_{k_p}$ which maximize the sum Eq. (5.2.11) for the reduced basis

$$g^{p+1}_{k_p}(\mathbf{U}_p) f^{p+1}_{i_{p+1}} h^{p+1}_{k_{p+1}}, \tag{5.2.13}$$

see Fig. 5.3 for an illustration. This trick is inspired by the MALS [34] and the original DMRG algorithm.
Let us touch on the most important points of this optimization problem. First, we can set up a numerical optimization method for Eq. (5.2.11) if r_p is fixed, please see App. B.2 for an explanation. Therefore, we sequentially determine the optimal component $\mathbf{U}_p$ for increasing values of the rank r_p, and accept $\mathbf{U}_p$ as the solution if the eigenvalue sum $L_p(\mathbf{U}_p)$ matches a reference value L_{ref} up to a tolerance ϵ_{rank}. If accepted, $\mathbf{U}_p$ becomes the new p-th component, the functions $g^{p+1}_{k_p}(\mathbf{U}_p)$ become the new left interfaces at position $p+1$ and r_p is the new rank. Otherwise, r_p is increased by one and the above optimization is

Algorithm 5.1 Summary of optimization algorithm.

$q = 0$
repeat
  $q+ = 1$
  for p=1,...,d-2 **do**
    Solve Eq. (3.1.7) for the four-fold basis Eq. (5.2.14).
    Obtain eigenvalues $\hat{\lambda}_m^{p,p+1}$.
    Update reference eigenvalue sum L_{ref}.
    for $r_p = 1, \ldots$ **do**
      Optimize coefficients $\mathbf{U}_p(k_{p-1}, i_p, k_p)$ s.t. $L_p(\mathbf{U}_p) = \max$.
      if $L_p(U_p) \geq \epsilon_{\text{rank}} \cdot L_{\text{ref}}$ **then**
        Update $\mathbf{U}_p$, $g_{k_p}^{p+1}$, r_p.
        break
      end if
    end for
  end for
  Repeat this in backward direction for $p = d, \ldots, 3$
until $|L_p^q(\mathbf{U}_p) - L_p^{q-1}(\mathbf{U}_p)| < \epsilon_{\text{iter}} \quad \forall p$

repeated. The reference L_{ref} is obtained as follows: As a first step, we always evaluate a four-fold product basis defined by the functions

$$g_{k_{p-1}}^{p} f_{i_p}^{p} f_{i_{p+1}}^{p+1} h_{k_{p+1}}^{p+1}, \tag{5.2.14}$$

and solve the generalized eigenvalue problem Eq. (3.1.7) for this basis. We compute the dominant eigenvalue sum resulting from this problem,

$$L_p \quad = \quad \sum_{m=1}^{M} \hat{\lambda}_m^{p,p+1}. \tag{5.2.15}$$

The variational principle Theorem 3.1 implies that for any $\mathbf{U}_p$, the eigenvalue sum $L_p(\mathbf{U}_p)$ is bounded from above by L_p. Thus, we keep track of the maximal value obtained for L_p during the entire optimization process, and store this maximum as the reference L_{ref}. Second, we enforce the first interface function g_1^{p+1} to be the constant function. This constraint ensures that the largest eigenvalue $\hat{\lambda}_1(\mathbf{U}_p)$ is always equal to one, which turned out to be an important stabilization of the method. Third, the full optimization is considered converged if all of the objective functions $L_p(\mathbf{U}_p)$ from two subsequent forward and backward sweeps do not differ by more than a tolerance ϵ_{iter}. A summary of the complete method is given in Algorithm 5.1.

5.3 Results

In this section, we present two examples for the approximation of dominant eigenfunctions of molecular systems in the tensor-train-format. The first is

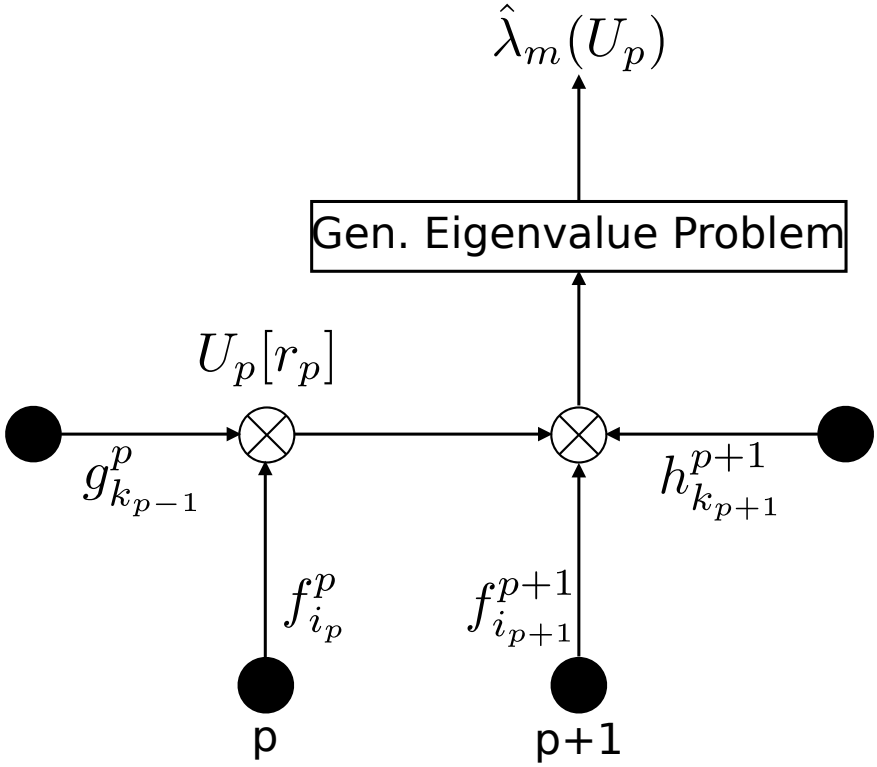

Figure 5.3: Schematic representation of the optimization problem for the component $\mathbf{U}_p$. This array selects r_p linear combinations of the products $g^p_{k_{p-1}} \cdot f^p_{i_p}$ (see Eq. (5.2.8)) to form a new basis $g^{p+1}_{k_p}(\mathbf{U}_p)$ in an optimal way. Optimality is defined as follows: We combine the basis $g^{p+1}_{k_p}(\mathbf{U}_p)$ with the one-coordinate functions $f^{p+1}_{i_{p+1}}$ and with the right interfaces $h^{p+1}_{k_{p+1}}$ at position $p+1$, to form the basis Eq. (5.2.13). For this basis, we solve the generalized eigenvalue problem Eq. (3.1.7) to obtain dominant eigenvalues $\hat{\lambda}_m(\mathbf{U}_p)$. Optimality of the $\mathbf{U}_p$ is defined by maximizing the sum Eq. (5.2.11) of the $\hat{\lambda}_m(\mathbf{U}_p)$. This figure has been re-used with permission from Nüske *et al.*, J. Chem. Phys. 144, 054105 (2016) [Fig. 4]. Copyright 2016 AIP Publishing LLC.

the ten residue peptide deca-alanine (Ala_{10}), the second is the 58 residue protein BPTI. Equilibrium trajectories that are orders of magnitude longer than the slowest relaxation timescales are available for both of these systems.
The ALS-optimization is initialized as being completely uninformed, we set all ranks $r_p = 1$ and prepare the components $\mathbf{U}_p$ to parametrize just the constant function. We choose the rank acceptance threshold as $\epsilon_{\text{rank}} = 0.995$ and the overall stopping criterion as $\epsilon_{\text{iter}} = 0.01$. Both of these choices are based on our experience with the method so far, and a more systematic or automatic choice of parameters will be subject of further research. The setting for ϵ_{rank} ensures that no important information is lost along the course of the iteration. The setting for ϵ_{iter} reflects the general level of accuracy that we can achieve for

the eigenvalues obtained from the analysis of MD data, based on the general experience we have.
Our analysis of the examples consists of four steps. First, we monitor the slowest implied timescale t_2 over the course of the optimization and compare it to reference values. Second, we analyse the structural transition encoded in the slowest eigenfunction $\hat{\psi}_2$. To this end, we evaluate the eigenfunction at all frames and histogram the resulting time series. Following the theory in section 2.4, we expect to find peaks of the population corresponding to the most negative and most positive values attained by the eigenfunction. As these peaks should correspond to metastable states, we extract representative structures for each of them in order to determine the structural transition described by the eigenfunction. Third, we attempt to identify coordinates which are relevant for the slow dynamics. To this end, we solve the following problem after every iteration step (we illustrate the problem for the forward sweep again, it works analogously for the backward sweep): after the new interface functions $g_{k_p}^{p+1}$ have been determined, we compute the best approximation to these functions in the least squares sense from the previous interfaces $g_{k_{p-1}}^{p}$ only, leaving out the one-coordinate basis for coordinate p. We record the average approximation error $E(p)$ for all of the new interface functions as a measure for the information contained in the basis at position p, see appendix B.3 for the details. Once the main iteration is completed, we re-run the ALS-iteration using only those coordinates p which satisfy that $E(p)$ is greater than a certain cutoff, and repeat this for various choices of the cutoff. By this procedure, we attempt to find a reduced set of coordinates which allows us to build an equally good model as the full one.

5.3.1 Deca Alanine

We return to the deca alanine example from section 4.2.3. The input coordinates used for this system are $d = 16$ backbone dihedral angles from the eight internal residues of the chain. This time, we left out the two outermost residues as the chain was not capped in the simulation, increasing the flexibility of the outer residues. Our set of one-coordinate basis function used for each dihedral consisted of the first $n = 7$ real Fourier (sine and cosine) waves. The lag time used for our analysis was $\tau = 2\,\text{ns}$. We can compare our results to the adapted reference Markov model from section 4.1.3, using $8^3 = 6561$ states. Recall from section 4.2.3 that the slowest dynamical process in the system is the formation of a helix and occurs at an implied timescale $t_2 \approx 7.5$-8ns.
Fig. 5.4A shows that the implied timescale t_2 as estimated by our model reaches the correct regime over the first forward sweep, then corrects slightly along the backward sweep, and remains more or less constant afterwards. Panel B displays the relative histogram of the second estimated eigenfunction $\hat{\psi}_2$ over all data points of the MD trajectory. We can identify a number of peaks of the population, of which we select the two outermost ones (around -1.3 ± 0.3 and 1.6 ± 0.2) to analyse the slow transition. An overlay of 200 random structures from each of these peaks confirms that the eigenfunction $\hat{\psi}_2$ encodes the transition from an extended structure to a helix, as expected. The final values of the least squares approximation error $E(p)$ (thus resulting from the final backward sweep) are shown in panel C. It can be observed that five interior ψ-angles

from the chain display the largest least squares error, indicating that these coordinates are important. This is consistent with the slowest process being the formation of a helix, and is strengthened further by the analysis shown in panel D. Here, we find that these five coordinates allow us to build a model which equals the quality of the full model.

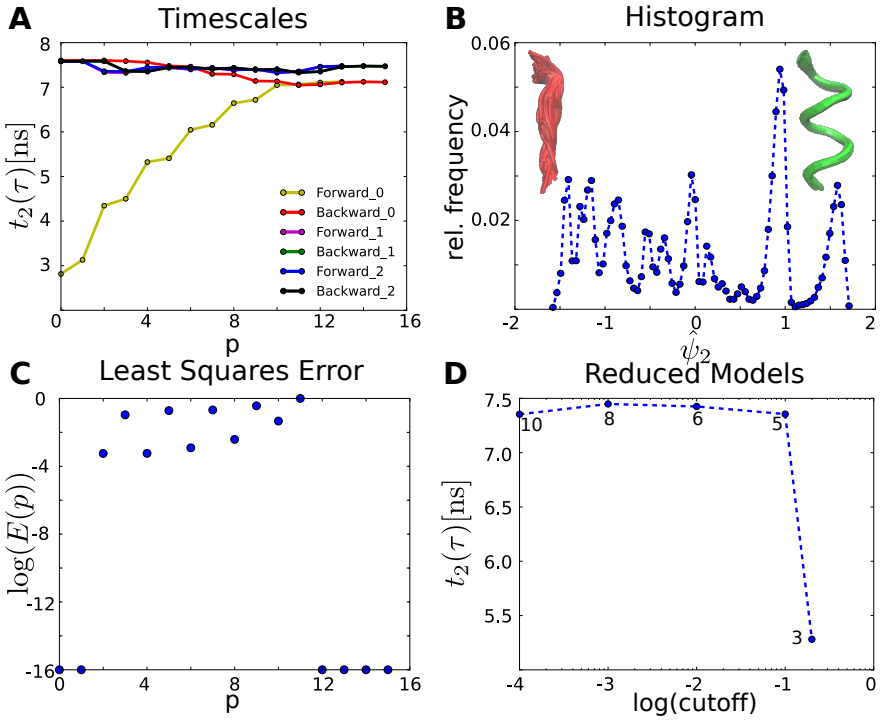

Figure 5.4: Results for deca alanine peptide. A: Second implied timescale t_2 in ns along the three forward and backward sweeps of the ALS-iteration. B: Relative histogram of the simulation data along the $\hat{\psi}_2$-coordinate. We identify two peaks of the population corresponding to the most negative (around -1.3 ± 0.3) and the most positive values (1.6 ± 0.2) of the coordinate. Extracting 200 random frames from each of these peaks and superimposing their molecular structures shows that the $\hat{\psi}_2$-coordinate encodes the transition from an elongated conformation to the helix. C: Average approximation error $E(p)$ for the newly determined interface functions at position p, normalized by the maximum error over all coordinates p. D: Second implied timescale t_2 estimated by ALS using only the coordinates satisfying that $E(p)$ is greater than the cutoff given on the horizontal axis. The small numbers next to the data points indicate the number of coordinates used in each model. This figure has been re-used with permission from Nüske *et al.*, J. Chem. Phys. 144, 054105 (2016) [Fig. 5]. Copyright 2016 AIP Publishing LLC.

5.3.2 BPTI

We also study the 1.05 ms folded-state simulation of the 58-residue protein BPTI produced on the Anton supercomputer and provided by D.E. Shaw research [62]. This large dataset has become a benchmark system used in numerous studies in recent years. The slowest structural transition included in the C_α dynamics has been identified by other kinetic models to be on a timescale $t_2 \approx 40\,\mu$s, see Ref. [63, 64] for details.

The coordinates used in order to apply our method are the distances between all C_α atoms in the system which are at least three residues apart. For each distance, we construct a minimal basis set consisting of only $n = 2$ functions: The first is the constant, while the second is a smooth switching function indicating whether a contact between two C_α atoms has formed or not:

$$f_2^p(x_p) \quad = \quad \frac{1 - \left(x_p/r_0\right)^{64}}{1 - \left(x_p/r_0\right)^{96}}, \tag{5.3.1}$$

where x_p is the C_α distance under consideration and $r_0 = 0.7$ nm is an empirically obtained cutoff distance. The function is mostly equal to one for $x_p < r_0$, indicating that a contact between the two atoms has formed, while it is mostly zero for $x_p > r_0$, thus indicating that the contact is broken. The function smoothly transitions between one and zero in a small neighborhood of r_0. With this basis, it is easy to reduce the number of input coordinates by checking if a contact has at least once transitioned from the formed to the broken state or vice versa, and only using those contacts while leaving out all others. For the given data set, this preprocessing reduces the number of contacts from initially around 1500 to $d = 258$. Still, this system is a lot larger than the previous one. We conduct our analysis at lag time $\tau = 5\,\mu$s.

Figure 5.5A shows that again, the second implied timescale t_2 rises to the appropriate regime over the course of the first forward sweep, improves further during the first backward sweep, and changes only slightly afterwards. The histogram of the data over the estimated second eigenfunction $\hat{\psi}_2$ displays two clearly distinguishable peaks at its extremal values (around -0.2 ± 0.5 and 6.5 ± 1.0). A set of 200 molecular structures extracted from these peaks confirm that $\hat{\psi}_2$ encodes the structural transition as it was determined previously, namely a re-folding of the loop on the N-terminal side of the backbone [63, 64]. The results of the least squares approximations are not as clear as in the previous example. It is apparent from Fig. 5.5C that more than 100 of the coordinates at the end of the sequence are identified as completely unimportant, with $E(p) \approx 0$. This finding is in agreement with the fact that the part of the chain near the C-terminus is not involved in the slow transition. For the remaining 140 coordinates, $E(p)$ varies between 10^0 and 10^{-6}, but there is no obvious gap or cutoff which separates the important from the unimportant coordinates. However, such a cutoff can be determined by building various reduced models. We can conclude from Fig. 5.5D that choosing the cutoff as $E(p) \geq 10^{-3}$, we can determine a set of 58 coordinates which are sufficient to build a reduced model of the same quality as the full model, while using an even higher cutoff entails loss of information.

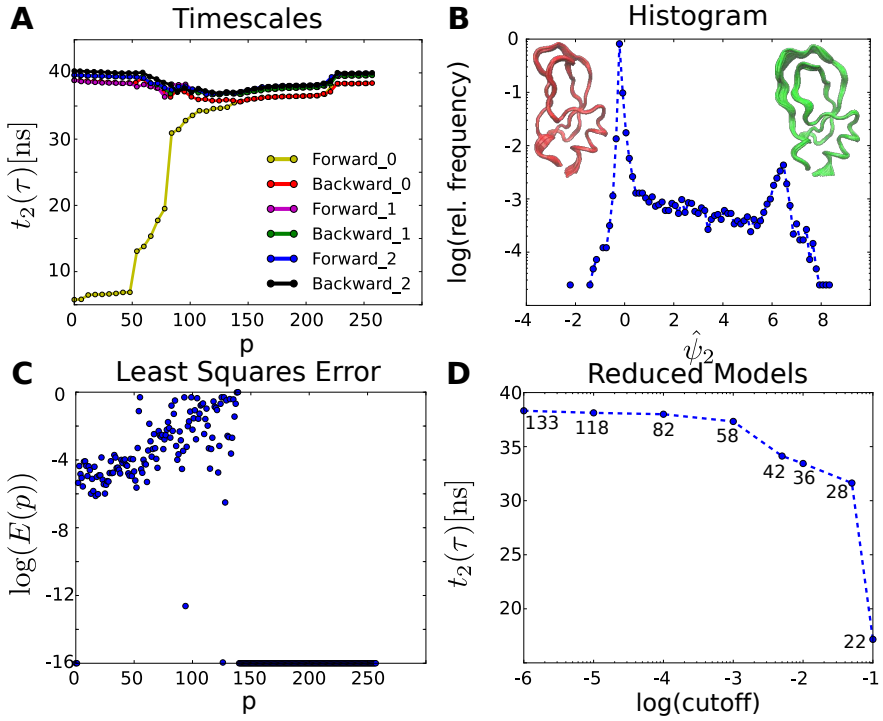

Figure 5.5: Results for BPTI. A: Second implied timescale t_2 in μs along the three forward and backward sweeps of the ALS-iteration. B: Relative histogram of the simulation data along the $\hat{\psi}_2$-coordinate. We identify two peaks of the population. Extracting 200 random frames from each of these peaks and superimposing their molecular structures shows that the $\hat{\psi}_2$-coordinate encodes the structural transition observed previously in the literature. C: Average approximation error $E(p)$ for the newly determined interface functions at position p, normalized by the maximum error over all coordinates p. D: Second implied timescale t_2 estimated by ALS using only the coordinates satisfying that $E(p)$ is greater than the cutoff given on the horizontal axis. The small numbers next to the data points indicate the number of coordinates used in each model. This figure has been re-used with permission from Nüske *et al.*, J. Chem. Phys. 144, 054105 (2016) [Fig. 6]. Copyright 2016 AIP Publishing LLC.

