

## Variational tensor approach for approximating the rare-event kinetics of macromolecular systems

Feliks Nüske, Reinhold Schneider, Francesca Vitalini, and Frank Noé

Citation: *The Journal of Chemical Physics* **144**, 054105 (2016); doi: 10.1063/1.4940774

View online: <http://dx.doi.org/10.1063/1.4940774>

View Table of Contents: <http://scitation.aip.org/content/aip/journal/jcp/144/5?ver=pdfcov>

Published by the [AIP Publishing](#)

---

### Articles you may be interested in

[Overcoming potential energy distortions in constrained internal coordinate molecular dynamics simulations](#)  
*J. Chem. Phys.* **144**, 044112 (2016); 10.1063/1.4939532

[Comparison of the kinetics of different Markov models for ligand binding under varying conditions](#)  
*J. Chem. Phys.* **142**, 094104 (2015); 10.1063/1.4908531

[Calculation of vibrational frequencies through a variational reduced-coupling approach](#)  
*J. Chem. Phys.* **127**, 164118 (2007); 10.1063/1.2798104

[Calculation of the entropy and free energy by the hypothetical scanning Monte Carlo method: Application to peptides](#)  
*J. Chem. Phys.* **122**, 054903 (2005); 10.1063/1.1835911

[Macromolecular conformational dynamics in torsional angle space](#)  
*J. Chem. Phys.* **108**, 271 (1998); 10.1063/1.475378

---



**NEW Special Topic Sections**

**NOW ONLINE**  
Lithium Niobate Properties and Applications:  
Reviews of Emerging Trends

**AIP** | Applied Physics  
Reviews

# Variational tensor approach for approximating the rare-event kinetics of macromolecular systems

Feliks Nüske,<sup>1,a)</sup> Reinhold Schneider,<sup>2</sup> Francesca Vitalini,<sup>3</sup> and Frank Noé<sup>1,a)</sup>

<sup>1</sup>Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 6, 14195 Berlin, Germany

<sup>2</sup>Institut für Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany

<sup>3</sup>Department of Chemistry, Freie Universität Berlin, Takustr. 3, 14195 Berlin, Germany

(Received 17 November 2015; accepted 13 January 2016; published online 2 February 2016)

Essential information about the stationary and slow kinetic properties of macromolecules is contained in the eigenvalues and eigenfunctions of the dynamical operator of the molecular dynamics. A recent variational formulation allows to optimally approximate these eigenvalues and eigenfunctions when a basis set for the eigenfunctions is provided. In this study, we propose that a suitable choice of basis functions is given by products of one-coordinate basis functions, which describe changes along internal molecular coordinates, such as dihedral angles or distances. A sparse tensor product approach is employed in order to avoid a combinatorial explosion of products, i.e., of the basis set size. Our results suggest that the high-dimensional eigenfunctions can be well approximated with relatively small basis set sizes. © 2016 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4940774>]

## I. INTRODUCTION

Molecular dynamics (MD) simulations have matured to be a standard tool to explore the conformations and conformational dynamics of biological macromolecules, such as proteins.<sup>1</sup> In MD simulations, the time integration of classical mechanics is combined with stochastic contributions, which ensure that certain thermodynamic quantities such as the temperature remain constant on average. For any atomic configuration  $\mathbf{x}$  in the configuration space  $\Omega$ , the classical force acting on the system is given by the gradient of an empirical potential energy function  $V(\mathbf{x})$ , which is the model for the molecular system under investigation. Such a potential energy function is usually the sum of many terms involving only a few atoms, typically depending on intramolecular distances  $\rho_{ij}$ , bond angles  $\alpha_{ijk}$ , and dihedral angles  $\beta_{ijkl}$ . A standard form of  $V$  is

$$\begin{aligned}
 V(\mathbf{x}) = & \sum_{\text{bonds}(i,j)} V_{\text{bond}}(\rho_{ij}) \\
 & + \sum_{\text{angles}(i,j,k)} V_{\text{angle}}(\alpha_{ijk}) \\
 & + \sum_{\text{dihedrals}(i,j,k,l)} V_{\text{dih}}(\beta_{ijkl}) \\
 & + \sum_{\text{nonbonded}(i,j)} V_{\text{nonbonded}}(\rho_{ij}) + V_{\text{other}}(\mathbf{x}). \quad (1)
 \end{aligned}$$

The nonbonded terms normally comprise the van der Waals and electrostatic interactions. The long-ranged part of the nonbonded forces, which might be computed through particle-field methods such as particle mesh Ewald (PME), is included in  $V_{\text{other}}(\mathbf{x})$ .

The extraction of essential stationary and kinetic information from a set of trajectories  $\{\mathbf{x}_t\}$ , generated by MD simulations, has been an active field of research in recent years. A very powerful approach has been the construction of Markov state models (MSMs, also Markov models in short).<sup>2–7</sup> First, the sampled molecular conformations are clustered and the trajectories  $\{\mathbf{x}_t\}$  are re-written into discrete time series. Second, the conditional transition probabilities between the clusters are estimated. These transition probabilities form a row-stochastic transition matrix best approximating the discretized dynamics. Software for the automatic construction and estimation of Markov models is available.<sup>8,9</sup> Markov models have initiated a large amount of follow-up research, e.g., on the analysis of transition pathways through Markov models,<sup>10,11</sup> connecting molecular simulation and experimental observables via Markov models<sup>12–15</sup> and applications to various molecular processes.<sup>10,16–19</sup> A particularly interesting result from this line of research is that the essential information of the molecule's stationary and kinetic quantities is contained in the dominant eigenvalues and eigenfunctions of the dynamical operator of the molecular dynamics<sup>2</sup> and that the eigenvalues and eigenvectors of a Markov model transition matrix can be a good approximation thereof.<sup>20</sup>