5.4 Conclusions

The results of this chapter suggest that the TT-approach is suitable for selecting a low-rank tensor product representation to approximate the high-dimensional eigenfunctions of molecular conformation spaces. As the resulting eigenfunction approximations are directly related to the molecular coordinates, they can be interpreted via post-processing methods, and may serve as a way to select the most relevant molecular features that are good reaction coordinates. In the two examples presented, specific coordinates could be recognized as relevant for the slow kinetics or as irrelevant.
Future work will have to address the question of how stably this method can

perform for significantly larger systems. The success of our iterative scheme depends on the ranks r_p, as the computational effort grows with increasing ranks. It will be important to see how these ranks can be controlled for large systems. Also, we expect the ordering of input coordinates to play an important role in the future. Apart from that, we were able to use equilibrium trajectories in the examples presented so far. For large systems, it is usually impossible to provide equilibrium data because of the sampling problem. In the next chapter, we discuss the use of non-equilibrium simulations in the context of Markov state model construction. The results will serve as a basis to apply the general VAC to non-equilibrium data.

Chapter 6

MSM Estimation from Short Simulations

In this chapter, we focus on the special case of Markov state models (MSMs) from section 3.3, but we will explain in the end how the techniques from this chapter generalize to the case of an arbitrary basis set. We will discuss how Markov models can be estimated from ensembles of relatively short non-equilibrium simulations. The text is adapted from Ref. [65].

One of the strengths of Markov models is that the simulations used to construct them do not necessarily need to sample from the global equilibrium distribution, as only conditional transition probabilities between the states are required [4]. In particular, at least in principle, these transition probabilities can be obtained without bias from simulations started out of local equilibrium in each state which only run for the length of a single lag time step. However, it is much more practical to produce simulations that are longer than one lag time and estimate MSMs by counting transitions along these trajectories. Even if the simulations are started out of local equilibrium, the distribution deviates from local equilibrium over time until global equilibrium is restored. The estimation of transition probabilities is therefore subjected to a bias [9]. In order to keep the bias small, it must be assumed that local equilibrium is approximately restored after every time step.

The effect of the initial distribution onto the MSM quality or even the justification of using an MSM for data analysis has been controversially discussed, and this issue has not been resolved yet. At least three ideas have been discussed [66]:

1. This effect exists [9], but may be small and can be ignored in practice.
2. We can reduce the effect of non-equilibrium starting points by discarding the first bit of simulation trajectories, enough to reach local equilibrium [23].
3. We can avoid this problem by preparing local equilibrium distributions in the starting states using biased simulations and then shooting trajectories out of them [35, 36, 37, 38].

Here we qualify and quantify these ideas by systematically analyzing the effect

of non-equilibrium starting conditions onto MSM quality, and we suggest effective correction mechanisms. Throughout the chapter, we use the term "non-equilibrium" to describe the problem that simulations are started from a distribution which is not in global equilibrium, and their simulation time is too short to reach that global equilibrium. Briefly, our main results are:

1. We provide an expression for the error between unbiased transition probabilities and the expected estimate from many simulations running for multiple discrete time steps, see section 6.1. We find that there is no fundamental advantage of starting simulations in local equilibrium. Rather, the estimation error depends on the discretization, the simulation length and the lag time. In the limit of long lag times and fine discretization, MSMs are estimated without bias even when non-equilibrium starting points are used. However, for a given discretization the lag time required to practically achieve a small estimation bias might be large.

2. We derive an unbiased MSM estimator that corrects the error due to non-equilibrium starting conditions at short lag times, by exploiting the framework of *observable operator models* (OOMs) - see sec. 6.2. OOMs are powerful finite-dimensional models that provide unbiased estimates of stationary and kinetic properties of stochastic processes under fairly mild assumptions, see [39, 40, 41]. Most importantly, OOMs can be estimated from non-equilibrium simulations [41] and are not limited to a local equilibrium assumption.

3. We utilize the fact that exact relaxation timescales that are not contaminated by the MSM projection error (i.e. quality of the coordinates and the clustering used) can be estimated using the OOM framework. The difference between the unbiased estimate and the uncorrected or corrected MSM estimate is very insightful as it provides an indicator of the quality of the MSM discretization. If this difference is too large, it is suggested to rather improve the coordinate selection or discretization used for MSM construction and re-analyze. Note that while OOMs offer the more general theory, they are not as easy to interpret and their estimation from finite data is not as stable and mature as MSM estimation.

We also provide a meaningful strategy to select the model rank of an OOM which is required in order to obtain practically useful estimates, by using a statistical analysis of singular values of the count matrix (sec. 6.2.4).

Sec. 6.3, demonstrates the usefulness of the OOM framework for two model systems and MD simulation data of alanine dipeptide. We show that accurate estimates of spectral and stationary properties can be obtained from short non-equilibrium simulations, even for short lag times or poor discretizations. We explain how the discretization quality is revealed by the difference between spectral estimates of MSM and OOM. We also show that the rank selection strategy helps to choose a suitable model rank even for small lag times, when no apparent timescale separation can be utilized.

As an illustration, consider the one-dimensional model system governed by the potential shown in Fig. 6.1 A, see sec. 6.3.1 for details. We study the estimation of a Markov model using the two state discretization indicated in panel A of Fig. 6.1. For various lag times, we investigate the expected transition matrix

if 90 per cent of the simulations are started from local equilibrium within state 1, while the other 10 per cent are started from local equilibrium within state 2. Note that we do not use any simulation data here, we only compute expected values over an ensemble of trajectories, with the trajectory length set to 2000 steps, which is shorter than the slowest relaxation timescale.

For short lag times, the standard MSM provides a strongly biased estimate of the equilibrium population of the two wells (Fig. 6.1C, green curve). For longer lag times, the MSM converges towards the correct equilibrium population, but the bias only disappears when the lag time approaches the longest relaxation timescale of the system, so if the initial distribution is far from equilibrium this can entail a significant error at practically feasible lag times. In contrast, the corrected MSM estimate proposed in this chapter achieves the correct estimate of equilibrium populations even at short lag times (Fig. 6.1C, red curve). The standard MSM relaxation timescales are underestimated at short lag times, consistent with previous variational results (Thm. 3.1 and Ref. [52]), but they can be improved by using the unbiased MSM estimator proposed here (Fig. 6.1E). The OOM can provide a model-free estimate of the relaxation timescale that is unbiased at a relatively short lag time (Fig. 6.1E, blue line). The difference between the OOM and the corrected MSM estimate (blue versus red lines in Fig. 6.1E) is an indicator of the MSM model error due to the state space discretization. Please note that all MSM results in this figure can be dramatically improved if a finer clustering is used. For example, if the five state partitioning from Fig. 6.1B is used instead, the estimation of stationary properties converges much faster (Fig. 6.1D), and there is hardly a difference between the timescales estimated by a direct and an unbiased MSM (Fig. 6.1F).

6.1 MSM Estimation from Simulations with Arbitrary Starting Points

6.1.1 Count Matrix and Transition Matrix

In this chapter, we assume that the transfer operator is of rank M at lag time τ, and that the transition kernel possesses a density which can then be written as

$$p(x, y; \tau) \quad = \quad \sum_{m=1}^{M} \lambda_m(\tau)\, \psi_m(x)\, \pi(y)\, \psi_m(y). \tag{6.1.1}$$

Note that exact equality in Eq. (6.1.1) is an assumption, but often it is satisfied approximately for a large range of lag times τ. Throughout the chapter, we will consider decompositions of state space into disjoint sets $S_1, ..., S_N$, where $S = \bigcup_i S_i$, as in sec. 3.3. The indicator function of set S_i is denoted in brief by χ_i. For a simulation of the continuous dynamics which samples positions at discrete time steps $(k-1)\Delta t$, $k = 1, \ldots, K$, we will denote the position at the k-th time step simply by X_k, $k = 1, \ldots, K$, s.t. K is the total number of time steps in the simulation. We use the symbol $\mathbf{Y}$ as a shorthand notation for an entire simulation. If multiple different simulations need to be distinguished, we will denote them by $\mathbf{Y}_q$, $q = 1, \ldots, Q$, i.e. Q is the total number of available simulations.

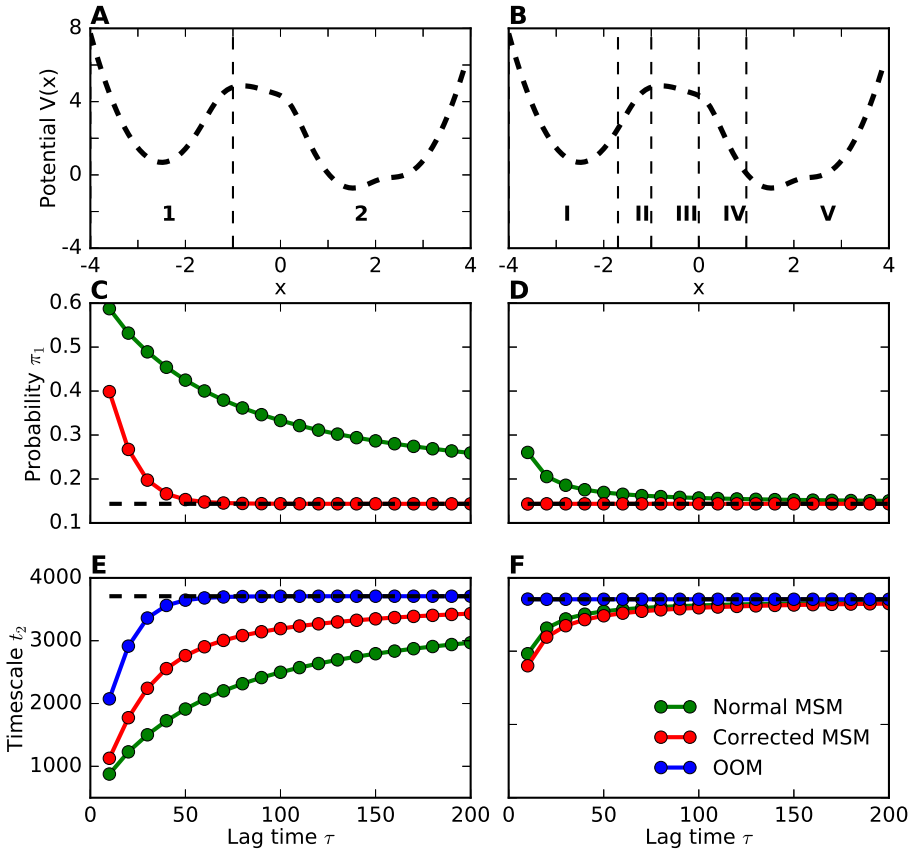

Figure 6.1: A: One-dimensional potential function and discretization into two states. B: The same potential with a five state discretization. C, D: Estimates for the equilibrium probability of state 1 from the direct MSM (green) and the unbiased MSM (red), reference in black. E, F: Estimates for the slowest relaxation timescale t_2 from a direct MSM (green), c.f. Eq. (6.1.16), the unbiased MSM (red), c.f. Eqs. (6.2.5-6.2.6), and the spectral OOM estimation (blue), Eqs. (6.2.29-6.2.30). The black dashed line corresponds to the reference value. This figure has been re-used with permission from Ref. [65], Nüske *et al.*, J. Chem. Phys. (2017, in press) [Fig. 1]. Copyright 2017 AIP Publishing.

For a trajectory as above, we define the empirical histograms and correlations (also called state-to-state time-correlations) as follows:

$$\mathbf{s}(i) := \frac{1}{K-2\tau}\sum_{k=1}^{K-2\tau}\chi_i(X_k), \tag{6.1.2}$$

$$\mathbf{S}^{\tau}(i,j) := \frac{1}{K-2\tau}\sum_{k=1}^{K-2\tau}\chi_i(X_k)\chi_j(X_{k+\tau}), \tag{6.1.3}$$

$$\mathbf{S}_r^{2\tau}(i,j) := \frac{1}{K-2\tau}\sum_{k=1}^{K-2\tau}\chi_i(X_k)\chi_r(X_{k+\tau})\chi_j(X_{k+2\tau}). \tag{6.1.4}$$

Apart from the pre-factor, Eq. (6.1.3) agrees with Eq. (3.2.1) for the basis set χ_i, $i = 1, \ldots, N$. Up to the normalization, the matrix $\mathbf{S}^\tau \in \mathbb{R}^{N\times N}$ is a *count matrix* because it simply counts the number of transitions from state S_i to S_j over a time window τ that have occurred in the simulation, while the vector $\mathbf{s} \in \mathbb{R}^N$ counts the total visits to state S_i and corresponds to the i-th row sum of $\mathbf{S}^\tau$. For each set S_r, the matrix $\mathbf{S}_r^{2\tau} \in \mathbb{R}^{N\times N}$ is proportional to a two-step count matrix counting subsequent transitions from state S_i to S_r and on to state S_j. At first sight, it may seem confusing that $\mathbf{S}^\tau$ and $\mathbf{s}$ only count transitions and visits up to time $K - 2\tau$, but further below, we will use all three matrices in conjunction which requires estimating all of them over the same part of the data. We will continue to refer to these matrices as count matrix, count vector and two-step count matrix in what follows. Also note that in the literature, the count matrix and vector are often denoted by $\mathbf{C}^\tau$, $\mathbf{c}$, but we will use these symbols differently in what follows. Let us note at this point that $\mathbf{s}$, $\mathbf{S}^\tau$, $\mathbf{S}_r^{2\tau}$ can be seen as random variables that map a (stochastic) trajectory $\mathbf{Y}$ of discrete time steps to the values given in Eqs. (6.1.2-6.1.4). To emphasize this dependence, we will also write $\mathbf{s}(\mathbf{Y})$, $\mathbf{S}^\tau(\mathbf{Y})$, $\mathbf{S}_r^{2\tau}(\mathbf{Y})$ if appropriate.

We are concerned with the estimation of a transition probability matrix between the sets S_i of a given discretization of state space. If the process is in equilibrium, the conditional transition probabilities can be expressed as

$$\mathbf{T}_{Eq}^\tau(i,j) = \frac{\mathbb{P}\left(X_t \in S_i,\, X_{t+\tau} \in S_j\right)}{\mathbb{P}(X_t \in S_i)} \tag{6.1.5}$$

$$= \frac{\int_{S_i} \mathrm{d}x \int_{S_j} \mathrm{d}y\, \pi(x)\, p(x,y;\tau)}{\int_{S_i} \mathrm{d}x\, \pi(x)} \tag{6.1.6}$$

$$= \frac{\mathbf{C}_{Eq}^\tau(i,j)}{\pi_i}. \tag{6.1.7}$$

Here, we have defined the *equilibrium correlation* between sets S_i and S_j by the nominator of Eq. (6.1.6) and denoted it by $\mathbf{C}_{Eq}^\tau(i,j)$. Also, we have adopted the usual notation $\pi_i = \int_{S_i} \mathrm{d}x\, \pi(x)$ for the equilibrium probabilities of the discrete states. From a long simulation X_k, $k = 1, \ldots, K$ that samples points from the stationary density π, the matrix $\mathbf{T}_{Eq}^\tau$ can be estimated by the formula

$$\mathbf{T}_{Eq}^\tau(i,j) \approx \frac{\mathbf{S}^\tau(i,j)}{\mathbf{s}(i)}. \tag{6.1.8}$$

6.1.2 Starting from local Equilibrium

In practice, producing simulation data that samples from the global equilibrium density π is often not tractable. One of the strengths of Markov models is the fact that the transition matrix can also be expressed in terms of local equilibrium densities

$$\pi_{S_i}(x) = \frac{1}{\pi_i}\chi_i(x)\pi(x). \tag{6.1.9}$$

The density π_{S_i} is the normalized restriction of π to state S_i. A Markov model transition matrix can also be estimated by preparing an ensemble of trajectories in such a way that, within each state, the distribution of starting points

equals the local density Eq. (6.1.9). These trajectories are simulated until time τ, and the fraction of trajectories starting in S_i and ending up in S_j provides an estimate for the transition matrix entry $\mathbf{T}_{Eq}^{\tau}(i,j)$ [67, 35]. To see this, note that in the setting just described, the initial distribution is a convex combination ρ_L of the local densities π_{S_i}:

$$\rho_L = \sum_{i=1}^{N} a_i \pi_{S_i}, \tag{6.1.10}$$

$$\sum_{i=1}^{N} a_i = 1. \tag{6.1.11}$$

Here, a_i is the probability to start in state S_i. Upon replacing π by ρ_L in Eq. (6.1.6), it follows that

$$\mathbf{T}_{Eq}^{\tau}(i,j) = \frac{\int_{S_i} \mathrm{dx} \int_{S_j} \mathrm{dy}\, \rho_L(x) p(x,y;\tau)}{\int_{S_i} \mathrm{dx}\, \rho_L(x)}. \tag{6.1.12}$$

Only very short trajectories and knowledge of the local densities are needed for the application of this method. However, this method suffers from three major disadvantages: first, the intermediate data points of the simulations cannot be used. Second, estimation of the local densities requires the use of biased sampling methods, which is a significant extra effort and entails additional difficulties. Third, changing the discretization requires to redo the simulations, which is not acceptable if a suitable discretization is not easy to find.

6.1.3 Multiple-Step Estimator

A common way to construct MSMs in practice is by conducting a large set of distributed simulations $\mathbf{Y}_q$, $q = 1, \ldots, Q$ of lengths that are shorter than the largest relaxation timescales of the system, but are longer than the lag time τ. For our theoretical investigation we will assume that each of these trajectories has the same length of K stored simulation steps, but for the estimators we will be deriving later uniform length is not a requirement, see appendix C.4.

The simulations are started from some arbitrary initial distribution at time $k = 1$. The transition probability matrix is estimated by replacing $\mathbf{S}(i,j)$ and $\mathbf{s}(i)$ by their empirical mean values over all simulations $\mathbf{Y}_q$. These are defined by the following equations, where we include the corresponding definition for $\mathbf{S}_r^{2\tau}$ for later use:

$$\overline{\mathbf{s}} = \frac{1}{Q} \sum_{q=1}^{Q} \mathbf{s}(\mathbf{Y}_q), \tag{6.1.13}$$

$$\overline{\mathbf{S}}^{\tau} = \frac{1}{Q} \sum_{q=1}^{Q} \mathbf{S}^{\tau}(\mathbf{Y}_q), \tag{6.1.14}$$

$$\overline{\mathbf{S}}_r^{2\tau} = \frac{1}{Q} \sum_{q=1}^{Q} \mathbf{S}_r^{2\tau}(\mathbf{Y}_q). \tag{6.1.15}$$

In analogy to Eq. (6.1.8), the transition matrix is then estimated by

$$\overline{\mathbf{T}}^{\tau}(i,j) \quad = \quad \frac{\overline{\mathbf{S}}^{\tau}(i,j)}{\overline{\mathbf{s}}(i)}. \tag{6.1.16}$$

Additional constraints can be incorporated in order to obtain more specific estimators than Eq. (6.1.16), such as estimators obeying detailed balance [14, 9, 15].

The argument from Sec. 6.1.2 cannot be transferred directly to a multiple step estimator like Eqs. (6.1.13-6.1.14): Even if the simulations are started from local equilibrium, this property is lost after the first simulation step, and the resulting estimates are no longer unbiased. A detailed illustration of this phenomenon has been provided by Ref. [9, Fig. 4], and we repeat it here in Figure 6.2. It can be argued that if the discretization is chosen well enough such that the dynamics equilibrates to an approximate local equilibrium within all states over a single time step, the bias can be expected to be very small. This assumption is difficult to check or quantify in practice. In the next section, we analyze the bias introduced by the multiple step estimator, as well as its dependence on the lag time and simulation length.

6.1.4 Estimation Error from Non-Equilibrium Simulations

Now we study the effect of using an initial distribution of simulation data that is not in local equilibrium when the transitions are counted. This deviation from local equilibrium could come either from the fact that we start trajectories in an arbitrary initial condition, or that our trajectories exceed the lag time τ such that an initially prepared local equilibrium is lost for all transition counts harvested after the first one (sec. 6.1.3).