A more recent development has generalized this result. Instead of constructing a Markov model, one can define any set of basis functions that map molecular configurations to real values and then try to approximate the eigenvalues and eigenfunctions of the MD operator by a linear combination of these basis functions.<sup>21</sup> This approach, named variational approach to conformation dynamics (VAC), may improve our ability to approximate and interpret the high-dimensional eigenfunctions. The construction of a Markov model requires a state space discretization that is typically obtained by some clustering routine. The discrete states obtained in this way lack a clear physical meaning in many cases. Within

<sup>a)</sup>feliks.nueske@fu-berlin.de and frank.noé@fu-berlin.de

the more general variational approach, we can replace the discretization step by the use of functions that more faithfully describe configuration changes in molecules. We note that there is a close connection to quantum mechanics, where a craft of choosing suitable basis sets for modeling specific chemical transitions has evolved. However, it is yet unclear which choices of basis functions are suitable to model classical conformational changes of macromolecules. We were prompted to model such conformational changes in terms of the elementary internal coordinates  $\rho_{ij}$ ,  $\alpha_{ijk}$ , and  $\beta_{ijkl}$ , as the potential in Eq. (1) depends on those, see Ref. 22 for some first results. However, these internal coordinates are coupled, thus we must allow our basis functions to represent couplings between internal coordinates in order to provide a good approximation of the high-dimensional eigenfunctions. In Ref. 23, a basis set for peptides with coupling between adjacent  $\phi/\psi$  backbone angles was designed, and products between the basis functions were considered.

In general, this idea leads to the following approach: for all  $d$  elementary internal coordinates  $x_p$ ,  $p = 1, \dots, d$  under consideration, we select  $n$  one-coordinate basis functions  $f_{i_p}^p(x_p)$ ,  $i_p = 1, \dots, n$ , defined on the single coordinate  $x_p$ . These functions should be chosen such as to be able to represent every significant variation along the single coordinates. A coupled basis set is then obtained by considering all possible products between one-coordinate basis functions,

$$\mathcal{X} = \{f_{i_1}^1(x_1)f_{i_2}^2(x_2)\dots f_{i_d}^d(x_d), i_1, i_2, \dots, i_d = 1, \dots, n\}. \quad (2)$$

Clearly, the size of basis set Eq. (2) will combinatorially explode, as for  $n$  one-coordinate functions, we have  $n^d$  possible products, making even the sheer enumeration of them unfeasible for relatively small molecules.

Consequently, we need a systematic method to determine a sparse representation of the eigenfunctions from basis set Eq. (2). The high degree of redundancy among the set of intramolecular coordinates (for instance, among many distances) suggests that finding such a representation should be possible. Here, we propose a new method based on the tensor-train (TT) format and its learning algorithm, the alternating linear scheme (ALS). Tensor-trains have been introduced by Refs. 24 and 25. However, they have been known in quantum physics as matrix product states since the 1990s, see Ref. 26. A version of the ALS was introduced by Ref. 27 as the density matrix renormalization group (DMRG) algorithm, developed for the ground-state calculation of 1D spin chains. For a recent review on matrix product states and tensor product approximations in quantum physics, see Ref. 28 and references therein. It should also be noted that tensor-trains are special cases of hierarchical tensor representations as introduced by Ref. 29, also called tree tensor networks.<sup>28</sup> Please see Ref. 30 for a general overview of these concepts. A hierarchical tensor representation requires the choice of a dimension tree. One of the simplest and most straightforward choices is a linear and unbalanced tree, which gives rise to a tensor-train. In this work, we will stick to the TT-format, mainly because of its conceptual simplicity and on the basis of tensor-trains and ALS having provided promising results for model systems.<sup>31</sup>

The main contribution of this paper is to demonstrate the usefulness of the variational tensor approach for the analysis of data generated by MD simulations. To this end, after briefly reviewing the spectral theory of transfer operators and the variational approach in Sections II and III, we present some essential theory of tensor-trains in Section IV A, in a format which is suitable to our problem. In contrast to most of the previous applications of tensor product formats, the dynamical operator in our setting is only sampled by the simulation data, and no explicit low-rank representation of the operator is available. In Section IV B, we present a modification of the ALS which is suitable for our problem setting. The method is intuitive and appears to be useful in practical cases. We conclude by presenting the successful application of the method to MD simulations of two benchmark systems, deca-alanine (Ala<sub>10</sub>) and bovine pancreatic trypsin inhibitor (BPTI), for which reference solutions using Markov models are known. We also suggest a postprocessing method that allows to identify a reduced subset of input coordinates which is a representative for the slow dynamics. Its usefulness is confirmed by the applications.

## II. THEORY

We first outline the basic assumptions made for our MD implementation, following Ref. 6. The MD simulation samples from a time- and space-continuous ergodic Markov process in a state space  $\Omega$ . These dynamics have a unique stationary distribution  $\pi$ , given by the Boltzmann distribution,

$$\pi(x) \propto \exp(-\beta V(x)). \quad (3)$$

This means that independent of the starting point, every MD simulation that is sufficiently long will sample from this stationary distribution. Here,  $\beta = (k_B T)^{-1}$  is the inverse temperature,  $k_B$  is the Boltzmann constant, and  $T$  is the temperature.