Let ρ denote the *empirical* distribution sampled by the simulations. We need to study the error between the equilibrium transition matrix $\mathbf{T}_{Eq}^{\tau}$ and the asymptotic limit of Eq. (6.1.16). To this end, we study the asymptotic limits of $\overline{\mathbf{S}}^{\tau}(i,j)$ and $\overline{\mathbf{s}}(i)$ in the limit of infinitely many simulations, $Q \to \infty$, but each having finite lengths:

$$\mathbf{C}_{\rho}^{\tau}(i,j) \quad := \quad \mathbb{E}\left(\mathbf{S}^{\tau}(i,j)\right), \tag{6.1.17}$$

$$\mathbf{c}_{\rho}(i) \quad := \quad \mathbb{E}\left(\mathbf{s}(i)\right), \tag{6.1.18}$$

$$\mathbf{T}_{\rho}^{\tau}(i,j) \quad := \quad \frac{\mathbf{C}_{\rho}^{\tau}(i,j)}{\mathbf{c}_{\rho}(i)}. \tag{6.1.19}$$

Thus, we use the symbols $\mathbf{C}_{\rho}^{\tau}$, $\mathbf{c}_{\rho}$ for the *expected* count matrix and vector of total counts associated with the empirical distribution ρ. Using the spectral decomposition Eq. (6.1.1), the expected count matrix can be expressed in terms of the spectral components of the dynamics:

$$\mathbf{C}_{\rho}^{\tau}(i,j) \quad = \quad \sum_{m=1}^{M} \lambda_m(\tau)\langle\chi_i,\psi_m\rangle_{\rho}\langle\chi_j,\psi_m\rangle_{\pi}, \tag{6.1.20}$$

$$\langle\chi_i,\psi_m\rangle_{\rho} \quad = \quad \int_S \mathrm{d}x\, \chi_i(x)\psi_m(x)\rho(x), \tag{6.1.21}$$

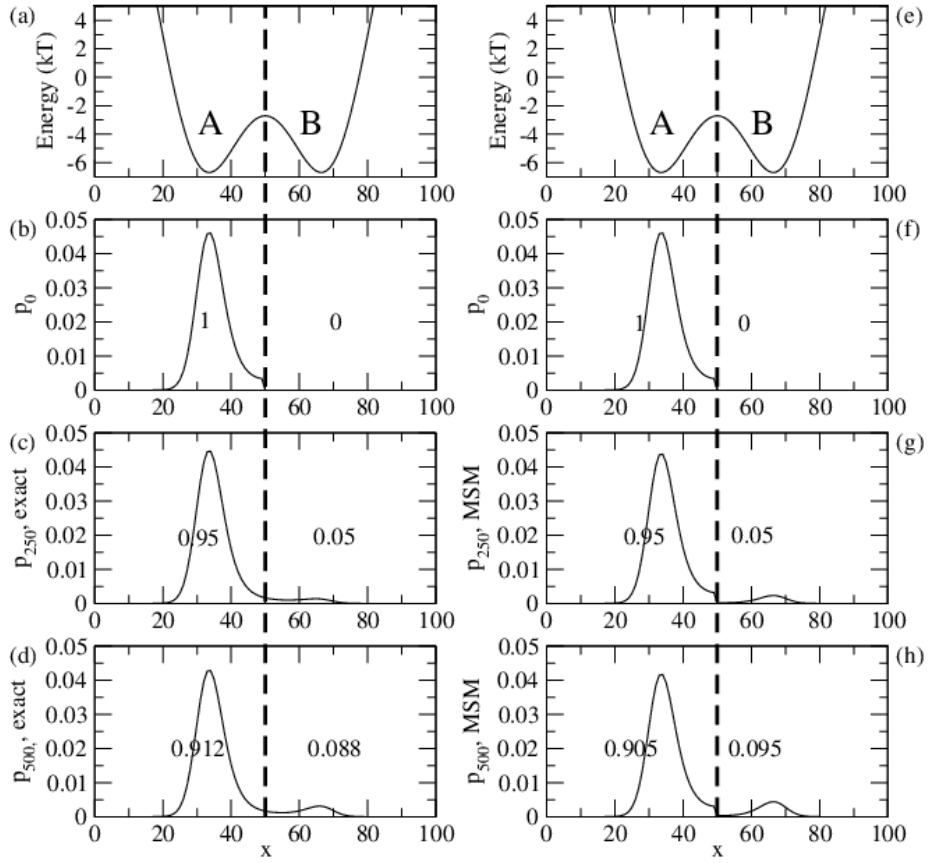

Figure 6.2: Loss of local equilibrium property illustrated by comparing the dynamics of the diffusion in a double-well potential (a,e) at time steps 0 (b), 250 (c), 500 (d) with the predictions of a Markov model parameterized at lag time $\tau = 250$ at the same times 0 (f), 250 (g), 500 (h). Please refer to the supplementary material of Ref. [9] for details of the system. (b, c, d) show the true distribution of the system (solid black line) and the probabilities associated with the two discrete states left and right of the dashed line. The numbers in (f, g, h) are the discrete state probabilities $p_i(k\tau)$, $i = 1, 2$, $k = 0, 1, 2$, predicted by the Markov model. The solid black lines shows the hypothetical density $p_i(k\tau)\pi_{S_i}$ that is inherently assumed when estimating a Markov model by counting transitions over multiple steps. This figure has been re-used with permission from Prinz *et al.*, J. Chem. Phys. 134, 174105 (2011) [Fig. 4]. Copyright 2011 American Institute of Physics.

$$\langle \chi_i, \psi_m \rangle_\pi = \int_S \mathrm{d}x\, \chi_i(x)\psi_m(x)\pi(x). \tag{6.1.22}$$

In matrix form, Eq. (6.1.20) can be written as

$$\mathbf{C}_\rho^\tau = \mathbf{Q}_\rho \mathbf{\Lambda}(\tau)\mathbf{Q}_\pi^T, \tag{6.1.23}$$

$$\mathbf{Q}_\rho(i,m) = \langle \chi_i, \psi_m \rangle_\rho, \tag{6.1.24}$$

$$\mathbf{Q}_\pi(j,m) = \langle \chi_j, \psi_m \rangle_\pi. \tag{6.1.25}$$

These matrices contain the MSM projections of the true eigenfunctions, i.e. their approximations by step functions, that is extensively discussed in [12, 9]. Let us emphasize that Eq. (6.1.20) also holds for arbitrary basis functions, i.e. χ_i is not required to be a basis of indicator functions. Thus, it is the most general expression for a correlation matrix from Markovian dynamics.

Summation over j shows that

$$\mathbf{c}_\rho(i) \quad = \quad \langle \chi_i \rangle_\rho. \tag{6.1.26}$$

It follows from Eq. (6.1.20) that the spectral expansion of $\mathbf{C}^\tau_{Eq}$ is given by

$$\mathbf{C}^\tau_{Eq}(i,j) \quad = \quad \sum_{m=1}^{M} \lambda_m(\tau) \langle \chi_i, \psi_m \rangle_\pi \langle \chi_j, \psi_m \rangle_\pi, \tag{6.1.27}$$

using the fact that for trajectories started from global equilibrium we have $\rho = \pi$. Combining Eqs. (6.1.20), (6.1.26) and (6.1.27), we obtain an expression for the estimation error $\mathbf{E}^\tau := \mathbf{T}^\tau_\rho - \mathbf{T}^\tau_{Eq}$:

$$\begin{aligned}
\mathbf{E}^\tau(i,j) \quad &= \quad \frac{\mathbf{C}^\tau_\rho(i,j)}{\mathbf{c}_\rho(i)} - \frac{\mathbf{C}^\tau_{Eq}(i,j)}{\pi_i} && (6.1.28)\\
&= \quad \sum_{m=2}^{M} \lambda_m(\tau) \langle \chi_j, \psi_m \rangle_\pi \left[\frac{\langle \chi_i, \psi_m \rangle_\rho}{\langle \chi_i \rangle_\rho} - \frac{\langle \chi_i, \psi_m \rangle_\pi}{\langle \chi_i \rangle_\pi} \right] && (6.1.29)\\
&= \quad \sum_{m=2}^{M} \lambda_m(\tau) \langle \chi_j, \psi_m \rangle_\pi \left[\frac{\langle \chi_i, \psi_m - q_{im}\psi_1 \rangle_\rho}{\langle \chi_i \rangle_\rho} \right], && (6.1.30)
\end{aligned}$$

where $q_{im} = \frac{\langle \chi_i, \psi_m \rangle_\pi}{\langle \chi_i \rangle_\pi}$, and we were able to drop the $m = 1$ terms on both sides as they are equal. Inspecting this expression leads to a number of insights that are practically important for analyzing simulation data with MSMs:

1. **MSM estimation from long trajectories**: In the limit that our trajectories are longer than the timescale of the slowest process, the empirical distribution ρ converges to the equilibrium distribution π, and the bias becomes zero. This offers an explanation why MSMs built from ultra-long simulations [62, 68] are quite well-behaved and have been extensively used for benchmarking and method validation.

2. **MSM estimation from short trajectories**: Even if the trajectories are not long enough to reach global equilibrium, because of Eq. (3.4.4), the bias decays multi-exponentially with the lag time τ. This is an important insight, because MSMs are in practice constructed in the limit of long enough lag times in which the timescale estimates converge [69, 9], and the above equation shows that this limit is meaningful as it approaches an unbiased estimate.

3. **Dependence of bias on the discretization error**: The above formula reflects the well-known insight that Markov models are free of bias if the discretization perfectly approximates the dominant eigenfunctions, meaning that the eigenfunctions are constant on the states S_i [69, 9].

4. **Consequences for adaptive sampling**: Previous adaptive sampling approaches have suggested to prepare an initial local equilibrium distribution in order to shoot trajectories out of selected states [35]. The above analysis shows that this strategy is effective if we only count a single transition out of the state, but is ineffective when longer trajectories are shot. In the latter case, it is simpler to ignore the initial distribution and to reduce the effect of bias by extending the lag time τ, see again Fig. 6.1 and also the next example.

6.1.5 Example

Before proceeding, we illustrate these findings by re-visiting the one-dimensional model system presented above. We study the same two different discretizations, the two state model from panel A of Fig. 6.3 and the five state discretization shown in Fig. 6.3 B. Again, simulations are initiated from local equilibrium in states 1 and 2 of the coarse discretization, with $a_1 = 0.9$, $a_2 = 0.1$. We study the expected estimate of the equilibrium probability of state 1, which equals the equilibrium probability of states I and II for the finer state definition. Panels C and D of Fig. 6.3 show the respective estimates for the coarse and fine discretization as a function of the lag time, for simulation lengths $K = 1000, 2000, 5000, 10000, 50000$. Indeed, the estimates improve if the lag time is increased, if the simulation length is increased, or if the discretization is improved. From the coarse partitioning example, we conclude that relaxation to global equilibrium can be required in order to obtain unbiased estimates from simulations initiated out of local equilibrium.

6.2 Correction of Estimation Bias using Observable Operator Models

In this section, we show how to go beyond just using a longer lag time τ and suggest correction mechanisms to obtain the correct equilibrium transition matrix $\mathbf{T}_{Eq}^{\tau}$ (Eqs. (6.1.5-6.1.7)) from an ensemble of short simulations. This can be accomplished regardless of the starting distribution being in global equilibrium, in local equilibrium, or far from any equilibrium.

As discussed above, limitations of MSMs include the assumption of Markovianity, sensitivity to projection error, and sensitivity to the distribution of trajectory starting points. All of these limitations can be overcome by realizing that molecular dynamics that is observed in a chosen set of variables, reaction coordinates or order parameters at a certain lag time τ can be *exactly* described by projected Markov models (PMMs) [63]. This insight allows us to employ estimators that are not affected by the MSM limitations, such as hidden Markov models (HMMs) [63] or observable operator models (OOMs) [39, 40, 41], that operate on the discretized state space.

Here, we employ OOMs in order to get improved MSM estimators that are not subject to the bias caused by a non-equilibrium distribution of the trajectories used. In a nutshell, OOMs are spectral estimators able to provide unbiased estimates of stationary and dynamical quantities for dynamical systems that can be well described by a finite number of dynamical components. Here

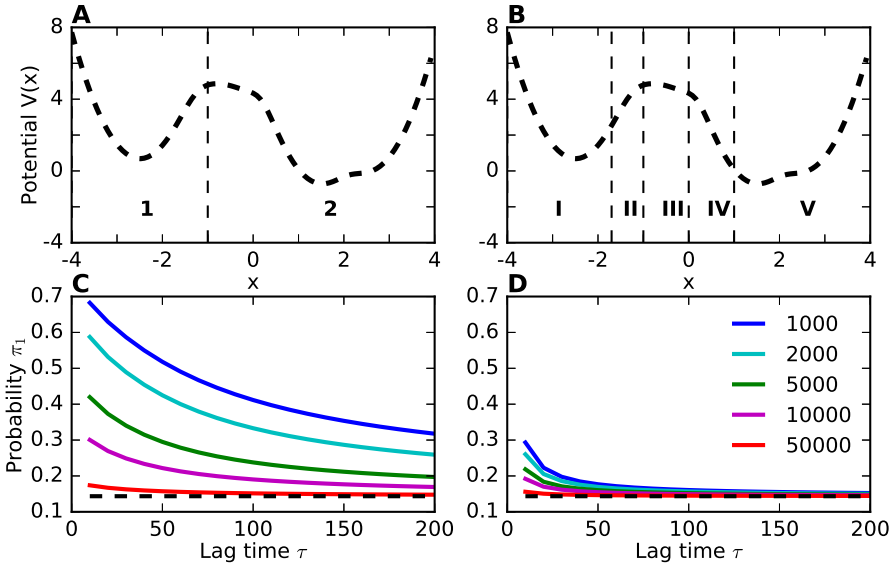

Figure 6.3: A, B: One-dimensional potential function with two different discretizations into two states and five states, resp. C: Expected estimate of the equilibrium probability of state 1 as a function of the lag time, for simulation lengths $K = 1000, 2000, 5000, 10000, 50000$, and using the discretization from panel A. The simulations are initiated in local equilibrium in both states 1 and 2, but predominantly in state 1 ($a_1 = 0.9$, $a_2 = 0.1$). D: The same for the five state discretization from panel B. This figure has been re-used with permission from Ref. [65], Nüske *et al.*, J. Chem. Phys. (2017, in press) [Fig. 3]. Copyright 2017 AIP Publishing.

we only summarize a few aspects of OOMs that are relevant to the present paper and present an algorithm that can be used to estimate MSMs without bias from the initial trajectory distribution. To fully understand the theoretical background and derivation, please refer to [39, 40, 41].

6.2.1 Observable Operator Models

Observable operator models (OOMs) provide a framework that completely captures the dynamics of a stochastic dynamical system by a finite-dimensional algebraic system if only a finite number M of relaxation processes contribute in Eq. (6.1.1), see Refs. [39, 40]. For molecular dynamics, this property is achieved if we observe and model the dynamics at a finite lag time τ. The *full-state observable operator* Ξ_S is an $M \times M$ matrix which contains the scalar products between the eigenfunctions:

$$\Xi_S(m, m') \quad = \quad \lambda_m(\tau) \int_S \mathrm{d}x \, \psi_m(x) \psi_{m'}(x) \pi(x). \tag{6.2.1}$$

In statistical terms, Ξ_S is the expectation value of the covariance matrix between eigenfunctions. As eigenfunctions are orthogonal with respect to the equilibrium distribution π, or in other words, statistically uncorrelated, Ξ_S is

just a diagonal matrix of the eigenvalues:

$$\mathbf{\Xi}_S \quad = \quad \mathbf{\Lambda}. \tag{6.2.2}$$

If we do not integrate over the full state space S in Eq. (6.2.1), but only over a subset $A \subset S$, we can define a matrix $\mathbf{\Xi}_A$ of size $M \times M$, called the *set-observable operator* for set A. All set-observable operators and two vectors $\boldsymbol{\omega}$, $\boldsymbol{\sigma} \in \mathbb{R}^M$ are the key ingredients of OOM theory. The vectors $\boldsymbol{\omega}$, $\boldsymbol{\sigma}$ equal the first canonical unit vector $\mathbf{e}_1$, i.e. $\boldsymbol{\omega} = \boldsymbol{\sigma} = \mathbf{e}_1 = (1, 0, \ldots, 0)^T$, and they are called *information state* and *evaluator*, respectively. If the finite-rank assumption Eq. (6.1.1) holds, these components form an algebraic system that allows to compute equilibrium probabilities of finite observation sequences. Let $A_1, \ldots, A_l$ be arbitrary subsets of S that do not need to form a partition of the state space. If Eq. (6.1.1) is satisfied, we can compute the probability that a trajectory in equilibrium visits its set A_1 at time τ, set A_2 at time 2τ, ..., and set A_l at time $l\tau$ by the following matrix-vector product:

$$\mathbb{P}(X_\tau \in A_1, X_{2\tau} \in A_2, \ldots, X_{l\tau} \in A_l) \quad = \quad \boldsymbol{\omega}^T \mathbf{\Xi}_{A_1} \ldots \mathbf{\Xi}_{A_l} \boldsymbol{\sigma}. \tag{6.2.3}$$

The proof can be found in Ref. [40], we also repeat it in appendix C.3. Note that, in case that $A_1, \ldots, A_l$ form a partition of state space, the probability of such an observation sequence cannot be computed from a Markov model transition matrix between the sets $A_1, \ldots, A_l$, unless the dynamics is Markovian on these sets. This clearly distinguishes an OOM from a Markov model: An OOM can correctly describe arbitrary projected dynamics as long as Eq. (6.1.1) holds.

As a Markov process is determined entirely by finite observation probabilities like Eq. (6.2.3), it follows that we can compute several key equilibrium, kinetic and mechanistic quantities in an unbiased fashion if we can somehow estimate the OOM components. For a fixed decomposition of state space into sets S_r, $r = 1, \ldots, N$ as before, let us denote the set-observable operators of sets S_r by $\mathbf{\Xi}_r$, which implies that

$$\mathbf{\Xi}_S \quad = \quad \sum_{r=1}^{N} \mathbf{\Xi}_r. \tag{6.2.4}$$

It follows from Eq. (6.2.3) that we can compute the unbiased equilibrium correlation matrix and the stationary probabilities by the formulas

$$\mathbf{C}_{Eq}^{\tau}(i,j) \quad = \quad \boldsymbol{\omega}^T \mathbf{\Xi}_i \mathbf{\Xi}_j \boldsymbol{\sigma}, \tag{6.2.5}$$

$$\pi_i \quad = \quad \boldsymbol{\omega}^T \mathbf{\Xi}_i \boldsymbol{\sigma}. \tag{6.2.6}$$

In practice we cannot directly estimate $\mathbf{\Xi}_r$ but only a *similar* operator $\hat{\mathbf{\Xi}}_r$. However, it follows directly from Eqs. (6.2.5-6.2.6) that if an unknown similarity transform $\mathbf{R} \in \mathbb{R}^{M \times M}$ affects all OOM quantities via

$$\hat{\mathbf{\Xi}}_r \quad = \quad \mathbf{R} \mathbf{\Xi}_r \mathbf{R}^{-1}, \tag{6.2.7}$$

$$\hat{\boldsymbol{\omega}}^T \quad = \quad \boldsymbol{\omega}^T \mathbf{R}^{-1}, \tag{6.2.8}$$

$$\hat{\boldsymbol{\sigma}} \quad = \quad \mathbf{R} \boldsymbol{\sigma}, \tag{6.2.9}$$

then Eqs. (6.2.5-6.2.6) remain exactly valid using $\hat{\omega}$, $\hat{\Xi}_r$, $\hat{\sigma}$. In other words, all OOMs that can be constructed by choosing some transformation matrix $\mathbf{R}$ form a family of equivalent OOMs. A specific member of this family *can* be estimated directly from simulation data, and thus we can use it in order to obtain unbiased estimates of Eqs. (6.2.5-6.2.6) even from a large ensemble of trajectories that do not need to sample from global equilibrium. It has been shown in Ref. [41] that Eqs. (6.2.14-6.2.15) and (6.2.16-6.2.17) in the next subsection indeed provide the components of an equivalent OOM, i.e. there is an invertible matrix $\mathbf{R}$ s.t. Eqs. (6.2.7-6.2.9) are satisfied in the absence of statistical noise.

6.2.2 Unbiased Estimation of Markov State Models

To construct an *exact* unbiased estimator we need three ingredients: (i) the expectation values of the empirical count matrix $\mathbf{C}_\rho^\tau$, (ii) the vector of total counts $\mathbf{c}_\rho$ from Eqs. (6.1.17-6.1.18), and additionally (iii) the two-step count matrices

$$\mathbf{C}_{\rho,r}^{2\tau} \quad := \quad \mathbb{E}\left(\mathbf{S}_r^{2\tau}\right). \tag{6.2.10}$$

As a reminder, expectation values here denote the expectation over a trajectory ensemble sampling from the empirical (non-equilibrium) distribution ρ. In practice, only finitely many simulations are available, and we thus replace $\mathbf{c}_\rho$, $\mathbf{C}_\rho^\tau$ and $\mathbf{C}_{\rho,r}^{2\tau}$ by count vectors and matrices $\overline{\mathbf{s}}$, $\overline{\mathbf{S}}^\tau$ and $\overline{\mathbf{S}}_r^{2\tau}$ (Eqs. (6.1.13-6.1.15)), which are asymptotically unbiased estimators. The unbiased estimation algorithm can be summarized as follows:

1. Obtain the empirical mean $\overline{\mathbf{s}}$, count matrix $\overline{\mathbf{S}}^\tau$ and two-step count matrices $\overline{\mathbf{S}}_r^{2\tau}$ from simulation data using Eqs. (6.1.13-6.1.15).

2. Decompose the count matrix $\overline{\mathbf{S}}^\tau$ by singular value decomposition (SVD)

   $$\overline{\mathbf{S}}^\tau \quad = \quad \mathbf{V}\Sigma\mathbf{W}^T, \tag{6.2.11}$$

   and compute weighted projections onto the leading M left and right singular vectors by

   $$\mathbf{F}_1 \quad = \quad \mathbf{V}_M\Sigma_M^{-1/2}, \tag{6.2.12}$$
   $$\mathbf{F}_2 \quad = \quad \mathbf{W}_M\Sigma_M^{-1/2}. \tag{6.2.13}$$

   We have used the symbols $\mathbf{V}_M$, $\mathbf{W}_M$, Σ_M to denote the restriction of these matrices to their first M columns.

3. Use $\mathbf{F}_1$, $\mathbf{F}_2$ to obtain the set-observable operators $\hat{\Xi}_r$ and the evaluation state vector $\hat{\sigma}$ of an equivalent OOM via

   $$\hat{\Xi}_r \quad = \quad \mathbf{F}_1^T\overline{\mathbf{S}}_r^{2\tau}\mathbf{F}_2, \tag{6.2.14}$$
   $$\hat{\sigma} \quad = \quad \mathbf{F}_1^T\overline{\mathbf{s}}. \tag{6.2.15}$$

   Compute the full-state observable operator $\hat{\Xi}_S = \sum_{r=1}^N \hat{\Xi}_r$ and obtain the information state vector $\hat{\omega}$ as the solution to the eigenvalue problem:

   $$\hat{\omega}^T\hat{\Xi}_S \quad = \quad \hat{\omega}^T, \tag{6.2.16}$$

$$\hat{\omega}^T\hat{\sigma} = 1. \tag{6.2.17}$$

The normalization Eq. (6.2.17) can be achieved by dividing the arbitrarily scaled solution $\hat{\omega}^T$ by $\hat{\omega}^T\hat{\sigma}$.

4. Compute the unbiased equilibrium correlation matrix and unbiased equilibrium distribution by

$$\mathbf{C}^{\tau}_{Eq}(i,j) = \hat{\omega}^T\hat{\Xi}_i\hat{\Xi}_j\hat{\sigma}, \tag{6.2.18}$$

$$\pi_i = \hat{\omega}^T\hat{\Xi}_i\hat{\sigma} \tag{6.2.19}$$

$$= \sum_{j=1}^{N}\mathbf{C}^{\tau}_{Eq}(i,j). \tag{6.2.20}$$

and then obtain the unbiased MSM transition matrix $\mathbf{T}^{\tau}_{Eq}$ either using the nonreversible estimator

$$\mathbf{T}^{\tau}_{Eq}(i,j) = \frac{\mathbf{C}^{\tau}_{Eq}(i,j)}{\pi_i}, \tag{6.2.21}$$

or the reversible estimator

$$\mathbf{T}^{\tau}_{Eq}(i,j) = \frac{\mathbf{C}^{\tau}_{Eq}(i,j) + \mathbf{C}^{\tau}_{Eq}(j,i)}{\sum_{j=1}^{N}\mathbf{C}^{\tau}_{Eq}(i,j) + \sum_{j=1}^{N}\mathbf{C}^{\tau}_{Eq}(j,i)}. \tag{6.2.22}$$

Let us briefly comment on the central idea behind this algorithm, which is the estimation of an equivalent OOM in the third step, particularly in Eq. (6.2.14). Using the path probability formula Eq. (6.2.3), it can be shown that the expected two-step count matrix is given by

$$\mathbf{C}^{2\tau}_{\rho,r} = \mathbf{Q}_{\rho}\Xi_r\mathbf{\Lambda}(\tau)\mathbf{Q}^T_{\pi}, \tag{6.2.23}$$

where the matrices $\mathbf{Q}_{\rho}$, $\mathbf{Q}_{\pi}$ are the same as in Eqs. (6.1.24-6.1.25). Thus, by the intermediate step, the set-observable operator is introduced into the decomposition of the two-step count matrix. Now, the idea is to find two matrices $\mathbf{F}_1$, $\mathbf{F}_2 \in \mathbb{R}^{N\times M}$, such that $\mathbf{R}_1 := \mathbf{F}_1^T\mathbf{Q}_{\rho}$ and $\mathbf{R}_2 := \mathbf{\Lambda}(\tau)\mathbf{Q}^T_{\pi}\mathbf{F}_2$ are inverse to each other, because this implies that

$$\mathbf{F}_1^T\mathbf{C}^{2\tau}_{\rho,r}\mathbf{F}_2 = \mathbf{R}_1\Xi_r\mathbf{R}_2 \tag{6.2.24}$$

$$= \mathbf{R}\Xi_r\mathbf{R}^{-1} \tag{6.2.25}$$

is the r-th component of an equivalent OOM. The properties of SVD and the decomposition Eq. (6.1.23) guarantee that the choice of $\mathbf{F}_1$, $\mathbf{F}_2$ in the second step above achieves this goal:

$$\mathbf{Id} = \mathbf{F}_1^T\mathbf{C}^{\tau}_{\rho}\mathbf{F}_2 \tag{6.2.26}$$

$$= \left(\mathbf{F}_1^T\mathbf{Q}_{\rho}\right)\left(\mathbf{\Lambda}(\tau)\mathbf{Q}^T_{\pi}\mathbf{F}_2\right) \tag{6.2.27}$$

$$= \mathbf{R}_1\mathbf{R}_2. \tag{6.2.28}$$

Similar arguments can be used to justify the equations for ω, σ. We also note that different choices of $\mathbf{F}_1$, $\mathbf{F}_2$ in step 2 are possible. For detailed explanations and proofs, please refer to the previous publications [39, 40, 41].

6.2.3 Recovery of Exact Relaxation Timescales

A remarkable by-product of the procedure described above is that the transformed full-state two-step count matrix $\hat{\Xi}_S$ is similar to a diagonal matrix of the system eigenvalues $\lambda_m(\tau)$ *without any MSM projection error.* This has been shown for equilibrium data in Ref. [70] and also applies to non-equilibrium data [40]:

$$\hat{\Xi}_S = \mathbf{R}\Xi_S\mathbf{R}^{-1} \quad (6.2.29)$$
$$= \mathbf{R}\mathbf{\Lambda}(\tau)\mathbf{R}^{-1}. \quad (6.2.30)$$

Thus, diagonalization of $\hat{\Xi}_S$ provides an estimate of the leading system eigenvalues, and consequently also of the relaxation rates or timescales, that is not distorted by the fact that we coarse-grain the dynamics to a Markov chain between coarse sets in state space. These eigenvalue and timescale estimates are only subject to statistical error, but not to any MSM model error. It is impossible to directly build an MSM that produces these timescales - when an MSM is desired, the timescales can only be approximated, and they will only be correct in the limit of long lag times and good discretization.

However, the fact that we can get a model-free estimate of the eigenvalues and relaxation timescales can be used to assess the discretization quality: According to Thm. 3.1, the exact system eigenvalues provide an upper bound to the eigenvalues of the equilibrium transition matrix $\mathbf{T}_{Eq}^{\tau}$. By comparing the eigenvalues of $\mathbf{T}_{Eq}^{\tau}$ to those from Eqs. (6.2.29-6.2.30), the MSM discretization error theoretically studied in [12, 9, 52] can be practically quantified.

6.2.4 Selection of Model Rank

The above method is theoretically guaranteed to work whenever the number of MSM states N is at least equal to the number M of relaxation processes in Eq. (6.1.1), and the count matrix $\mathbf{C}_{\rho}^{\tau}$ is of rank M. In the absence of statistical noise, the model rank M can then be determined by the number of non-zero singular values of $\mathbf{C}_{\rho}^{\tau}$. For finite data, the numerical rank of $\overline{\mathbf{S}}^{\tau}$ is not necessarily equal to M, as the singular values can be perturbed by noise. Classical matrix perturbation theory predicts that small singular values will be particularly affected by noise, see, e.g., Ref. [71], and also Fig. 6.4 A. Including noisy and small singular values can severely affect the accuracy of the method, most likely due to the presence of the matrix of inverse singular values in Eqs. (6.2.12-6.2.13). Also, we expect small singular values to have little impact on the dominant spectral and stationary properties of the final OOM, but this will be backed up by further theoretical investigation.

Consequently, it seems appropriate to cut off small and statistically unreliable singular values and select a smaller model rank $\hat{M} < M$ in Eqs. (6.2.12-6.2.13). In order to determine the uncertainties of the singular values, we use the bootstrapping procedure, and we discard all singular values with a signal-to-noise ratio of less than 10. This has proven to be a useful choice in all applications presented further below. Figure 6.4 B illustrates this procedure for a simple model system.

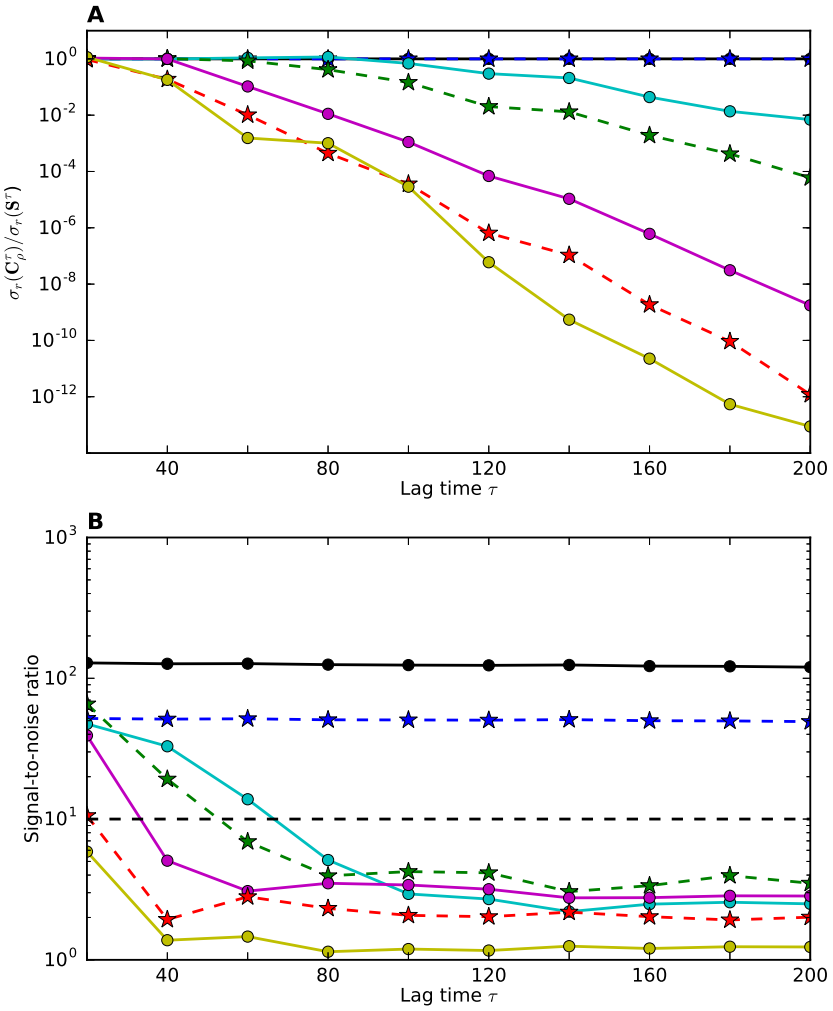

Figure 6.4: Analysis of statistical uncertainties for singular values of the count matrix. We use the one-dimensional model system and seven state discretization as in Sec. 6.3.1, the sample consists of $Q = 5000$ trajectories of length $K = 2000$. A: For each of the seven singular values (distinguished in descending order by the colors black, blue, cyan, green, magenta, red and yellow), we show the ratio of the true singular value $\sigma_r(\mathbf{C}_\rho^\tau)$, $r = 1, \ldots, 7$ of the expected count matrix $\mathbf{C}_\rho^\tau$ to the corresponding singular value $\sigma_r(\overline{\mathbf{S}}^\tau)$ of the empirical count matrix $\overline{\mathbf{S}}^\tau$, as a function of the lag time. As the small singular values decay quickly with the lag time, they are dominated by the noise even for small lag times. Including these noisy singular values would ruin the results. B: Ratio between mean value and uncertainty (signal-to-noise ratio) from the bootstrapping for the seven singular values as a function of the lag time. The thin black dashed line indicates the cut-off we have used in applications. Only singular values above this line are included in the estimation, the number of points above this line corresponds to the OOM model rank, see Fig. 6.5 H. This figure has been re-used with permission from Ref. [65], Nüske *et al.*, J. Chem. Phys. (2017, in press) [Fig. 4]. Copyright 2017 AIP Publishing.

6.2.5 Algorithmic Details, and Analysis of Computational Effort

We close this section by pointing out a few more details of practical importance. First, while it was convenient for the theoretical analysis to assume that all trajectories sample the same number of simulation steps K, this is not required (see appendix C.4). Moreover, we also argue in appendix C.4 that all normalizations in Eqs. (6.1.2-6.1.4) and (6.1.13-6.1.15) can be dropped in practice. All of the matrices $\overline{\mathbf{s}}$, $\overline{\mathbf{S}}^{\tau}$, $\overline{\mathbf{S}}_r^{2\tau}$ used in the estimation algorithm can be replaced by integer valued matrices that simply count the number of visits, transitions and two-step transitions.

Secondly, we have suggested to use the bootstrapping procedure in order to estimate uncertainties for the singular values of the count matrix. One way to realize this is to re-draw trajectories with replacement from the set of all available simulations, and to re-estimate the count matrix from this modified set of simulations. As individual simulations are statistically independent, this procedure is theoretically justified and can also be used to estimate uncertainties of further derived quantities, like timescales and stationary probabilities. We used the trajectory-based bootstrapping in all examples shown below. However, if only a small number of rather long simulations is available, it may be more practical to re-draw individual transitions from the set of all available transitions in the data set. Let T denote the total number of data points, which equals $T = KQ$ for uniform trajectory length, and Eq. (C.4.1) otherwise. If the transitions were statistically independent, one could simply re-sample T transition pairs from the set of all N^2 possible pairs, where the probability of drawing the pair (i, j) is given by $\overline{\mathbf{S}}^{\tau}(i, j)$. In fact, transitions are not statistically independent. Therefore, we suggest to replace the count matrix $\overline{\mathbf{S}}^{\tau}$ by the effective count matrix described in [72], but it should be noted that this procedure relies on several approximations and must be improved in the future.

Thirdly, we present an overview of the computational cost of each step in the estimation algorithm in Table 6.1 below, assuming that dense matrix algebra is used in every step. It is expressed in terms of the total number of data points T, the number of MSM states N, the OOM model rank M, and the number of bootstrapping samples n_b.

Operation	Cost
Count Matrix Estimation	$\propto T$
Bootstrapping	$\propto n_b T N^3$
SVD of $\overline{\mathbf{S}}^{\tau}$	$\propto N^3$
Computation of OOM components	$\hat{\boldsymbol{\sigma}}:\ MN + N^2$
	$\hat{\boldsymbol{\Xi}}:\ N\left(N^2M + NM^2\right)$
	$\hat{\boldsymbol{\omega}}:\propto M^3 + NM^2$
Transition Matrix $\mathbf{T}_{Eq}^{\tau}$	$N\left(2M^2 + M\right)$

Table 6.1: Analysis of computational effort required by the OOM-based estimation algorithm, if all operations are performed in dense matrix algebra.

The first step requires an effort which is linear in the data size and can be performed efficiently. In most cases, we can also assume the count matrices

$\overline{\mathbf{S}}^{\tau}$, $\overline{\mathbf{S}}_r^{2\tau}$ to be sparse, and the model rank M to be small. In this case, the cubic term appearing for the calculation of $\hat{\Xi}$ becomes quadratic, while the contributions of the model rank are small. The only real bottleneck is the singular value decomposition of $\overline{\mathbf{S}}^{\tau}$, accounting for the factor N^3 in the second and third step. As we generally require all singular values of the count matrix, this step must be performed using dense matrix algebra, which can be time-consuming. Future research may provide a method that only requires the computation of the leading singular values, thus allowing for sparse algebra to be employed.

6.3 Examples

For each of the following examples, we use the trajectory-based bootstrapping strategy to determine the OOM model rank. Mean values and standard errors for the singular values are estimated from $n_b = 10000$ re-samplings, singular values with a signal-to-noise ratio of at least 10.0 are accepted. We also generate error estimates for all quantities derived from the OOM-based Markov model by trajectory bootstrapping, using 1000 re-samplings. In addition, we compute a conventional Markov model without OOM-based correction as a comparison.

6.3.1 One-dimensional Toy Potential

As a first example, we study in more detail the one-dimensional system used in the introduction. The system is defined by the double-well potential function shown in Fig. 6.5 A. The dynamics here is a finite state space Markov chain with 100 microstates distributed along the x-axis, where transitions can occur between neighboring states based on a Metropolis criterion. The system is kinetically two-state, as the slowest relaxation timescale of the system, corresponding to the transition process between the two wells, is $t_2 = 3708$ steps and clearly dominates all others (Fig. 6.5B).

We investigate the estimation of a seven state Markov model ($N = 7$) using the discretization indicated by dashed lines in Fig. 6.5 A. Using seven states instead of two accelerates the convergence of OOM estimates. Still, the seven state discretization is a poor one - note that state 4 contains large parts of the transition region as well as parts of the right minimum. This choice was made deliberately in order to test the robustness of our method with respect to poor MSM clusterings. We produced two different data sets, each comprising $Q = 5000$ simulations. The first set contains short simulations of length $K = 250$, while the simulations of the second set are $K = 2000$ steps long. For the analysis of the smaller data set, we can use lag times up to $\tau = 30$, while we can go to up to $\tau = 200$ for the larger data set. Panels C, E, G of Fig. 6.5 display the results for the short simulations, while the corresponding results for the larger data set are shown in panels D, F, H. All simulations were initiated from a non-equilibrium starting distribution, where the probabilities to start in each of the seven states are given by the vector

$$\rho_1 = \begin{bmatrix} 0.3 & 0.3 & 0.3 & 0 & 0.05 & 0.05 & 0 \end{bmatrix}, \tag{6.3.1}$$

that is, 90 per cent of the simulations were started in the left three states, while

only 10 per cent were initialized in the deeper minimum on the right. Within each state, the actual microstate was selected from a uniform distribution.