Another crucial ingredient for the system's description is the transition density function  $p(x, y; \tau)$ , which describes the conditional probability of the system travelling from  $x$  to  $y$  over a finite time step  $\tau > 0$ , also called the lag time. Although the transition density is not known to us, we can assume that it satisfies

$$\pi(x)p(x, y; \tau) = \pi(y)p(y, x; \tau) \quad \forall x, y \in \Omega, \quad (4)$$

which is the ‘‘detailed balance’’ condition. Equation (4) is physically motivated: It prohibits the existence of a preferred direction in the system, which could be used to produce work. Of course, it must be enforced by the molecular dynamics integrator in order to hold in the simulation. Using the transfer density, we can understand how ensembles of trajectories evolve in time: If an ensemble of trajectories is started according to a distribution  $p_0$  at time  $t = 0$ , then the corresponding distribution  $p_\tau$  at time  $\tau$  is given by

$$p_\tau(y) = \mathcal{P}(\tau)p_0(y) \quad (5)$$

$$= \int_{\Omega} p(x, y; \tau)p_0(x)dx. \quad (6)$$

This defines a linear integral operator, the propagator  $\mathcal{P}(\tau)$ . Let us summarize the main properties of the propagator, see again Ref. 6 for details. If we introduce the weighted scalar product

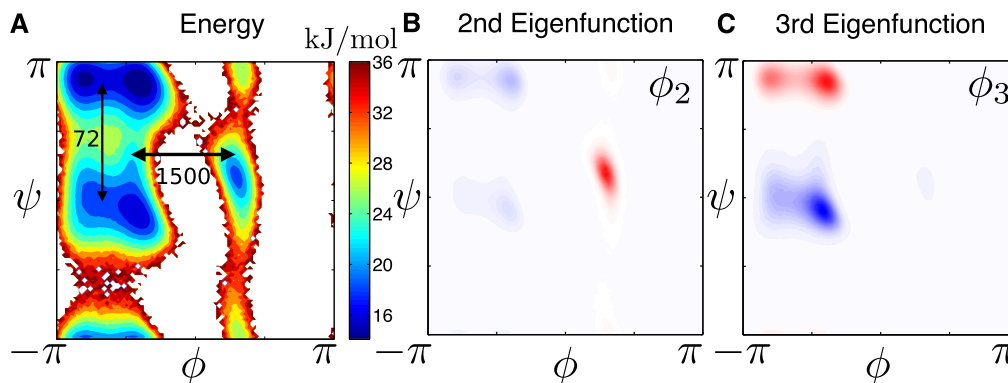


FIG. 1. Conformation dynamics of the alanine dipeptide. In panel (a), we show the energy of the system as a function of the two dihedral angles  $\phi$  and  $\psi$  (not to be confused with the eigenfunctions) in units of kJ/mol. There are three minima of the energy landscape (blue color) in the upper left, central left, and central right region, where the system remains for most of the time. The two slow transitions are indicated by black arrows: the slowest is from the left to the right part of the plane, while the second slowest transition occurs between the upper left and central left minimum. The corresponding relaxation time scales  $t_2$ ,  $t_3$  in units of ps are printed next to the arrows. These two transitions are encoded in the sign structure of the second and third propagator eigenfunctions  $\phi_2$ ,  $\phi_3$ , shown in panels (b) and (c). It can be seen that  $\phi_2$  changes sign (red: positive, blue: negative) in the  $\phi$ -direction, i.e., the slowest transition with time scale  $t_2$  is associated with a switch of the  $\phi$ -angle. The intensities are not equal on the left and on the right as the population of the two left minima is much higher compared to the right. In the same way, the second slowest transition can be recognized from the sign structure of  $\phi_3$ . This figure has been adapted with permission from Figure 4 of Nüske *et al.*, J. Chem. Theory Comput. **10**, 1739–1752 (2014). Copyright 2014 American Chemical Society.

$$\langle u, v \rangle_{\pi^{-1}} = \int_{\Omega} u(x) v(x) \pi^{-1}(x) dx, \quad (7)$$

the propagator is bounded and self-adjoint (due to Eq. (4)) on the space  $L^2_{\pi^{-1}}$  of functions  $u$  that satisfy  $\langle u, u \rangle_{\pi^{-1}} < \infty$ . It can be shown that its largest eigenvalue is  $\lambda_1 = 1$ , whereas all remaining eigenvalues are strictly smaller than one in absolute value. Moreover, there is a number  $M$  of dominant positive eigenvalues,

$$1 = \lambda_1 > \lambda_2 \geq \dots \geq \lambda_M > 0. \quad (8)$$

The remaining spectrum is contained in an interval  $[-R, R]$ , with  $R < \lambda_M$ . The eigenfunction corresponding to the largest eigenvalue  $\lambda_1 = 1$  is the stationary density  $\pi$ .

The dominant eigenvalues  $\lambda_m$  and corresponding eigenfunctions  $\phi_m$  are the key to understand the complex dynamics of the molecule. All eigenvalues except the first decay exponentially with the lag time  $\tau$ , i.e.,

$$\lambda_m(\tau) = e^{-\kappa_m \tau}, \quad m = 2, \dots, M, \quad (9)$$

with some relaxation rates  $\kappa_m > 0$ . Using the eigenfunctions and eigenvalues, the action of the propagator on a probability density  $p_0$  can be written as

$$\mathcal{P}(\tau)p_0 = \sum_{m=1}^M e^{-\kappa_m \tau} \langle \phi_m, p_0 \rangle_{\pi^{-1}} \phi_m + \mathcal{P}_{\text{fast}} p_0. \quad (10)$$

In most systems that we are interested in, the contribution of  $\mathcal{P}_{\text{fast}}$  vanishes quickly with the lag time. This implies that for large enough  $\tau$ , only a finite number  $M$  of the terms in expansion (10) are still present, and the operator's action can be understood in terms of only finitely many processes

$$\mathcal{P}(\tau)p_0 \approx \sum_{m=1}^M e^{-\kappa_m \tau} \langle \phi_m, p_0 \rangle_{\pi^{-1}} \phi_m. \quad (11)$$