Fig. 6.5C, D compare estimates of stationary probabilities from direct MSMs based on Eq. (6.1.16) and corrected MSMs with transition matrix given by Eq. (6.2.22). Due to the non-equilibrium initial distribution, the simulations visit the left minimum much more frequently than a simulation in equilibrium would do. While the MSM estimates of the stationary distribution converge to the true equilibrium distribution at long lag times, they are surprisingly inaccurate at short times, where the effect of the non-equilibrium starting distribution still has a strong effect. Even at the largest lag time $\tau = 200$, the bias is still visible. In contrast, the corrected MSM provides an excellent and stable estimate at lag times of 15 steps or longer.

In Fig. 6.5E, F, we compare estimates of the slowest implied relaxation timescale t_2 from three different estimators: A direct Markov model based on Eq. (6.1.16), the corrected Markov model based on Eq. (6.2.22), and the OOM-based spectral estimation Eqs. (6.2.29-6.2.30). First, we notice that the direct and corrected MSMs provide different estimates because of the combination of non-equilibrium starting points and the poor discretization quality. The corrected MSM timescales converge faster to the true timescales than the uncorrected ones. Second, the OOM-based direct estimation of relaxation timescales by Eq. (6.2.30) provides accurate results already at lag time $\tau = 15$, which is a regime where the number of relevant relaxation processes cannot be easily determined by a timescale separation, see again panel B of Fig. 6.5. The OOM timescale estimates become very accurate for larger lag times if more data can be used. Third, the large deviation between the corrected MSM and the OOM timescales are indicative of the poor discretization quality employed here.

Finally, in Fig. 6.5G, H we show the model rank selected by the bootstrapping procedure as a function of the lag time. We can observe how our criterion based on statistical uncertainties helps to select an appropriate model rank for each lag time, even when it is not obvious from the timescale plot. As expected, the system becomes effectively of rank 2 for lag times $\tau \geq 80$.

6.3.2 Molecular Dynamics Simulations of Alanine Dipeptide

Our second example is, again, molecular dynamics simulation data of alanine dipeptide (Ac-A-NHMe) in explicit water. Figure 6.6A shows the equilibrium probability distribution in the space of dihedral angles ϕ, ψ with its three metastable minima in the upper left, central left and central right part of the plane. The slow dynamics consists of exchanges between the left and right part ($t_2 \approx 1400\,\mathrm{ps}$) and between the two minima on the left ($t_3 \approx 70\,\mathrm{ps}$). We study the estimation of a Markov model using the discretization also indicated in panel A of Fig. 6.6. It was generated by kmeans clustering of the data set described below using $N = 40$ clustercenters. We produced an ensemble of roughly 11000 very short simulations of length 20 ps each. Simulations were initiated from eight different starting structures labelled by the numbers 1-8 in Fig. 6.6B, see appendix A.2 for details. It can be seen that the resulting empirical distribution does not even reach local equilibrium within the three metastable regions.

Like in the previous example, we find that it is possible to obtain precise estimates of stationary probabilities as soon as convergence of the OOM-based

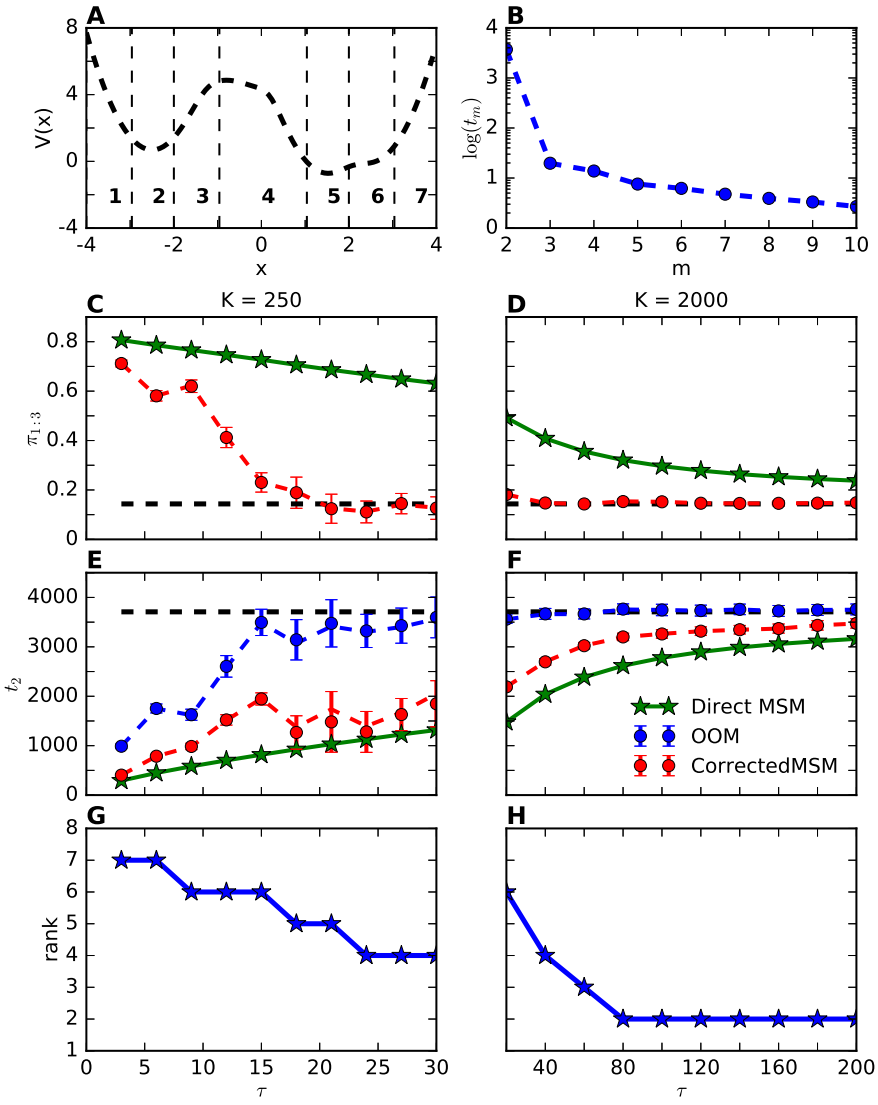

Figure 6.5: A) One-dimensional potential function and discretization of the landscape into seven states. B) Decadic logarithm of the first nine implied timescales of the model system. C, D) Estimates of the stationary probability of states 1-3 from the direct MSM (green) and the corrected MSM (red), compared to the reference (black dashed line). E, F) Estimates of the slowest relaxation timescale t_2 from a direct MSM (green), the corrected MSM (red) and the OOM-based spectral estimation (blue), compared to the reference (black dashed line). G, H) Model rank selected by the bootstrapping procedure. For all quantities derived from the OOM, the dashed lines indicate the estimated values using the complete data set, whereas the bullets and errorbars correspond to mean and standard error from the bootstrapping procedure. Note that errorbars are hardly visible in panels D and F. This figure has been re-used with permission from Ref. [65], Nüske *et al.*, J. Chem. Phys. (2017, in press) [Fig. 5]. Copyright 2017 AIP Publishing.

timescales is achieved. In panel C of Fig. 6.6, we compare results for the equilibrium probability of all states in the right part of the plane, from a direct MSM and the corrected MSM. For lag times $\tau \geq 500$ fs, we are able to correct the bias introduced by strong non-equilibrium sampling.

In panels D and F of Fig. 6.6, we present estimates of the two slowest timescales t_2, t_3 produced by the same estimators as before (OOM in blue, direct MSM in green and corrected MSM in red). Additionally, the cyan lines correspond to the timescale estimates of an MSM using equilibrium simulations and the same discretization (see appendix A.2). We find that the OOM-based spectral estimation provides accurate timescale estimates for short lag times starting at $\tau = 500$ fs. Moreover, we notice that for lag times as small as these, MSM timescales are clearly lower than the true timescales, although a decent discretization is employed. The difference between OOM and MSM estimates indicates that an even finer discretization would be required to match the references at these lag times. The direct estimates, the reference equilibrium timescales, and our OOM-based estimates of equilibrium timescales, are nearly identical. Only the mean values extracted from bootstrapping for t_2 seem to be a bit low. This will be investigated further.

Finally, the selected model ranks shown in Fig. 6.6E confirm that our framework can work in situations where low-rank descriptions of the dynamics using only a few processes are not adequate.

6.3.3 Two-dimensional model system with poor discretization

Our final example is another finite state space Markov chain in the two-dimensional energy landscape shown in Fig. 6.7A, defined by 40×40 microstates. Here we show the behavior of different estimators in an extreme case, where the discretization is so poor that MSM estimates fail completely. Transitions between neighboring states are now possible in both x- and y-direction, again based on a Metropolis criterion. We study the estimation of a Markov model using a discretization into 16 MSM states, also shown in Fig. 6.7A. As can be seen in Fig. 6.7B, there are two dominant timescales, $t_2 \approx 144000$ steps and $t_3 \approx 17000$ steps. The next timescale is clearly separated from the first two, after that, there is no more apparent timescale separation. This time, we fix the simulation length at $K = 5000$ steps, i.e. the trajectories are approximately 30 times shorter than the slowest timescale. The simulations are started from a uniform distribution over all microstates. In panels C-H of Fig. 6.7, we display the results if the number of simulations is set to $Q = 2000$ (C, E, G) and $Q = 10000$ (D, F, H).

In Fig. 6.7C, D,, we show the estimation results for the equilibrium probability of the states labeled 13, 14 and 15. We expect it to be difficult to estimate this probability, as the states are blending different metastable regions and transition regions. It can be observed that the estimation of stationary probabilities is more sensitive to noise, see the results for $Q = 2000$. This observation is not surprising, as the stationary probabilities require accurate estimation of the two-step count matrices Eq. (6.1.4) from the data, which can be more difficult for rarely visited states. Still, for $Q = 10000$, a reliable estimate is achieved and the biased estimate of the direct MSM can be corrected. Another comparison we make is between the estimates from the corrected MSM and those from long equilibrium simulations that use the same number of total data points,

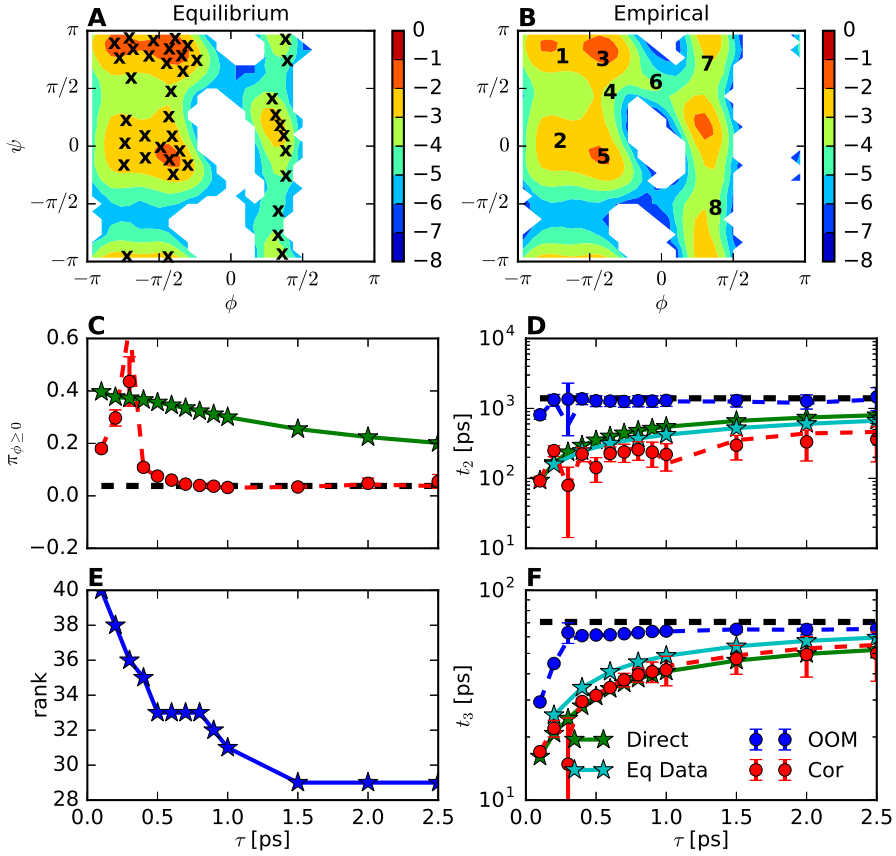

Figure 6.6: Results for alanine dipeptide. A) Equilibrium distribution (logarithmic scale) in the space of backbone dihedral angles ϕ, ψ and clustercenters of a fourty state kmeans discretization used to analyze the data. B) Empirical distribution (logarithmic scale) sampled by the data initiated from eight starting structures indicated by the numbers 1-8. C) Equilibrium probability of all states in the right part of the plane estimated from the direct MSM (green) and the corrected MSM (red). Reference in black. D) Estimates for the slowest relaxation timescale t_2 from a direct MSM (green), the corrected MSM (red) and the OOM-based estimation (blue). Reference values from equilibrium simulations are displayed in black. We also show the expected timescale estimate using the same fourty state discretization if equilibrium data was used (cyan line). E) Model rank used for the OOM estimation as determined by the bootstrapping. F) The same as D) for the second slowest timescales t_3. For all quantities derived from the OOM, the dashed lines indicate the estimated values using the complete data set, whereas the bullets and errorbars correspond to mean and standard error from the bootstrapping procedure. Note that errorbars are hardly visible in panels C and F. This figure has been re-used with permission from Ref. [65], Nüske *et al.*, J. Chem. Phys. (2017, in press) [Fig. 6]. Copyright 2017 AIP Publishing.

i.e. $K = 2000 \cdot 5000 = 10^7$ for $Q = 2000$ and $K = 10000 \cdot 5000 = 5 \cdot 10^7$ for $Q = 10000$. We show mean values and standard errors from roughly 900 long simulations for $Q = 2000$, and roughly 400 simulations for $Q = 10000$. In both cases, the estimates from long equilibrium trajectories provide more accurate estimates. In practice, however, one needs to strike a balance between long trajectories that are more beneficial for the analysis, and short trajectories that can be more efficient for sampling and state exploration [73, 74, 75].

Again, we also compare the estimates for the slowest timescales t_2 (E-F) and t_3 (G-H) from a direct MSM, the corrected MSM and the OOM-based spectral estimation. In both cases, correct estimates of both timescales can be obtained from the OOM, while both the direct and corrected MSMs estimate timescales one order of magnitude too small. This suggests that for a bad enough discretization, correcting for the effect of the non-equilibrium starting distribution will not be sufficient to achieve convergence in the timescales. However, the poor discretization quality is revealed by a large error between the OOM-based estimate and the corrected MSM, and this observation can be exploited in order to improve the discretization and repeat the analysis.

6.4 Conclusions

We have investigated the quality of Markov state models when estimated from many simulations of short length, initiated from non-equilibrium starting conditions. We have derived an expression for the error between unbiased MSM transition probabilities and the expected estimate from many short simulations. This error is shown to depend on the simulation length, the lag time and the state discretization. If ultra-long trajectories are employed, i.e. trajectories that are long compared to the slowest relaxation timescales, then the effect of the initial distribution is negligible and no further correction is needed. For ensembles of short trajectories, the situation is more complex. Preparing simulation trajectories in such a way that they emerge from a local equilibrium distribution does not appear to be of much practical use: this would only correct the first transition count of every trajectory while the subsequent trajectory segments are still biased. The local equilibrium will be lost for intermediate times along the trajectory as the trajectory ensemble is not in global equilibrium. In a similar sense discarding initial simulation fragments can reduce the bias, but cannot systematically remove it. In particular, since the effect of the bias disappears with the slowest relaxation times of the system, discarding pieces of simulation trajectories appears more harmful in terms of reducing the statistics than it is useful to reduce the bias. With the standard MSM estimator, the most effective and simplest method to reduce the bias from the initial trajectory distribution in fact seems to be using a longer lag time or a better state space discretization. These are already the usual objectives of MSM construction. However, if the discretization is poor, the estimation bias due to a non-equilibrium distribution can be dramatic at practically usable lag times.

The main result of this chapter is an improved estimator of the MSM transition matrix which is not biased by the initial distribution. This new estimator is based on theory of observable operator models. In contrast to the standard MSM estimator, the corrected MSM estimator does not only use the number of transitions observed between pairs of states at lag time τ, but also the number

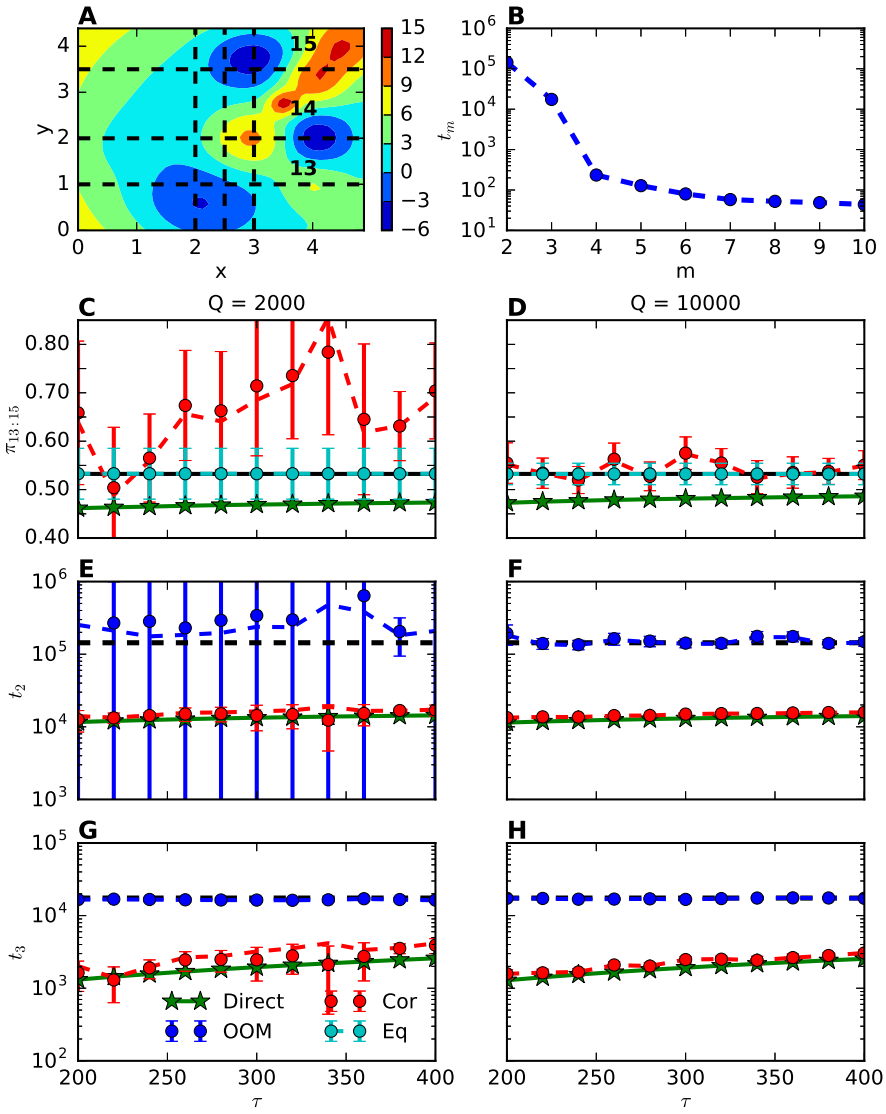

Figure 6.7: A) Two-dimensional potential function with discretization into 16 MSM states indicated by dashed lines. B) Leading nine implied timescales t_m of the system. C, D) Estimates of equilibrium probability of states 13, 14 and 15 from direct MSM (green) and the corrected MSM (red), compared to the reference (black line) and estimates from 900 (C) / 400 (D) different equilibrium simulations, shown by the cyan lines. E, F) Estimates of slowest relaxation timescale t_2 from a direct MSM (green), the corrected MSM (red) and the OOM-based spectral estimation (blue), compared to the reference (black dashed line). G, H) The same for t_3. For all quantities derived from the OOM, the dashed lines indicate the estimated values using the complete data set, whereas the bullets and errorbars correspond to mean and standard error from the bootstrapping procedure. Note that errorbars are hardly visible in panels F and H. This figure has been re-used with permission from Ref. [65], Nüske *et al.*, J. Chem. Phys. (2017, in press) [Fig. 7]. Copyright 2017 AIP Publishing.

of transitions at lag time 2τ. These statistics are combined to get a transition matrix estimate at lag time τ that is unbiased by the initial trajectory distribution. While it may seem that having to estimate statistics at 2τ is a deficiency compared to standard MSM estimation when only short simulation trajectories are available, please note that the corrected MSM estimator can get significantly better estimates at short lag times, so in practice the lag times needed for a converged MSM will be smaller than for the standard estimator.