Each of the dominant eigenfunctions  $\phi_2, \dots, \phi_m$  typically carries the same sign on some specific regions of the state space, but changes its sign between those regions. In this way, the function encodes parts of the state space where the system

generally remains for a long time, while it rarely transitions between them.<sup>2</sup> This concept is known as metastability and is a typical feature of biomolecules, where the metastable regions are frequently associated with biological function of the molecule, e.g., the ability/inability to bind to a binding partner. Thus, each term in Eq. (11) corresponds to a relaxation process that transports probability between the metastable regions in order to equilibrate the system towards the stationary distribution. The relaxation time of each process is called the implied time scale and can be computed from the dominant eigenvalues,

$$t_m = \frac{1}{\kappa_m} = -\frac{\tau}{\log(\lambda_m(\tau))}. \quad (12)$$

An illustration of this concept is shown in Figure 1 for a very small toy system, the alanine dipeptide (Ac-Ala-NHMe). This molecule consists of a single amino acid, alanine, capped by two protection groups mimicking the behaviour of the amino acid in a chain, like in a protein. It has been used as a test case for many studies in recent years, e.g., Ref. 32, because it is easy to produce sufficient simulation data and consequently, its slow dynamics are very well understood.

Given this theoretical assessment, we now describe how to practically approximate the eigenvalues and eigenfunctions.

### III. DISCRETIZATION

#### A. Method of linear variation

In order to perform the approximation of eigenfunctions in practice, we consider a different operator, the transfer operator

$$\mathcal{T}(\tau)u(y) = \frac{1}{\pi(y)} \int_{\Omega} p(x, y; \tau) \pi(x) u(x) dx. \quad (13)$$

The transfer operator is equivalent to the propagator in the sense that it possesses the same eigenvalues  $\lambda_m$ , and its corresponding eigenfunctions  $\psi_m$  can be computed from the propagator eigenfunctions  $\phi_m$  via

$$\psi_m(x) = \pi^{-1}(x)\phi_m(x). \quad (14)$$

In particular, the sign structure of the  $\psi_m$  is identical to that of the  $\phi_m$ . The transfer operator eigenfunctions  $\psi_m$  are orthonormal with respect to the scalar product

$$\langle u, v \rangle_\pi = \int_\Omega u(x) v(x) \pi(x) dx, \quad (15)$$

and the operator is self-adjoint with respect to this inner product. We have the traditional variational formulation<sup>33</sup>

$$\sum_{m=1}^M \lambda_m = \sup_{u_1, \dots, u_M} \sum_{m=1}^M \langle \mathcal{T}(\tau) u_m, u_m \rangle_\pi, \quad (16)$$

$$\langle u_m, u_{m'} \rangle_\pi = \delta_{m, m'}. \quad (17)$$

The sum of the first  $M$  exact eigenvalues maximizes the partial trace  $\sum_{m=1}^M \langle \mathcal{T}(\tau) u_m, u_m \rangle_\pi$ , where the  $u_m$  are a set of  $M$  orthonormal functions. Equality is attained exactly for the true eigenfunctions  $\psi_1, \dots, \psi_M$ . Our goal is to obtain an approximation of the true eigenfunctions from a finite space of trial functions  $\chi_1, \dots, \chi_N$ . Therefore, one can restrict the above problem to the space spanned by the functions  $\chi_i$ . It is also a well-known result<sup>34</sup> that the right-hand side of Eq. (16) is then maximized, subjected to constraint (17), by the functions

$$\hat{\psi}_m = \sum_{i=1}^N \mathbf{V}_{im} \chi_i. \quad (18)$$

The columns of the matrix  $\mathbf{V}$  correspond to the first  $M$  eigenvectors of the generalized eigenvalue problem

$$\mathbf{C}^\tau \mathbf{V}(:, m) = \hat{\lambda}_m \mathbf{C}^0 \mathbf{V}(:, m), \quad m = 1, \dots, M, \quad (19)$$

$$c_{ij}^\tau = \langle \mathcal{T}(\tau) \chi_i, \chi_j \rangle_\pi, \quad (20)$$

$$c_{ij}^0 = \langle \chi_i, \chi_j \rangle_\pi. \quad (21)$$

The eigenvalues  $\hat{\lambda}_m$  can be used as estimates of the true eigenvalues  $\lambda_m$ , and the functions  $\hat{\psi}_m$  serve as approximations of the true eigenfunctions. We have that  $\hat{\lambda}_m = \langle \mathcal{T}(\tau) \hat{\psi}_m, \hat{\psi}_m \rangle_\pi$ , and it follows from Eq. (16) that

$$\sum_{m=1}^M \hat{\lambda}_m \leq \sum_{m=1}^M \lambda_m. \quad (22)$$

Variational principle Eq. (22) also holds for the individual eigenvalues  $\hat{\lambda}_m$ , see, e.g., Ref. 21

$$\hat{\lambda}_m < \lambda_m, \text{ if } \hat{\psi}_m \neq \psi_m, \quad (23)$$

$$\hat{\lambda}_m = \lambda_m, \text{ if } \hat{\psi}_m = \psi_m \quad m = 1, \dots, M. \quad (24)$$

The matrix entries in Eqs. (20) and (21) are not exactly computable, because the transfer density defining  $\mathcal{T}(\tau)$  is not known in an analytically practical form, and the integration spaces are very high-dimensional. An exception are models that directly parametrize the transfer density in a way that allows us to evaluate Eqs. (20) and (21), see, e.g., Ref. 35.