Finally, we report a result from the OOM framework that shows how the model-free relaxation timescales can be computed from the same statistics used for the corrected MSM estimator (i.e. transition matrices at lag times τ and 2τ). These estimates are only impaired by statistical error, but are not affected by systematic MSM error as no MSM is used in the process of obtaining them. The difference between the corrected MSM timescales and the OOM timescales can be used in order to assess the discretization quality, as this difference goes to zero in the limit of good discretization.

6.5 Outlook: OOM Estimation for General Basis Sets

We close this study by pointing out how the OOM-based estimation can be extended to the case of arbitrary basis functions. The first step is to replace the indicator functions χ_i, χ_j in Eqs. (6.1.2-6.1.4) by the general basis functions f_i, f_j. The second modification is that we need to compute a two-step correlation matrix for every intermediate data point in the simulations. Let (y_1, y_2, y_3) denote triples of two subsequent transitions over lag time τ in the data, and call their total number T. Then, for every intermediate data point x_2, we compute

$$\overline{\mathbf{S}}^{2\tau}_{x_2}(i,j) \quad = \quad \frac{1}{T} \sum_{(y_1,y_2,y_3)} f_i(y_1)\delta_{x_2}(y_2)f_j(y_3). \tag{6.5.1}$$

The estimation of $\mathbf{F}_1$, $\mathbf{F}_2$, σ, ω is similar to what was done before, and we estimate a set-observable operator for every x_2 by $\hat{\Xi}_{x_2} = \mathbf{F}_1^T \overline{\mathbf{S}}^{2\tau}_{x_2} \mathbf{F}_2$. Finally, we re-compute the instantaneous and time-lagged correlation matrix over all the data, but the data points are re-weighted. In the computation of $\mathbf{C}^0$, every data point x is weighted by $\hat{\omega}\hat{\Xi}_x\hat{\sigma}$, while for $\mathbf{C}^\tau$, every transition pair (x, x') is weighted by $\hat{\omega}\hat{\Xi}_x\hat{\Xi}_{x'}\hat{\sigma}$. It was shown in Ref. [41] that these estimators converge to the equilibrium correlation matrices $\mathbf{C}^0_{Eq}$, $\mathbf{C}^\tau_{Eq}$ in the limit of infinite sampling.

Chapter 7

Summary

In this thesis, we have presented the variational approach to conformational dynamics (VAC), a method to extract the essential information from simulations of high-dimensional stochastic dynamics. We have focussed on molecular dynamics simulations of biological macromolecules, but the methods presented are applicable to any reversible and ergodic Markov process, and probably even to more general stochastic processes. The basic idea is to approximate the dominant eigenfunctions and eigenvalues of the associated transfer operator from a pre-selected library of basis functions. The approximate eigenfunctions serve as a low-dimensional representation of the essential dynamics. We have explained that a generalized eigenvalue problem must be solved in order to obtain these approximations, and shown that the required matrices can be estimated from equilibrium simulation data. After presenting applications of the method to model systems, it was suggested to use tensor products of one-dimensional functions defined on elementary coordinates in order to model the dynamics of complex systems. In order to circumvent the resulting dimensionality problem, we have discussed the tensor-train-format as a suitable low-rank representation. An adapted learning algorithm was formulated and promising applications were presented. Finally, we have discussed the use of short non-equilibrium simulations in conjunction with the VAC, by focussing on the special case of Markov state models. We have derived an expression for the error between the MSM transition matrix in equilibrium and the expected transition matrix from non-equilibrium sampling. Subsequently, it was explained how the framework of observable operator models can be used to estimate the equilibrium transition matrix from short simulations. Algorithmic details were discussed and successful applications to model systems were presented. We have also outlined how the OOM-based estimation can be applied to the VAC using a general basis set.

Appendix A

Simulation Setups

A.1 Alanine Dipeptide (Long Simulations)

We performed all-atom molecular dynamics simulations of alanine dipeptide (Ac-Ala-NHMe), in explicit water using the GROMACS 4.5.5 [76] simulation package, the AMBER ff-99SB-ILDN force field [57], and the TIP3P water model [77]. The simulations were performed in the canonical ensemble at a temperature of 300 K. The energy-minimized starting structure of Ac-Ala-NHMe was solvated into a cubic box with a minimum distance between solvent and box wall of 1 nm, corresponding to a box volume of 2.72 nm^3 and 651 water molecules. After an initial equilibration of 100 ps, 20 production runs of 200 ns each were performed, yielding a total simulation time of 4 μs. Covalent bonds to hydrogen atoms were constrained using the LINCS algorithm [78] (lincs_iter = 1, lincs_order = 4), allowing for an integration time step of 2 fs. The leap-frog integrator was used. The temperature was maintained by the velocity-rescale thermostat [79] with a time constant of 0.01 ps. Lennard-Jones interactions were cut off at 1 nm. Electrostatic interactions were treated by the Particle-Mesh Ewald (PME) algorithm [80] with a real space cutoff of 1 nm, a grid spacing of 0.15 nm, and an interpolation order of 4. Periodic boundary conditions were applied in the x, y, and z-direction. The trajectory data was stored every 1 ps.

A.2 Alanine Dipeptide (Short Simulations)

Molecular dynamics simulations of alanine dipeptide in explicit water at temperature 300 K were generated with AceMD [81] software using the AMBER ff-99SB-ILDN force field [82] and an integration time step of 2 fs. The peptide was simulated inside a cubic box of volume $(2.7222\,\mathrm{nm})^3$ containing 651 TIP3P water molecules. The Langevin thermostat was used. The electrostatics were computed every two time steps by the particle-mesh Ewald (PME) method [83], using real-space cutoff 0.9 nm and grid spacing 0.1 nm. All bonds between hydrogens and heavy atoms were constrained.

We have produced 11388 ultra short simulations of length 20 ps each, with 50 fs saving interval. The simulations were initiated from eight different structures, their projections into $\phi - \psi$-space are indicated by the number 1-8 in

Fig. 6.6 B. The probabilities to start in each of these structures are given by the vector

$$\rho_1 = \begin{bmatrix} 0.05 & 0.05 & 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 \end{bmatrix}. \quad \text{(A.2.1)}$$

These simulations were used to perform the analyses described in Sec. 6.3.2. Using the same setup, we produced 2363 long runs of 1 ns simulation time each, with 1 ps saving interval. We estimated a Markov model on the 40-state kmeans discretization at lag time $\tau = 100$ ps using this data set, and extracted the reference timescales and equilibrium probabilities shown as black lines in Fig. 6.6. Also, we used the stationary probabilities estimated from this model to initialize 203 short equilibrium runs of 500 ps simulation time each, with 100 fs saving interval. This data set was used to compute the equilibrium timescales of the kmeans discretization shown as cyan lines in Fig. 6.6 D, F.

A.3 Deca Alanine

We performed all-atom molecular-dynamics simulations of deca alanine, which is protonated at the amino terminus and deprotonated at the carboxy terminus, using the GROMACS 4.5.5 simulation package, the Amber03 force field [84] and the TIP3P water model. A completely elongated conformation was chosen as an initial structure.
The structure was solvated in a cubic box of volume $V = 232.6\,\mathrm{nm}^3$, with 7647 pre-equilibrated TIP3P water molecules. First, an equilibration run of 500 ps in the NVT ensemble with full position restraints, using the velocity-rescale thermostat, was carried out. This was followed by a 500ps NPT equilibration run. The temperature was set to $T = 300$ K. The equilibration run was followed by a 500 ns production run, again at $T = 300$ K. Two temperature coupling groups were used with a velocity-rescale thermostat and a time constant of 0.01 ps. Periodic boundary conditions were applied in the x, y and z direction. For the long range electrostatic interaction PME was used with a PME-order of 4 and a Fourier grid spacing of 0.15 nm. Covalent bonds to hydrogen bonds were constrained using the LINCS algorithm, allowing for a 2 fs time step. The leap frog integrator was used. Data was saved every 1 ps, resulting in $5 \cdot 10^5$ data frames. Six independent simulations from the same equilibrated configuration were carried out resulting in 3 μs total data.

Appendix B

Optimization in Tensor-Train-Format

B.1 Relation to the Block-TT-Format

Our optimization method shown in Alg. 5.1 is built on the modification of the ALS (Ref. [34]) for the block-TT-format, see Refs. [60, 61]. The block-TT-format allows for the simultaneous parametrization of a number $M > 1$ functions using only a few additional parameters. A tensor is in block-p-format if there is exactly one component $\mathbf{U}_p$ which carries an additional index m, enumerating the different functions, while all remaining components retain their structure as before. Eqs. (5.2.2) and (5.2.6) then turn into

$$\begin{aligned} \hat{\psi}_{\mathbf{m}} &= \sum_{i_1,\ldots,i_d} \left[\sum_{k_1=1}^{r_1} \cdots \sum_{k_{d-1}=1}^{r_{d-1}} \mathbf{U}_1(i_1,k_1) \ldots \mathbf{U}_p(k_{p-1},i_p,k_p,\mathbf{m}) \ldots \right. \qquad \text{(B.1.1)} \\ &\quad \left. \mathbf{U}_d(k_{d-1},i_d) \right] f^1_{i_1}(x_1) \ldots f^d_{i_d}(x_d), \\ \hat{\psi}_{\mathbf{m}} &= \sum_{k_{p-1},i_p,k_p} \mathbf{U}_p(k_{p-1},i_p,k_p,\mathbf{m})\, f^p_{i_p}(x_p) \cdot \qquad \text{(B.1.2)} \\ &\quad g^p_{k_{p-1}}(x_1,\ldots,x_{p-1}) \cdot h^p_{k_p}(x_{p+1},\ldots,x_d), \end{aligned}$$

where we have highlighted the additional index in bold-face letters. The ALS-optimization of multiple eigenfunctions proceeds as follows: suppose we are on the forward sweep, the tensor is in block-p-format and we seek to update component $\mathbf{U}_p$ while all others are fixed. We observe that we can solve the eigenvalue problem Eq. (3.1.7) for the three-fold product basis in Eq. (B.1.2), and we can update every slice $\mathbf{U}_p(:,:,:,m)$ by the m-th eigenvector thus obtained. In order to proceed to the optimization of the next component $\mathbf{U}_{p+1}$, however, the index m needs to be moved into $\mathbf{U}_{p+1}$ first. Otherwise, $\mathbf{U}_p$ would parametrize different left interfaces for every value of m, which violates the idea of the TT-format. In the literature, it is suggested to perform the index move as follows.

- Re-shape the component $\mathbf{U}_p$ into a matrix in $\mathbb{R}^{r_{p-1}\cdot n \times r_p \cdot M}$ and compute a low-rank decomposition, e.g. by SVD or QR-decomposition:

$$\mathbf{U}_p(k_{p-1}, i_p, k_p, m) = \sum_{k'_p=1}^{r'_p} \mathbf{V}_p(k_{p-1}, i_p, k'_p)\mathbf{W}_p(k'_p, k_p, m). \quad \text{(B.1.3)}$$

- Contract the arrays $\mathbf{W}_p$ and $\mathbf{U}_{p+1}$ by summing over k_p:

$$\tilde{\mathbf{U}}_{p+1}(k'_p, i_{p+1}, k_{p+1}, m) = \sum_{k_p=1}^{r_p} \mathbf{W}_p(k'_p, k_p, m)\mathbf{U}_{p+1}(k_p, i_{p+1}, k_{p+1}). \quad \text{(B.1.4)}$$

After this, the p-th component can be updated by $\mathbf{V}_p$, which carries no more than three indices, while the $p+1$-st component can be updated by $\tilde{\mathbf{U}}_{p+1}$, which now enumerates the index m. Furthermore, the p-th rank has changed to r'_p, thus allowing for rank-adaptivity during the iteration. Also note that the decomposition Eq. (B.1.3) needs to be truncated, otherwise the ranks r_p can easily blow up.
Initially, we attempted to apply ALS using the above method, but the truncation step turned out to be problematic. The main obstacle was that decompositions like SVD do not respect the underlying structure of the problem, namely that the solutions $\hat{\psi}_m$ need to be orthogonal with respect to the weighted inner product Eq. (2.3.3). Even for large ranks r'_p, yielding close approximations to the full matrix $\mathbf{U}_p$, the resulting functions $\hat{\psi}_m$ often failed to fulfill the orthogonality constraints. Consequently, we were facing either intolerably large ranks, or meaningless results.
Still, the optimization algorithm described in this work produces a tensor in the block-TT-format. Recall that the optimization of component $\mathbf{U}_p$ provides a new left interface $g_{k_p}^{p+1}(\mathbf{U}_p)$. The eigenvectors of the generalized eigenvalue problem Eq. (3.1.7) parametrize M eigenfunctions in terms of the reduced basis Eq. (5.2.13), yielding a component $\mathbf{U}_{p+1} \in \mathbb{R}^{r_p \times n \times r_{p+1} \times M}$. Thus, the tensor is in block-$p+1$-format after the optimization. However, this component is not used, as it is updated immediately afterwards by the next optimization step.

B.2 Optimization Problem for the Components $\mathbf{U}_p$

Here, we formulate the optimization problem which needs to be solved for increasing ranks r_p in every iteration step of Alg. 5.1. We seek to determine the optimal component $\mathbf{U}_p \in \mathbb{R}^{r_{p-1} \times n \times r_p}$, s.t. the eigenvalue sum Eq. (5.2.11) for the reduced basis Eq. (5.2.13) is maximal. This is an unconstrained optimization problem which can be solved numerically by a conjugate gradient method if we can provide the derivatives of the eigenvalues $\hat{\lambda}_m(\mathbf{U}_p)$ w.r.t. the entries of $\mathbf{U}_p$. These derivatives can be obtained as follows: The eigenvalues $\hat{\lambda}_m(\mathbf{U}_p)$ solve the generalized eigenvalue problem Eq. (3.1.7) using the reduced correlation matrices $\mathbf{C}^\tau(\mathbf{U}_p)$, $\mathbf{C}^0(\mathbf{U}_p)$ between the basis functions Eq. (5.2.13).

These correlation matrices can be computed from the larger correlation matrices $\mathbf{C}^{\tau}_{p,p+1}$, $\mathbf{C}^{0}_{p,p+1}$ of the four-fold product basis Eq. (5.2.14):

$$\mathbf{C}^{\tau}_{p,p+1} = \langle \mathcal{T}_\tau \, g^{p-1}_{k_{p-1}} f^{p}_{i_p} f^{p+1}_{i_{p+1}} h^{p+1}_{k_{p+1}}, \, g^{p-1}_{l_{p-1}} f^{p}_{j_p} f^{p+1}_{j_{p+1}} h^{p+1}_{l_{p+1}} \rangle_\pi, \tag{B.2.1}$$

$$\mathbf{C}^{0}_{p,p+1} = \langle g^{p-1}_{k_{p-1}} f^{p}_{i_p} f^{p+1}_{i_{p+1}} h^{p+1}_{k_{p+1}}, \, g^{p-1}_{l_{p-1}} f^{p}_{j_p} f^{p+1}_{j_{p+1}} h^{p+1}_{l_{p+1}} \rangle_\pi, \tag{B.2.2}$$

by the formulas

$$\left[\mathbf{C}^{\tau}(\mathbf{U}_p)\right]^{k_p,i_{p+1},k_{p+1}}_{l_p,j_{p+1},l_{p+1}} = \sum_{k_{p-1},i_p,l_{p-1},j_p} \mathbf{U}_p(k_{p-1},i_p,k_p) \cdot \left[\mathbf{C}^{\tau}_{p,p+1}\right]^{k_{p-1},i_p,i_{p+1},k_{p+1}}_{l_{p-1},j_p,j_{p+1},l_{p+1}} \mathbf{U}_p(l_{p-1},j_p,l_p), \tag{B.2.3}$$

$$\left[\mathbf{C}^{0}(\mathbf{U}_p)\right]^{k_p,i_{p+1},k_{p+1}}_{l_p,j_{p+1},l_{p+1}} = \sum_{k_{p-1},i_p,l_{p-1},j_p} \mathbf{U}_p(k_{p-1},i_p,k_p) \cdot \left[\mathbf{C}^{0}_{p,p+1}\right]^{k_{p-1},i_p,i_{p+1},k_{p+1}}_{l_{p-1},j_p,j_{p+1},l_{p+1}} \mathbf{U}_p(l_{p-1},j_p,l_p). \tag{B.2.4}$$

Using these formulas, we can differentiate the matrix entries of $\mathbf{C}^{\tau}(\mathbf{U}_p)$ and $\mathbf{C}^{0}(\mathbf{U}_p)$ w.r.t. the variables $\mathbf{U}_p$:

$$\frac{\partial \left[\mathbf{C}^{\tau}(\mathbf{U}_p)\right]^{k_p,i_{p+1},k_{p+1}}_{l_p,j_{p+1},l_{p+1}}}{\partial \mathbf{U}_p(k'_{p-1},i'_p,k'_p)} = \sum_{l_{p-1},j_p} \left[\mathbf{C}^{\tau}_{p,p+1}\right]^{k'_{p-1},i'_p,i_{p+1},k_{p+1}}_{l_{p-1},j_p,j_{p+1},k_{p+1}} \mathbf{U}_p(l_{p-1},j_p,l_p)\delta_{k_p,k'_p} + \sum_{k_{p-1},i_p} \left[\mathbf{C}^{\tau}_{p,p+1}\right]^{k_{p-1},i_p,i_{p+1},k_{p+1}}_{k'_{p-1},i'_p,j_{p+1},k_{p+1}} \mathbf{U}_p(k_{p-1},i_p,k_p)\delta_{l_p,k'_p}, \tag{B.2.5}$$

$$\frac{\partial \left[\mathbf{C}^{0}(\mathbf{U}_p)\right]^{k_p,i_{p+1},k_{p+1}}_{l_p,j_{p+1},l_{p+1}}}{\partial \mathbf{U}_p(k'_{p-1},i'_p,k'_p)} = \sum_{l_{p-1},j_p} \left[\mathbf{C}^{0}_{p,p+1}\right]^{k'_{p-1},i'_p,i_{p+1},k_{p+1}}_{l_{p-1},j_p,j_{p+1},k_{p+1}} \mathbf{U}_p(l_{p-1},j_p,l_p)\delta_{k_p,k'_p} + \sum_{k_{p-1},i_p} \left[\mathbf{C}^{0}_{p,p+1}\right]^{k_{p-1},i_p,i_{p+1},k_{p+1}}_{k'_{p-1},i'_p,j_{p+1},k_{p+1}} \mathbf{U}_p(k_{p-1},i_p,k_p)\delta_{l_p,k'_p}. \tag{B.2.6}$$