However, as pointed out in Refs. 21 and 22, we can still estimate the matrices  $\mathbf{C}^0$  and  $\mathbf{C}^\tau$  as correlation matrices between basis functions, as their entries are spatial expectation values. From a sufficiently long realization of the process, they can be estimated by time averages as follows:

$$c_{ij}^\tau \approx \frac{1}{T - \tau} \sum_{t=1}^{T-\tau} \chi_i(\mathbf{x}_t) \chi_j(\mathbf{x}_{t+\tau}), \quad (25)$$

$$c_{ij}^0 \approx \frac{1}{T} \sum_{t=1}^T \chi_i(\mathbf{x}_t) \chi_j(\mathbf{x}_t), \quad (26)$$

where  $T$  is the total length of the realization. This is not true for the corresponding matrix approximation of the propagator  $\mathcal{P}(\tau)$ , as the matrix elements cannot be interpreted as correlations. It is worth noting that Markov state models are a special case of the above formulation for the choice of characteristic functions of sets as basis functions.<sup>21,22</sup> MSMs have the additional benefit that the estimators for Eqs. (25) and (26) are conditional estimators, and this allows us to use short trajectories that are not in global equilibrium.<sup>6</sup> This important property can also be achieved for other basis sets that are probability densities, such as Gaussian distributions.<sup>35</sup> For arbitrary basis functions, estimator Eqs. (25) and (26) are incorrect if the data are not in global equilibrium. It is an open research question if these estimates can be used nevertheless.

## B. Coordinates and basis sets

In order to apply the above approach, it is still necessary to identify the set of functions or a sequence of sets of functions to be used. This problem has in fact three aspects, as we need to select the coordinates that serve as domains of these functions, the type of functions, and the way that functions depending on different coordinates are being combined.

Finding informative input coordinates is an ongoing research problem, see, e.g., Refs. 8, 36, and 37. However, this task is not the objective of the present study. In addition to that, we must decide what type of basis functions we use on the set of input coordinates and how to choose their parameters. In Ref. 22, we have used a set of one-coordinate basis functions defined on the individual dihedral angles (but no tensor products of them) for small peptides. The one-coordinate functions themselves were chosen as Gaussian functions, their parameters were selected for each dihedral angle separately. Let us emphasize that the choice of Gaussian basis functions was just for the sake of illustration — other basis sets might have worked equally well. In fact, it would be desirable to use basis functions which carry some chemical or physical information. They might, for instance, encode the torsional rotamer or whether a hydrogen bond is formed or dissociated. This is another open line of research, for some first results in this direction, see Ref. 23.

In this work, we are focusing on the coupling problem: 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. Assuming that there are  $d$  input coordinates labeled  $x_1, x_2, \dots, x_d$ , and for each coordinate  $x_p$ , we have  $n$  one-coordinate basis functions  $f_{i_p}^p(x_p)$ , where  $p = 1, \dots, d$  and  $i_p = 1, \dots, n$  (the theory can trivially deal with  $p$ -dependent  $n$ , 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

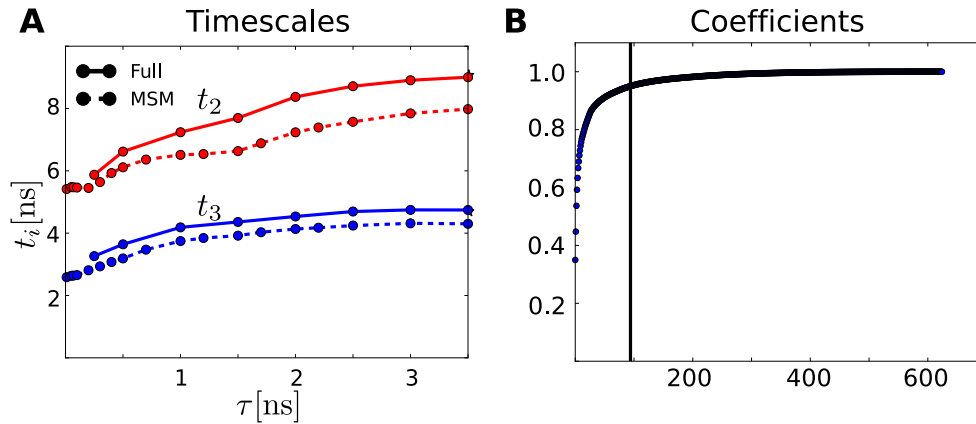


FIG. 2. Illustration of low-dimensional subspaces carrying the relevant information for the dimer Ac-Val-Ala-NHMe. Panel (a) shows the two slowest implied time scales  $t_2$ ,  $t_3$ , in red and blue, estimated by two different models: A reference MSM (dashed line, 9 states) and full product expansion Eq. (27) (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 with respect to weighted inner product Eq. (15). It takes about 90 basis functions to reproduce 95% of the norm, as indicated by the black vertical line.

$\psi_m$ ,  $m = 1, 2, \dots$  by a function  $\hat{\psi}_m$  that is a linear combination of all possible products of the  $f_{i_p}^p$ ,

$$\hat{\psi}_m(x_1, \dots, x_d) = \sum_{i_1, \dots, i_d} U_m(i_1, \dots, i_d) f_{i_1}^1(x_1) \dots f_{i_d}^d(x_d). \quad (27)$$

The basis functions for the variational approach are the products  $f_{i_1}^1(x_1) \dots f_{i_d}^d(x_d)$ , and  $U_m$  is a  $d$ -dimensional array (tensor) containing the expansion coefficients of all these products. As we can immediately see, the number of basis functions used in this expansion 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. (25) and (26) 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 is comprised of  $5^4 = 625$  functions. In Figure 2(a), we check the accuracy of the model by comparing the two slowest implied time scales  $t_2$ ,  $t_3$  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. 23. 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 2(b) 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  $\pi$ -weighted inner product Eq. (15). 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 Sec. IV.

## IV. TENSOR PRODUCT APPROXIMATIONS

### A. Tensor-train-format

The problem of finding a computationally feasible approximation to a high-dimensional representation like Eq. (27) 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 TT-format, which has been introduced in Refs. 24 and 25.