What remains is to compute derivatives of the eigenvalues $\hat{\lambda}_m(\mathbf{U}_p)$ w.r.t. the matrix entries of $\mathbf{C}^{\tau}(\mathbf{U}_p)$, $\mathbf{C}^{0}(\mathbf{U}_p)$. For isolated eigenvalues $\hat{\lambda}_m(\mathbf{U}_p)$ and positive definite $\mathbf{C}^{0}(\mathbf{U}_p)$, matrix perturbation theory yields the results:

$$\frac{\partial \hat{\lambda}_m(\mathbf{U}_p)}{\partial \mathbf{C}^{\tau}(\mathbf{U}_p)(i,j)} = \mathbf{a}_m(i)\mathbf{a}_m(j)\left(2-\delta_{ij}\right) \tag{B.2.7}$$

$$\frac{\partial \hat{\lambda}_m(\mathbf{U}_p)}{\partial \mathbf{C}^{0}(\mathbf{U}_p)(i,j)} = -\hat{\lambda}_m(\mathbf{U}_p)\mathbf{a}_m(i)\mathbf{a}_m(j)\left(2-\delta_{ij}\right), \tag{B.2.8}$$

where $\mathbf{a}_m$ is the m-th eigenvector corresponding to $\hat{\lambda}_m(\mathbf{U}_p)$. Combining Eqs. (B.2.7-B.2.8) and (B.2.5-B.2.6), we find the derivatives of $\hat{\lambda}_m(\mathbf{U}_p)$ w.r.t. the variables $\mathbf{U}_p$. Equations (B.2.7-B.2.8) can be obtained from perturbation theory. Consider an analytic perturbation of $\mathbf{C}^{\tau} = \mathbf{C}^{\tau}(\mathbf{U}_p)$ and $\mathbf{C}^{0} = \mathbf{C}^{0}(\mathbf{U}_p)$:

$$\tilde{\mathbf{C}}^{\tau} = \mathbf{C}^{\tau} + \epsilon \mathbf{C}_1^{\tau} + \dots \quad \text{(B.2.9)}$$

$$\tilde{\mathbf{C}}^{0} = \mathbf{C}^{0} + \epsilon \mathbf{C}_1^{0} + \dots \quad \text{(B.2.10)}$$

Then, the proof of [85, Theorem 1] can be imitated for the positive definite generalized eigenvalue problem to show that also the eigenvalue $\tilde{\lambda}_m$ of $\tilde{\mathbf{C}}^{\tau}$, $\tilde{\mathbf{C}}^{0}$ can be computed by a series expansion in a small neighborhood of $\mathbf{C}^{\tau}$, $\mathbf{C}^{0}$:

$$\tilde{\lambda}_m = \hat{\lambda}_m(\mathbf{U}_p) + \epsilon \hat{\lambda}_m^1 + \dots \quad \text{(B.2.11)}$$

Moreover, the proof of this theorem also provides an expression for the first order correction $\hat{\lambda}_m^1$. For the positive definite generalized eigenvalue problem, the correction becomes

$$\hat{\lambda}_m^1 = (\mathbf{a}_m)^T \left(\mathbf{C}_1^{\tau} - \hat{\lambda}_m(\mathbf{U}_p) \mathbf{C}_1^{0} \right) \mathbf{a}_m. \quad \text{(B.2.12)}$$

Equations (B.2.7-B.2.8) now follow if we use the perturbations $\mathbf{C}_1^{\tau} = \mathbf{C}_1^{0} = \mathbf{E}^{ij}$, where $\mathbf{E}^{ij}$ is a matrix whose elements (i, j) and (j, i) are equal to one while all others are zero. Note that the factor $2 - \delta_{ij}$ accounts for the symmetry of the matrices $\mathbf{C}^{\tau}(\mathbf{U}_p)$, $\mathbf{C}^{0}(\mathbf{U}_p)$.

B.3 Least Squares Approximation of Interfaces

In order to evaluate the contribution of the one-coordinate basis $f_{i_p}^p$ to the full solution, we suggest the following simple method. As before, we explain the method in the context of the forward iteration. The interface functions $g_{k_p}^{p+1}$ encode the relevant information about coordinates $x_1, \dots, x_p$ into a limited number r_p of functions. If coordinate x_p was relevant for the slow dynamics, these interfaces should differ from the ones computed previously, i.e. from the functions $g_{k_{p-1}}^{p}$. Therefore, after the interfaces $g_{k_p}^{p+1}$ have been optimized, we approximate these functions in the least squares sense from the basis of previous interfaces $g_{k_{p-1}}^{p}$. The expansion coefficient vector $\mathbf{u}^{k_p}$ of the best approximation for the interface $g_{k_p}^{p+1}$,

$$f_{k_p} = \sum_{l_{p-1}} \mathbf{u}^{k_p}(l_{p-1}) g_{l_{p-1}}^{p}, \quad \text{(B.3.1)}$$

is found as the solution of the linear system

$$\mathbf{A}^p \mathbf{u}^{k_p} = \mathbf{b}^{k_p} \quad \text{(B.3.2)}$$

$$\mathbf{A}^p(k_{p-1}, l_{p-1}) = \langle g_{k_{p-1}}^{p}, g_{l_{p-1}}^{p} \rangle_{\pi} \quad \text{(B.3.3)}$$

$$\mathbf{b}^{k_p}(k_{p-1}) = \langle g_{k_{p-1}}^{p}, g_{k_p}^{p+1} \rangle_{\pi}. \quad \text{(B.3.4)}$$

These quantities can be obtained from the correlation matrix $\mathbf{C}_{p,p+1}^{0}$ in Eq. (B.2.2). The matrix $\mathbf{A}^p$ is just a submatrix of $\mathbf{C}_{p,p+1}^{0}$, whereas the vector $\mathbf{b}^{k_p}$ can be computed via

$$\mathbf{b}^{kp}(k_{p-1}) = \sum_{l_{p-1},j_p} \mathbf{U}_p(l_{p-1},j_p,k_p)\langle g^{p-1}_{k_{p-1}}, g^{p-1}_{l_{p-1}} f^p_{j_p}\rangle_\pi, \tag{B.3.5}$$

where we have used the recursion formula Eq. (5.2.8). Next, we can compute the approximation error for $g^{p+1}_{k_p}$ via

$$\begin{aligned}\left(E(p)_{k_p}\right)^2 &= \langle g^{p+1}_{k_p} - f_{k_p}, g^{p+1}_{k_p} - f_{k_p}\rangle_\pi && \text{(B.3.6)}\\ &= 1 - 2\langle g^{p+1}_{k_p}, f_{k_p}\rangle_\pi + \langle f_{k_p}, f_{k_p}\rangle_\pi && \text{(B.3.7)}\\ &= 1 - 2\left(\mathbf{b}^{k_p}\right)^T \mathbf{u}^{k_p} + \left(\mathbf{u}^{k_p}\right)^T \mathbf{A}^p \mathbf{u}^{k_p}. && \text{(B.3.8)}\end{aligned}$$

Finally, we compute the average approximation error $E(p) = \frac{1}{r_p}\sum_{k_p=1}^{r_p} E(p)_{k_p}$ and use it as a measure of the importance of coordinate x_p.

Appendix C

Proofs

C.1 Definition of Transfer Operator on all Lebesgue Spaces

Proposition C.1. *If μ is an invariant measure, then $\mathcal{T}_t$ is well-defined on all spaces L^p_μ, $1 \leq p \leq \infty$.*

Proof. As μ is a probability measure, we have that $L^p_\mu \subset L^1_\mu$ for all $1 < p \leq \infty$, thus $\mathcal{T}_t$ can be defined on all these spaces by restriction. Moreover, an application of Jensen's inequality shows that the Koopman operator can be extended to all spaces L^p_μ, $1 \leq p \leq \infty$ [86, Lem. 1]. We show that for $\frac{1}{p} + \frac{1}{q} = 1$, the adjoint operator of $\mathcal{K}_t$, as an operator on L^q_μ, is the restriction of $\mathcal{T}_t$ to L^p_μ, as this implies that $\mathcal{T}_t$ is not only defined on L^p_μ, but also maps functions from that space back into L^p_μ. To this end, denote the extension of $\mathcal{K}_t$ to L^q_μ by $\mathcal{K}^q_t$. If we choose $g = \chi_A \in L^\infty_\mu \subset L^q_\mu$, we find for all $f \in L^p_\mu$:

$$\begin{aligned}
\int_A \left(\mathcal{K}^q_t\right)^* f(x)\mu(\mathrm{dx}) &= \langle \left(\mathcal{K}^q_t\right)^* f, g\rangle_\mu && \text{(C.1.1)}\\
&= \langle f, \mathcal{K}^q_t g\rangle_\mu && \text{(C.1.2)}\\
&= \langle f, \mathcal{K}_t g\rangle_\mu && \text{(C.1.3)}\\
&= \langle \mathcal{T}_t f, g\rangle_\mu && \text{(C.1.4)}\\
&= \int_A \mathcal{T}_t f(x)\mu(\mathrm{dx}). && \text{(C.1.5)}
\end{aligned}$$

As $A \in \mathfrak{S}$ was arbitrary, we have that $\left(\mathcal{K}^q_t\right)^* f = \mathcal{T}_t f$ μ-a.e. in S. Therefore, $\mathcal{T}_t = \left(\mathcal{K}^q_t\right)^*$. □

C.2 Ergodic Behaviour of Time-Lagged Correlations

The proof of Theorem 3.4 relies on the Birkhoff ergodic theorem [43, Thm. 4.2.4]. For simplicity, we assume $L = 1$, all of the following arguments work out equally well for $L > 1$. We briefly write $(\Omega, \Sigma, \mathbb{P})$ instead of $\Omega_{\Delta t}$, $\Sigma_{\Delta t}$, $\mathbb{P}_{\Delta t}$

and $\omega = (X_k)_{k=0}^{\infty}$ instead of $\omega = (X_{k\Delta t})_{k=0}^{\infty}$ for a discrete trajectory ω. On the space Ω of discrete trajectories, we define the transformation $G : \Omega \to \Omega$ that shifts a trajectory ω one step forward:

$$G[\omega] \quad = \quad (X_{k+1})_{k=0}^{\infty}, \tag{C.2.1}$$

and call this map the *shift*. As $\mathbb{P}$ is generated by the unique invariant measure μ, it follows from the Kolmogorov-Daniell Theorem that $\mathbb{P}$ is invariant under the shift, that is

$$\mathbb{P}(G^{-1}(A)) \quad = \quad \mathbb{P}(A) \tag{C.2.2}$$

for all $A \in \Sigma$. Moreover, it can be shown that $\mathbb{P}$ is *ergodic* with respect to the shift, that is, any set $A \in \Sigma$ that is unchanged under the shift, i.e.

$$G^{-1}(A) \quad = \quad A, \tag{C.2.3}$$

is either of $\mathbb{P}$-measure zero or one. The details can be found in the lecture notes [87, sec. 5], but the key point is to show that if $\mathbb{P}$ is not ergodic, the state space S can be decomposed into two dynamically disconnected sets. As a result, the invariant measure can be expressed as a convex combination of two measures that are also invariant, contradicting the uniqueness of the stationary measure. Then, because of the shift invariance and ergodicity of the measure $\mathbb{P}$, the Birkhoff ergodic theorem implies that for the following functions, defined on a discrete trajectory ω:

$$F_{ij}^0(\omega) \quad = \quad f_i(X_0) f_j(X_0), \tag{C.2.4}$$

$$F_{ij}^1(\omega) \quad = \quad f_i(X_0) f_j(X_1), \tag{C.2.5}$$

we find:

$$\frac{1}{K-1} \sum_{k=0}^{K-2} f_i(X_k) f_j(X_k) \quad = \quad \frac{1}{K-1} \sum_{k=0}^{K-2} F_{ij}^0(G^k(\omega)) \tag{C.2.6}$$

$$\to \quad \int_{\Omega} F_{ij}^0(\omega) \, \mathrm{d}\mathbb{P}(\omega) \tag{C.2.7}$$

$$= \quad \int_S f_i(x) f_j(x) \, \mu(\mathrm{dx}). \tag{C.2.8}$$

$$\frac{1}{K-1} \sum_{k=0}^{K-2} f_i(X_k) f_j(X_{k+1}) \quad = \quad \frac{1}{K-1} \sum_{k=0}^{K-2} F_{ij}^1(G^k(\omega)) \tag{C.2.9}$$

$$\to \quad \int_{\Omega} F_{ij}^1(\omega) \, \mathrm{d}\mathbb{P}(\omega) \tag{C.2.10}$$

$$= \quad \int_S \int_S f_i(x) f_j(y) \, \mu(\mathrm{dx}) p_1(x, \mathrm{dy}). \tag{C.2.11}$$

C.3 OOM Probability of Observation Sequence

Here, we show the derivation of the path probability formula Eq. (6.2.3), that can also be found in Ref. [40]. In general, the left-hand side of Eq. (6.2.3) can be expressed by repeated integrals over the transition kernel:

$$\begin{aligned}\mathbb{P}(X_\tau \in A_1, \ldots, X_{l\tau} \in A_l) &= \int_\Omega \int_{A_1} \ldots \int_{A_l} \mathrm{dx}_0 \ldots \mathrm{dx}_l \, \pi(x_0) p(x_0, x_1; \tau) \ldots \\ &\quad p(x_{l-1}, x_l; \tau). \end{aligned} \tag{C.3.1}$$

Note that π appears in the first integral as we assumed that the dynamics is in equilibrium, i.e. the initial distribution equals π. Next, we replace all transition kernels by the expansion in Eq. (6.1.1):

$$\begin{aligned}\mathbb{P}(X_\tau \in A_1, \ldots, X_{l\tau} \in A_l) &= \sum_{m_0=1}^{M} \sum_{m_1=1}^{M} \ldots \sum_{m_{l-1}=1}^{M} \left[\int_\Omega \mathrm{dx}_0 \, \pi(x_0) \psi_{m_0}(x_0) \right] \\ &\quad \lambda_{m_0}(\tau) \left[\int_{A_1} \mathrm{dx}_1 \, \psi_{m_0}(x_1) \pi(x_1) \psi_{m_1}(x_1) \right] \ldots \\ &\quad \lambda_{m_{l-1}}(\tau) \left[\int_{A_l} \mathrm{dx}_l \, \psi_{m_{l-1}}(x_l) \pi(x_l) \right] \qquad \text{(C.3.2)} \\ &= \sum_{m_0=1}^{M} \sum_{m_1=1}^{M} \ldots \sum_{m_{l-1}=1}^{M} \delta_{1,m_0} \Xi_{A_1}(m_0, m_1) \ldots \\ &\quad \Xi_{A_l}(m_{l-1}, 1). \qquad \text{(C.3.3)} \end{aligned}$$

In the second equation, we have used the π-orthogonality of the eigenfunctions ψ_{m_0} and the fact that $\psi_1 \equiv 1$ in order to replace the x_0-integral by δ_{1,m_0}. For the last integral, we have also used that $\psi_1 \equiv 1$. This is a sequence of matrix-vector products. It remains to use $\delta_{1,m_0} = \boldsymbol{\omega}(m_0)$ and that $\Xi_{A_l}(m_{l-1}, 1) = \left[\Xi_{A_l} \boldsymbol{\sigma}\right](m_{l-1})$. In matrix notation, Eq. (6.2.3) follows:

$$\mathbb{P}(X_\tau \in A_1, \ldots, X_{l\tau} \in A_l) = \boldsymbol{\omega}^T \Xi_{A_1} \ldots \Xi_{A_l} \boldsymbol{\sigma}. \tag{C.3.4}$$

Finally, note that this derivation also works if the dynamics is not in equilibrium. In this case, the vector $\boldsymbol{\omega}$ is given by $\boldsymbol{\omega}(m_0) = \int_\Omega \mathrm{dx}_0 \, \rho_0(x_0) \psi_{m_0}(x_0)$, where ρ_0 is the non-equilibrium initial condition.

C.4 Variable Simulation Length

Here, we verify that the estimation algorithm from Sec. 6.2.2 can be applied to data sets comprised of simulations of non-uniform length. We assume that for $j = 1, \ldots, J$, there is an ensemble of Q_j simulations of length $K_j + 2\tau$, i.e. K_j transition pairs / triples will be used from each of these trajectories. We assume that $Q_j \to \infty$ for all j, s.t. every sub-ensemble samples from an empirical distribution ρ_j. Define the number of data points generated by the j-th ensemble as $T_j = Q_j K_j$, and the total number of data points by

$$T := \sum_{j=1}^{J} Q_j K_j. \tag{C.4.1}$$

Moreover, we assume that $\frac{T_j}{T} \to \alpha_j$, i.e. the fraction of data points generated by the j-th ensemble approaches a constant for all j. Let us define the distribution

$$\rho = \sum_{j=1}^{J} \alpha_j \rho_j. \tag{C.4.2}$$

Trajectories of length $K_j + 2\tau$ are enumerated by q_j and labelled $\mathbf{Y}_{q_j}$. Further, let $s_{K_j}(\mathbf{Y}_{q_j})$ be any of the estimators from Eqs. (6.1.2-6.1.4), where the subscript K_j indicates that $K - 2\tau$ in Eqs. (6.1.2-6.1.4) must be replaced by K_j. In addition, denote by $s(\mathbf{Y}_{q_j})$ the same estimator, but without the normalization. Also, let c_{ρ_j} denote the corresponding correlation from Eqs. (6.1.17-6.1.18) and (6.2.10) w.r.t. the density ρ_j. It follows that

$$\begin{aligned}
\overline{s}_T &:= \frac{1}{T}\left[\sum_{q_1=1}^{Q_1} s(\mathbf{Y}_{q_1}) + \ldots + \sum_{q_J=1}^{Q_J} s(\mathbf{Y}_{q_J})\right] && \text{(C.4.3)}\\
&= \frac{T_1}{T}\left[\frac{1}{T_1}\sum_{q_1=1}^{Q_1} s(\mathbf{Y}_{q_1})\right] + \ldots + \frac{T_J}{T}\left[\frac{1}{T_J}\sum_{q_J=1}^{Q_J} s(\mathbf{Y}_{q_J})\right] && \text{(C.4.4)}\\
&= \frac{T_1}{T}\left[\frac{1}{Q_1}\sum_{q_1=1}^{Q_1} s_{K_1}(\mathbf{Y}_{q_1})\right] + \ldots + \frac{T_J}{T}\left[\frac{1}{Q_J}\sum_{q_J=1}^{Q_J} s_{K_J}(\mathbf{Y}_{q_J})\right] && \text{(C.4.5)}\\
&\to \alpha_1 \mathbb{E}\left(s_{K_1}\right) + \ldots + \alpha_J \mathbb{E}\left(s_{K_J}\right) && \text{(C.4.6)}\\
&= \alpha_1 c_{\rho_1} + \ldots + \alpha_J c_{\rho_J} && \text{(C.4.7)}\\
&= c_\rho. && \text{(C.4.8)}
\end{aligned}$$

The convergence in Eq. (C.4.6) is convergence in probability. Thus, if we sum up all visits / transitions / two-step transitions, and divide by the total number of data points in the end, we arrive at an asymptotically correct estimator of the correlations w.r.t. the density ρ. As the OOM estimation algorithm only relies on consistent estimators for correlations w.r.t. some empirical density ρ, it can still be applied in this setting. Finally, the normalization by $\frac{1}{T}$ can be omitted in practice, because it cancels out in Eqs. (6.2.14-6.2.19).

Bibliography

[1] M.P. Allen and D.J. Tildesley. *Computer Simulation of Liquids*. Oxford Science Publ. Clarendon Press, 1989. (Cited on p.1).