A function in TT-format still possesses a high-dimensional representation like Eq. (27), but the coefficient array  $U_m$  has a special structure as in Eq. (29),<sup>25</sup>

$$\hat{\psi}_m = \sum_{i_1, \dots, i_d} U_m(i_1, \dots, i_d) f_{i_1}^1(x_1) \dots f_{i_d}^d(x_d) \quad (28)$$

$$= \sum_{i_1, \dots, i_d} \left[ \sum_{k_1=1}^{r_1} \dots \sum_{k_{d-1}=1}^{r_{d-1}} U_1(i_1, k_1) U_2(k_1, i_2, k_2) \dots U_d(k_{d-1}, i_d) \right] f_{i_1}^1(x_1) \dots f_{i_d}^d(x_d). \quad (29)$$

Here,  $U_1 \in \mathbb{R}^{n \times r_1}$ ,  $U_d \in \mathbb{R}^{r_{d-1} \times n}$  are matrices and  $U_p \in \mathbb{R}^{r_{p-1} \times n \times r_p}$ ,  $p = 2, \dots, d-1$ , are three-dimensional arrays. Consequently, for every choice of  $i_1, \dots, i_d$ , the arrays  $U_1$  and

$U_d$  turn into vectors  $U_1(i_1)$ ,  $U_d(i_d)$ , whereas all other arrays  $U_2, \dots, U_{d-1}$  become matrices  $U_2(i_2), \dots, U_{d-1}(i_{d-1})$ , and the coefficient  $U_m(i_1, \dots, i_d)$  can be computed by a repeated

matrix-vector multiplication,

$$U_m(i_1, \dots, i_d) = U_1(i_1)U_2(i_2) \dots U_d(i_d). \quad (30)$$

Thus, only the arrays  $U_1, \dots, U_d$  need to be stored, and the number of parameters in these arrays is linear in the dimension  $d$ , see Ref. 25.

The intuition behind this representation is that only limited information is passed on from one variable to the next in the sequence  $x_1, \dots, x_d$ . To see this, consider the case  $d = 4$ , and re-order Eq. (29) as follows:

$$\begin{aligned} \hat{\psi}_m &= \sum_{k_1, i_2, k_2} U_2(\mathbf{k}_1, i_2, \mathbf{k}_2) f_{i_2}^2(x_2) \\ &\cdot \left[ \sum_{i_1} U_1(i_1, \mathbf{k}_1) f_{i_1}^1(x_1) \right] \\ &\cdot \left[ \sum_{i_3, i_4} \sum_{k_3} U_3(\mathbf{k}_2, i_3, k_3) U_4(k_3, i_4) f_{i_3}^3(x_3) \cdot f_{i_4}^4(x_4) \right] \\ &= \sum_{k_1, i_2, k_2} U_2(\mathbf{k}_1, i_2, \mathbf{k}_2) f_{i_2}^2(x_2) \cdot g_{\mathbf{k}_1}^2(x_1) \cdot h_{\mathbf{k}_2}^2(x_3, x_4). \quad (31) \end{aligned}$$

The expressions shown in brackets in Eq. (31) contain exactly one free index  $k_1$  and  $k_2$ , respectively, indicated by the bold-face letter. Thus, it makes sense to define functions  $g_{k_1}^2, h_{k_2}^2$  by these expressions, which leads us to the representation in Eq. (32). The meaning of Eq. (32) 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  $f_{i_2}^2$  depending on the variable  $x_2$ , a function  $g_{k_1}^2$  which depends on all variables up to  $x_2$ , and another function  $h_{k_2}^2$  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. (31) corresponds to panel B in Fig. 3. However, this is not the only way to re-order Eq. (29), 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. 3. In the general case, the re-ordering centered around coordinate  $x_p$  is given by

$$\begin{aligned} \hat{\psi}_m &= \sum_{k_{p-1}, i_p, k_p} U_p(\mathbf{k}_{p-1}, i_p, \mathbf{k}_p) f_{i_p}^p(x_p) \\ &\cdot \left[ \sum_{i_1, \dots, i_{p-1}} \sum_{k_1, \dots, k_{p-2}} U_1(i_1, k_1) \dots U_{p-1}(k_{p-2}, i_{p-1}, \mathbf{k}_{p-1}) f_{i_1}^1(x_1) \dots f_{i_{p-1}}^{p-1}(x_{p-1}) \right] \\ &\cdot \left[ \sum_{i_{p+1}, \dots, i_d} \sum_{k_{p+1}, \dots, k_{d-1}} U_{p+1}(\mathbf{k}_p, i_{p+1}, k_{p+1}) \dots U_d(k_{d-1}, i_d) f_{i_{p+1}}^{p+1}(x_{p+1}) \dots f_{i_d}^d(x_d) \right] \\ &= \sum_{k_{p-1}, i_p, k_p} U_p(\mathbf{k}_{p-1}, i_p, \mathbf{k}_p) f_{i_p}^p(x_p) \cdot g_{\mathbf{k}_{p-1}}^p(x_1, \dots, x_{p-1}) \cdot h_{\mathbf{k}_p}^p(x_{p+1}, \dots, x_d). \quad (32) \end{aligned}$$

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

$$g_{k_p}^{p+1} = \sum_{k_{p-1}, i_p} U_p(k_{p-1}, i_p, k_p) g_{k_{p-1}}^p f_{i_p}^p, \quad (35)$$

$$h_{k_{p-1}}^{p-1} = \sum_{i_p, k_p} U_p(k_{p-1}, i_p, k_p) f_{i_p}^p h_{k_p}^p. \quad (36)$$

## B. Alternating linear scheme

In order to make use of the tensor-train-format in practice, we need a method to determine the optimal components  $U_p$  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 ALS, which is an iterative learning algorithm that arises naturally from the TT-format, see Ref. 31. Second, the block-TT-format from Refs. 38 and 39, which is a modification of tensor-trains allowing for the simultaneous approximation of multiple functions 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.

The idea of alternating optimization is that in each iteration step, we attempt to update one component  $U_p$ , while keeping all others fixed. Starting from some initial guess for all  $U_p$ , the method will first update  $U_1$  while  $U_2, \dots, U_d$  are fixed, then it will update  $U_2$  with  $U_1, U_3, \dots, U_d$  fixed, and so on, until  $U_d$  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.

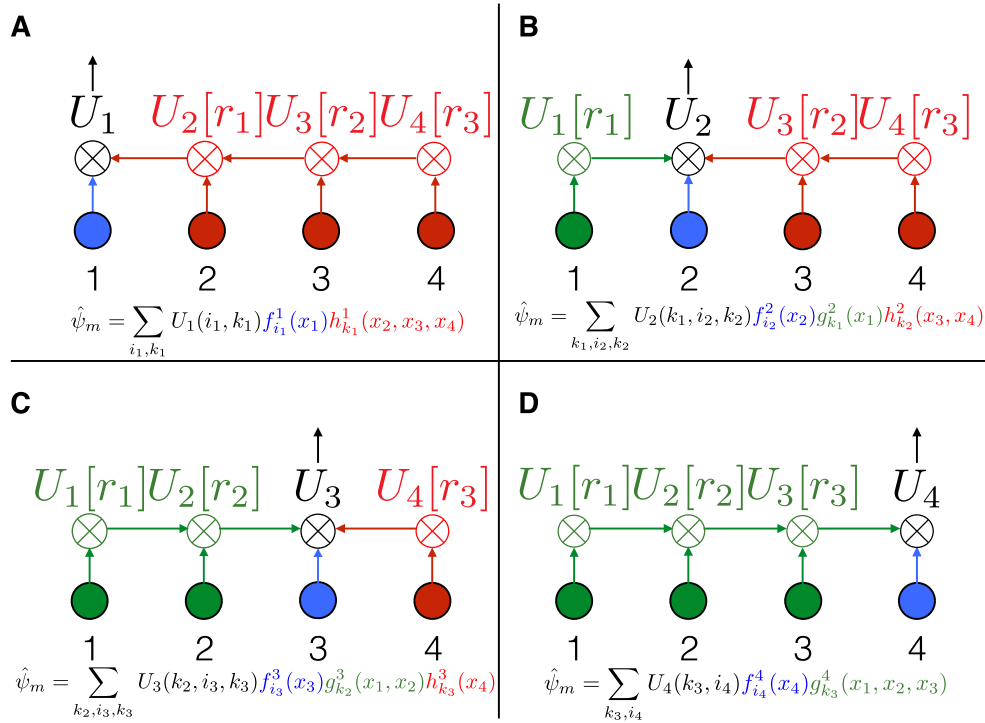


FIG. 3. Illustration of a function  $\hat{\psi}_m$  of  $d=4$  variables in tensor-train-format. The solid dots at the bottom represent the sets of one-coordinate basis functions  $f_{i_p}^p$ . Dots with the tensor product symbol  $\otimes$  contain all products of the incoming bases, indicated by the arrows. The arrays denoted by  $U_p[r_p]$  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  $U_1, \dots, U_{p-1}$  encode the information about the variables  $x_1, \dots, x_{p-1}$  into  $r_{p-1}$  functions. This process is shown in the green part of each panel. The arrays  $U_{p+1}, \dots, U_d$  encode the information about the variables  $x_{p+1}, \dots, x_d$  into  $r_p$  functions, which is shown in the red part of each panel. Both basis sets are combined with the one-coordinate functions  $f_{i_p}^p$  (shown in blue), and a linear combination of these products is selected by  $U_p$ , which is the final representation of  $\hat{\psi}_m$ .

As outlined in Sec. IV A, the component  $U_p$  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  $g_{k_p}^{p+1}$  or to encode the information about all coordinates  $x_p, \dots, x_d$  into  $r_{p-1}$  right interfaces  $h_{k_{p-1}}^{p-1}$ . 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  $U_p$  while all others are fixed. Following Sec. III A and recalling recursive definition Eq. (35), the optimal left interfaces  $g_{k_p}^{p+1}$  would be the linear combinations

$$g_{k_p}^{p+1}(U_p) = \sum_{k_{p-1}, i_p} U_p(k_{p-1}, i_p, k_p) g_{k_{p-1}}^p f_{i_p}^p \quad (37)$$

that maximize the eigenvalue sum

$$L_p(U_p) = \sum_{m=1}^M \hat{\lambda}_m(U_p) \quad (38)$$

resulting from generalized eigenvalue problem Eq. (19) for the basis

$$g_{k_p}^{p+1}(U_p) f_{i_{p+1}}^{p+1} f_{i_{p+2}}^{p+2} \cdots f_{i_d}^d, \quad (39)$$

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  $g_{k_p}^{p+1}$  which

maximize sum Eq. (38) for the reduced basis

$$g_{k_p}^{p+1}(U_p) f_{i_{p+1}}^{p+1} h_{k_{p+1}}^{p+1}, \quad (40)$$

see Fig. 4 for an illustration. This trick is inspired by the MALS<sup>31</sup> and the original DMRG algorithm.

Let us touch on the most important points of this optimization problem. First, we can set up a numerical

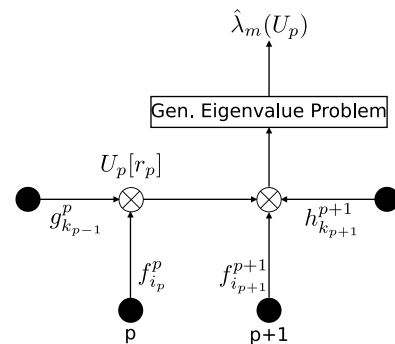


FIG. 4. Schematic representation of the optimization problem for the component  $U_p$ . This array selects  $r_p$  linear combinations of the products  $g_{k_{p-1}}^p \cdot f_{i_p}^p$  (see Eq. (35)) to form a new basis  $g_{k_p}^{p+1}(U_p)$  in an optimal way. Optimality is defined as follows: We combine the basis  $g_{k_p}^{p+1}(U_p)$  with the one-coordinate functions  $f_{i_{p+1}}^{p+1}$  and with the right interfaces  $h_{k_{p+1}}^{p+1}$  at position  $p+1$ , to form basis Eq. (40). For this basis, we solve generalized eigenvalue problem Eq. (19) to obtain dominant eigenvalues  $\hat{\lambda}_m(U_p)$ . Optimality of the  $U_p$  is defined by maximizing sum Eq. (38) of the  $\hat{\lambda}_m(U_p)$ .