[2] W. F. van Gunsteren, J. Dolenc, and A. E. Mark. Molecular simulation as an aid to experimentalists. *Current Opinion in Structural Biology*, 18(2):149 – 153, 2008. (Cited on p.1).

[3] G. A. Pavliotis. *Stochastic Processes and Applications*. Texts in Applied Mathematics. Springer New York, 2014. (Cited on p.1).

[4] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard. A Direct Approach to Conformational Dynamics based on Hybrid Monte Carlo. *Journal of Computational Physics*, 151:146–168, 1999. (Cited on pp.1, 11, and 43).

[5] E. B. Davies. Metastable States of Symmetric Markov Semigroups I. *Proceedings of the London Mathematical Society*, s3-45(1):133–150, 1982. (Cited on pp.2 and 11).

[6] E. B. Davies. Metastable States of Symmetric Markov Semigroups II. *Journal of the London Mathematical Society*, s2-26(3):541–556, 1982. (Cited on pp.2 and 11).

[7] M. Dellnitz and O. Junge. On the Approximation of Complicated Dynamical Behavior. *SIAM Journal on Numerical Analysis*, 36(2):491–515, 1999. (Cited on pp.2 and 11).

[8] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications*, 315(1):39 – 59, 2000. (Cited on pp.2 and 11).

[9] 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. (Cited on pp.2, 17, 18, 43, 49, 50, 51, and 57).

[10] C. Schütte and M. Sarich. *Metastability and Markov State Models in Molecular Dynamics: Modeling, Analysis, Algorithmic Approaches*, volume 24 of *Courant Lecture Notes*. American Mathematical Society and Courant Institute of Mathematical Sciences, 2013. (Cited on pp.2 and 17).

[11] G. R. Bowman, V. S. Pande, and F. Noé, editors. *An Introduction to Markov State Models and Their Application to Long Timescale Molecular Simulation*. Springer Netherlands, 2014. (Cited on pp.2 and 17).

[12] M. Sarich, F. Noé, and C. Schütte. On the Approximation Quality of Markov State Models. *SIAM Multiscale Modeling and Simulation*, 8:1154–1177, 2010. (Cited on pp.2, 3, 51, and 57).

[13] S.M. Ulam. *A Collection of Mathematical Problems*. Interscience tracts in pure and applied mathemtics, no.8. Interscience Publ., 1960. (Cited on p.2).

[14] G. R. Bowman, K. A. Beauchamp, G. Boxer, and V. S. Pande. Progress and Challenges in the Automated Construction of Markov State Models for Full Protein Systems. *Journal of Chemical Physics*, 131:124101, 2009. (Cited on pp.2 and 49).

[15] B. Trendelkamp-Schroer, H. Wu, F. Paul, and F. Noé. Estimation and Uncertainty of Reversible Markov Models. *The Journal of Chemical Physics*, 143(17), 2015. (Cited on pp.2 and 49).

[16] P. Deuflhard and M. Weber. Robust Perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161 – 184, 2005. (Cited on p.2).

[17] 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. (Cited on p.2).

[18] P. Metzner, C. Schütte, and E. Vanden-Eijnden. Transition Path Theory for Markov Jump Processes. *Multiscale Modeling & Simulation*, 7(3):1192–1219, 2009. (Cited on p.2).

[19] N. Singhal Hinrichs and V. S. Pande. Calculation of the distribution of eigenvalues and eigenvectors in Markovian state models for molecular dynamics. *The Journal of Chemical Physics*, 126(24), 2007. (Cited on p.2).

[20] F. Noé. Probability Distributions of Molecular Observables Computed from Markov Models. *J. Chem. Phys.*, 128(24):244103, 2008. (Cited on p.2).

[21] F. Noé, S. Doose, I. Daidone, M. Löllmann, M. Sauer, J. D. Chodera, and J. C. Smith. Dynamical fingerprints for probing individual relaxation processes in biomolecular dynamics with simulations and kinetic experiments. *Proceedings of the National Academy of Sciences*, 108(12):4822–4827, 2011. (Cited on p.2).

[22] J.-H. Prinz, B. Keller, and F. Noe. Probing molecular kinetics with Markov models: metastable states, transition pathways and spectroscopic observables. *Phys. Chem. Chem. Phys.*, 13:16912–16927, 2011. (Cited on p.2).

[23] 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. (Cited on pp.2, 3, and 43).

[24] F. Noé, I. Horenko, C. Schütte, and J. C. Smith. Hierarchical Analysis of Conformational Dynamics in Biomolecules: Transition Networks of Metastable States. *Journal of Chemical Physics*, 126:155102, 2007. (Cited on p.2).

[25] N.V. Buchete and G. Hummer. Coarse Master Equations for Peptide Folding Dynamics. *J. Phys. Chem. B*, 112:6057–6069, 2008. (Cited on p.2).

[26] V. A. Voelz, G. R. Bowman, K. Beauchamp, and V. S. Pande. Molecular Simulation of ab Initio Protein Folding for a Millisecond Folder NTL9. *Journal of the American Chemical Society*, 132(5):1526–1528, 2010. (Cited on p.2).

[27] W. Zhuang, R. Z. Cui, D.-A. Silva, and X. Huang. Simulating the T-Jump-Triggered Unfolding Dynamics of trpzip2 Peptide and Its Time-Resolved IR and Two-Dimensional IR Signals Using the Markov State Model Approach. *Journal of Physical Chemistry B*, 115:5415–5424, 2011. (Cited on p.2).

[28] L. Molgedey and H. G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72:3634–3637, 1994. (Cited on pp.2 and 17).

[29] G. Perez-Hernandez, F. Paul, T. Giorgino, G. de Fabritiis, and F. Noé. Identification of Slow Molecular Order Parameters for Markov Model Construction. *Journal of Chemical Physics*, 139:015102, 2013. (Cited on pp.2 and 17).

[30] C. R. Schwantes and V. S. Pande. Modeling Molecular Kinetics with tICA and the Kernel Trick. *Journal of Chemical Theory and Computation*, 11:600–608, 2015. (Cited on pp.2 and 17).

[31] W. Huisinga and B. Schmidt. *Metastability and Dominant Eigenvalues of Transfer Operators*, pages 167–182. Springer Berlin Heidelberg, 2006. (Cited on pp.3, 10, 11, and 12).

[32] I. Oseledets and E. Tyrtyshnikov. Breaking the Curse of Dimensionality, Or How to Use SVD in Many Dimensions. *SIAM Journal on Scientific Computing*, 31:3744–3759, 2009. (Cited on pp.3 and 31).

[33] I. Oseledets. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing*, 33:2295–2317, 2011. (Cited on pp.3, 31, and 32).

[34] 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:A683–A713, 2012. (Cited on pp.3, 34, 35, and 71).

[35] S. Röblitz. *Statistical Error Estimation and Grid-free Hierarchical Refinement in Conformation Dynamics*. PhD thesis, Freie Universität Berlin, 2009. (Cited on pp.3, 43, 48, and 52).

[36] M. Weber. A Subspace Approach to Molecular Markov State Models via an Infinitesimal Generator. Habilitation thesis, ZIB, 2009. (Cited on pp.3 and 43).

[37] X. Huang, G. R. Bowman, S. Bacallado, and V. S. Pande. Rapid equilibrium sampling initiated from nonequilibrium data. *Proceedings of the National Academy of Sciences*, 106(47):19765–19769, 2009. (Cited on pp.3 and 43).

[38] G. R Bowman, X. Huang, and V. S. Pande. Using generalized ensemble simulations and Markov state models to identify conformational states. *Methods*, 49(2):197–201, 2009. (Cited on pp.3 and 43).

[39] H. Jaeger. Observable Operator Models for Discrete Stochastic Time Series. *Neural Comput.*, 12(6):1371–1398, 2000. (Cited on pp.4, 44, 52, 53, and 56).

[40] H. Wu, J.-H. Prinz, and F. Noé. Projected Metastable Markov Processes and their Estimation with Observable Operator Models. *The Journal of Chemical Physics*, 143(14), 2015. (Cited on pp.4, 44, 52, 53, 54, 56, 57, and 78).

[41] H. Wu and F. Noé. Spectral Learning of Dynamic Systems from Nonequilibrium Data. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Adv. Neural Inf. Process. Syst. 29*, pages 4179–4187. Curran Associates, Inc., 2016. (Cited on pp.4, 44, 52, 53, 55, 56, and 67).

[42] B. Heinz. *Probability Theory*. De Gruyter CY - Berlin, Boston, 2011. (Cited on pp.6 and 7).

[43] A. Lasota and M. C. Mackey. *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics*. Springer, 1993. (Cited on pp.7, 8, and 76).

[44] A. Nielsen. *Computation Schemes for Transfer Operators*. PhD thesis, Freie Universität Berlin, 2016. (Cited on p.9).

[45] W. Huisinga. *Metastability of Markovian systems - A transfer operator based approach in application to molecular dynamics*. PhD thesis, Freie Universität Berlin, 2003. (Cited on pp.9 and 10).

[46] F. Noé and F. Nüske. A Variational Approach to Modeling Slow Processes in Stochastic Dynamical Systems. *SIAM Multiscale Modeling and Simulation*, 11:635–655, 2013. (Cited on p.12).

[47] C. Bandle. *Isoperimetric inequalities and applications*. Monographs and studies in mathematics. Pitman, 1980. (Cited on p.12).

[48] D. Werner. *Funktionalanalysis*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2007. (Cited on p.12).

[49] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry*. Dover Publications, Mineola, NY, 1982. (Cited on pp.13 and 14).

[50] 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. (Cited on p.16).

[51] Amnon Pazy. *Semigroups of linear operators and applications to partial differential equations*. Applied mathematical sciences. Springer, New York, Berlin, 1983. (Cited on p.17).

[52] N. Djurdjevac, M. Sarich, and C. Schütte. Estimating the Eigenvalue Error of Markov State Models. *Multiscale Modeling & Simulation*, 10(1):61–81, 2012. (Cited on pp.18, 45, and 57).

[53] H. Wu, F. Nüske, F. Paul, S. Klus, P Koltai, and F. Noé. Variational approximation of molecular kinetics from short off-equilibrium simulations. *J. Chem. Phys.*, (in revision), 2017. (Cited on p.19).

[54] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *J. Nonlinear Sci.*, 25(6):1307–1346, 2015. (Cited on p.19).

[55] 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*, 10:1739–1752, 2014. (Cited on p.20).

[56] M. Senne, B. Trendelkamp-Schroer, A. S. J. S. Mey, C. Schütte, and F. Noé. EMMA : A Software Package for Markov Model Building and Analysis. *J. Chem. Theory Comput.*, 8:2223–2238, 2012. (Cited on p.21).

[57] K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw. Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins*, 78(8):1950–1958, 2010. (Cited on pp.21 and 69).

[58] 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. (Cited on p.30).

[59] F. Vitalini, A. S. J. S. Mey, F. Noé, and B. G. Keller. Dynamic properties of force fields. *The Journal of Chemical Physics*, 142(8), 2015. (Cited on p.31).

[60] 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*, 185:1207 – 1216, 2014. (Cited on pp.34 and 71).

[61] D. Kressner, M. Steinlechner, and A. Uschmajew. Low-Rank Tensor Methods with Subspace Correction for Symmetric Eigenvalue Problems. *SIAM Journal on Scientific Computing*, 36:A2346–A2368, 2014. (Cited on pp.34 and 71).

[62] D.E. Shaw, P. Maragakis, K. Lindorff-Larsen, S. Piana, R.O. Dror, M.P. Eastwood, J.A. Bank, J.M. Jumper, J.K. Salmon, Y. Shan, and W. Wriggers. Atomic-Level Characterization of the Structural Dynamics of Proteins. *Science*, 330:341–346, 2010. (Cited on pp.40 and 51).

[63] 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. (Cited on pp.40 and 52).

[64] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé. PyEMMA 2: A Software Package for Estimation, Validation, and Analysis of Markov Models. *Journal of Chemical Theory and Computation*, 2015. (Cited on p.40).

[65] F. Nüske, H. Wu, J.-H. Prinz, C. Wehmeyer, C. Clementi, and F. Noé. Markov State Models from short non-Equilibrium Simulations - Analysis and Correction of Estimation Bias. *The Journal of Chemical Physics*, 2017, in press. (Cited on pp.43, 46, 53, 58, 62, 64, and 66).

[66] J. Chodera, V. S. Pande, M. Weber, F. Noé, and C. Schütte. Personal Communication. (Cited on p.43).

[67] Marcus Weber. *Meshless Methods in Conformation Dynamics*. PhD thesis, Freie Universität Berlin, 2006. (Cited on p.48).

[68] K. Lindorff-Larsen, S. Piana, R. O. Dror, and D. E. Shaw. How Fast-Folding Proteins Fold. *Science*, 334(6055):517–520, 2011. (Cited on p.51).

[69] W. C. Swope, J. W. Pitera, and F. Suits. Describing Protein Folding Kinetics by Molecular Dynamics Simulations. 1. Theory. *The Journal of Physical Chemistry B*, 108(21):6571–6581, 2004. (Cited on p.51).

[70] J.-H. Prinz. *Advanced Estimation Methods for Markov Models of Dynamical Systems*. PhD thesis, Freie Universität Berlin, 2012. (Cited on p.57).

[71] G. W. Stewart. Perturbation Theory for the Singular Value Decomposition. In R. J. Vaccaro, editor, *SVD and Signal Processing, II: Algorithms, Analysis and Applications*. Elsevier, 1991. (Cited on p.57).

[72] F. Noé. Statistical inefficiency of Markov model count matrices. 2015. (Cited on p.59).

[73] J. Preto and C. Clementi. Fast Recovery of Free Energy Landscapes via Diffusion-Map-directed Molecular Dynamics. *Phys. Chem. Chem. Phys.*, 16:19181–19191, 2014. (Cited on p.65).

[74] S. Doerr and G. De Fabritiis. On-the-Fly Learning and Sampling of Ligand Binding by High-Throughput Molecular Simulations. *Journal of Chemical Theory and Computation*, 10(5):2064–2069, 2014. (Cited on p.65).

[75] S. Doerr, M. J. Harvey, F. Noé, and G. De Fabritiis. HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. *Journal of Chemical Theory and Computation*, 12(4):1845–1852, 2016. (Cited on p.65).

[76] D. van der Spoel, E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen. GROMACS: Fast, Flexible and Free. *Journal of Computational Chemistry*, 26:1701–1718, 2005. (Cited on p.69).

[77] J. A. Kritzer, J. Tirado-Rives, S. A. Hart, J. D. Lear, W. L. Jorgensen, and A. Schepartz. Relationship between side chain structure and 14-helix stability of beta3-peptides in water. *J. Am. Chem. Soc.*, 127(1):167–178, 2005. (Cited on p.69).

[78] B. Hess, H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije. LINCS: A linear constraint solver for molecular simulations. *J. Comput. Chem.*, 18(12):1463–1472, 1997. (Cited on p.69).

[79] G. Bussi, D. Donadio, and M. Parrinello. Canonical Sampling through Velocity Rescaling. *Journal of Chemical Physics*, 126:014101, 2007. (Cited on p.69).

[80] T. Darden, D. York, and L. Pedersen. Particle Mesh Ewald: An N-log(N) Method for Ewald Sums in Large Systems. *Journal of Chemical Physics*, 98:10089–10092, 1993. (Cited on p.69).

[81] M. J. Harvey, G. Giupponi, and G. De Fabritiis. ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale. *Journal of Chemical Theory and Computation*, 5(6):1632–1639, 2009. (Cited on p.69).

[82] K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw. Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins*, 78(8):1950–1958, 2010. (Cited on p.69).

[83] T. Darden, D. York, and L. Pedersen. Particle mesh Ewald : An N -log (N) method for Ewald sums in large systems. *J. Chem. Phys*, 98:10089–10092, 1993. (Cited on p.69).

[84] Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, T. Lee, J. Caldwell, J. Wang, and P. Kollman. A Point-Charge Force Field for Molecular Mechanics Simulations of Proteins based on Condensed-Phase Quantum Mechanical Calculations. *Journal of Computational Chemistry*, 24:1999–2012, 2003. (Cited on p.70).

[85] J. R. Magnus. On Differentiating Eigenvalues and Eigenvectors. *Econometric Theory*, 1:179–191, 1985. (Cited on p.74).

[86] J.R. Baxter and J. S. Rosenthal. Rates of convergence for everywhere-positive Markov chains. *Statistics & Probability Letters*, 22(4):333 – 338, 1995. (Cited on p.76).

[87] M. Hairer. Ergodic theory for Stochastic PDEs. Lecture Notes, 2008. (Cited on p.77).

Acknowledgments

It is my pleasure to thank all the people who have helped me with this research project. First of all, I would like to thank Frank Noé for offering me to work on this project, for being an exceptionally good and supportive supervisor, and for creating the exciting atmosphere that is characteristic of the CMB group. I am very grateful to Reinhold Schneider and Cecilia Clementi for the ongoing collaboration and all the help I have received from them over the last years. I would like to thank Antonia Mey, Guillermo Pérez-Hernández, Bettina Keller, Fabian Paul, Francesca Vitalini, Hao Wu, Christoph Wehmeyer, Jan-Hendrik Prinz, Péter Koltai, Benjamin Trendelkamp-Schroer, Stefan Klus and Patrick Gelß, for their help and many fruitful discussions. Thanks to all members of the CMB group for making the last four years such an exciting time. I'm also grateful to my wife Tamara for her support throughout these years. I would like to thank Deutsche Forschungsgemeinschaft, the Dahlem Research School, the Berlin Mathematical School and Einstein Center EcMath for financial support.

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und alle dabei verwendeten Hilfsmittel und Hilfen angegeben habe. Die Arbeit ist nicht schon einmal in einem früheren Promotionsverfahren eingereicht worden.

Berlin, 18. Januar 2017, ...

Feliks Nüske

Zusammenfassung

Die vorliegende Arbeit beschreibt eine Methode, genannt *variational approach to conformational dynamics (VAC)*, zur Analyse von Simulationsdaten von hochdimensionalen stochastischen Prozessen. Dabei liegt der Fokus auf reversiblen Markov-Prozessen und auf der Anwendung im Bereich von Molekulardynamik Simulationen. Die grundlegende Idee ist es, die führenden Eigenfunktionen des mit dem Markov-Prozess assoziierten Transferoperators aus einer vorab gewählten Menge von Basisfunktionen zu approximieren. Die auf diese Art approximierten Eigenfunktionen können zur niedrig-dimensionalen Darstellung des Prozesses verwendet werden. Zur Bestimmung der Approximation muss ein generalisiertes Eigenwertproblem gelöst werden, wobei die dafür benötigten Matrizen aus langen Simulationen berechnet werden können. In der Arbeit wurde die Verwendung von Tensorprodukt Darstellungen diskutiert, damit die Methode mit einer möglichst großen und dennoch interpretierbaren Basis verwendet werden kann. Um den dabei auftretenden "Fluch der Dimension" zu vermeiden, wurde ein Niedrigrang-Format, das *tensor-train-format*, verwendet. Die zugehörigen Algorithmen wurden an die Problemstellung angepasst und erfolgreich auf Beispielsysteme angewandt. Im letzten Teil der Arbeit wurde untersucht, wie die Methode auch mit Hilfe von vielen Kurzzeit-Simulationen verwendet werden kann. Diese Frage wurde zunächst für *Markov state models (MSM)* untersucht, die einen Spezialfall des VAC darstellen. Wir haben einen Ausdruck für den Fehler bei der MSM Schätzung aus zu kurzen Simulationen hergeleitet. Anschließend wurde erklärt, wie sich der Fehler mit Hilfe von *observable operator models (OOM)* korrigieren lässt. Die Diskussion algorithmischer Details und die Anwendung auf Beispielsysteme bilden den Abschluss der Arbeit.