## ALGORITHM 1. Summary of optimization algorithm.

---

```

1:  $q = 0$ 
2: repeat
3:    $q += 1$ 
4:   for  $p = 1, \dots, d-2$  do
5:     Solve Eq. (19) for the four-fold basis Eq. (41), obtain eigenvalues
        $\hat{\lambda}_m^{p,p+1}$ 
6:     Update reference eigenvalue sum  $L_{\text{ref}}$ 
7:     for  $r_p = 1, \dots, d$  do
8:       Optimize coefficients  $U_p(k_{p-1}, i_p, k_p)$  s.t.  $L_p(U_p)$  defined
       by Eq. (38) is maximal
9:       if  $L_p(U_p) \geq \epsilon_{\text{rank}} \cdot L_{\text{ref}}$  then
10:         Update  $U_p, g_{k_p}^{p+1}, r_p$ 
11:       break
12:     end if
13:   end for
14: end for
15: Repeat this in backward direction for  $p = d, \dots, 3$ 
16: until  $|L_p^q(U_p) - L_p^{q-1}(U_p)| < \epsilon_{\text{iter}} \forall p$ 

```

---

optimization method for Eq. (38) if  $r_p$  is fixed, please see Appendix C for an explanation. Therefore, we sequentially determine the optimal component  $U_p$  for increasing values of the rank  $r_p$  and accept  $U_p$  as the solution if the eigenvalue sum  $L_p(U_p)$  matches a reference value  $L_{\text{ref}}$  up to a tolerance  $\epsilon_{\text{rank}}$ . If accepted,  $U_p$  becomes the new  $p$ th component, the functions  $g_{k_p}^{p+1}(U_p)$  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 repeated. The reference  $L_{\text{ref}}$  is obtained as follows: As a first step, we always evaluate a four-fold product basis defined by the functions

$$g_{k_{p-1}}^p f_{i_p}^p f_{i_{p+1}}^{p+1} h_{k_{p+1}}^{p+1} \quad (41)$$

and solve generalized eigenvalue problem Eq. (19) for this basis. We compute the dominant eigenvalue sum resulting from this problem,

$$L_p = \sum_{m=1}^M \hat{\lambda}_m^{p,p+1}. \quad (42)$$

Variational principle Eq. (16) implies that for any  $U_p$ , the eigenvalue sum  $L_p(U_p)$  is bounded from above by  $L_p$ . Thus, we keep a track of the maximal value obtained for  $L_p$  during the entire optimization process and store this maximum as the reference  $L_{\text{ref}}$ . Second, we enforce the first interface function  $g_1^{p+1}$  to be the constant function. This constraint ensures that the largest eigenvalue  $\hat{\lambda}_1(U_p)$  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  $L_p(U_p)$  from two subsequent forward and backward sweeps do not differ by more than a tolerance  $\epsilon_{\text{iter}}$ . A summary of the complete method is given in Algorithm 1.

## V. 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 the ten residue

peptide Ala<sub>10</sub>, the second is the 58 residue protein BPTI. Equilibrium trajectories that are orders of magnitude longer than the slowest relaxation time scales 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  $U_p$  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 a topic of further research. The setting for  $\epsilon_{\text{rank}}$  ensures that no important information is lost along the course of the iteration. The setting for  $\epsilon_{\text{iter}}$  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 time scale  $t_2$  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 Sec. II, we expect to find peaks of the population corresponding to the most negative and the 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 D 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.

### A. Deca-alanine

Ala<sub>10</sub> simply consists of ten residues of alanine in a chain. We use six equilibrium simulations of 500 ns each, accumulating to 3  $\mu$ s total simulation time. The simulations were produced at temperature 300 K using the GROMACS 4.5.5 simulation package, the Amber03 force field, and the TIP3P water model. See Appendix A for details.

The input coordinates used for this system are  $d = 16$  backbone dihedral angles from the eight internal residues of the chain. 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 functions used for each dihedral consisted of the first  $n = 7$  real Fourier (sine and cosine) waves. The lag time used for our analysis was  $\tau = 2$  ns. We can compare our results to a reference Markov state model, see, e.g., Ref. 22. This model was constructed by partitioning the Ramachandran plane of each pair of backbone dihedrals into three boxes corresponding to the minima of the single amino acid dynamics, see Fig. 1(a). The Markov states were then defined by all combinations of these boxes, the total number of states is thus  $8^3 = 6561$ . It is found that the slowest dynamical process in the system is the formation of a helix and occurs at an implied time scale  $t_2 \approx 7.5$ -8 ns.

Fig. 5(a) shows that the implied time scale  $t_2$  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  $-1.3 \pm 0.3$  and  $1.6 \pm 0.2$ ) 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  $\psi$ -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.

## B. 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.<sup>40</sup> This large dataset has become a benchmark system used in numerous studies in recent years. The slowest structural transition included in the  $C_\alpha$  dynamics has been identified by other kinetic models to be on a time scale  $t_2 \approx 40 \mu\text{s}$ , see Refs. 8 and 41 for details.

The coordinates used in order to apply our method are the distances between all  $C_\alpha$  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  $C_\alpha$  atoms has formed or not

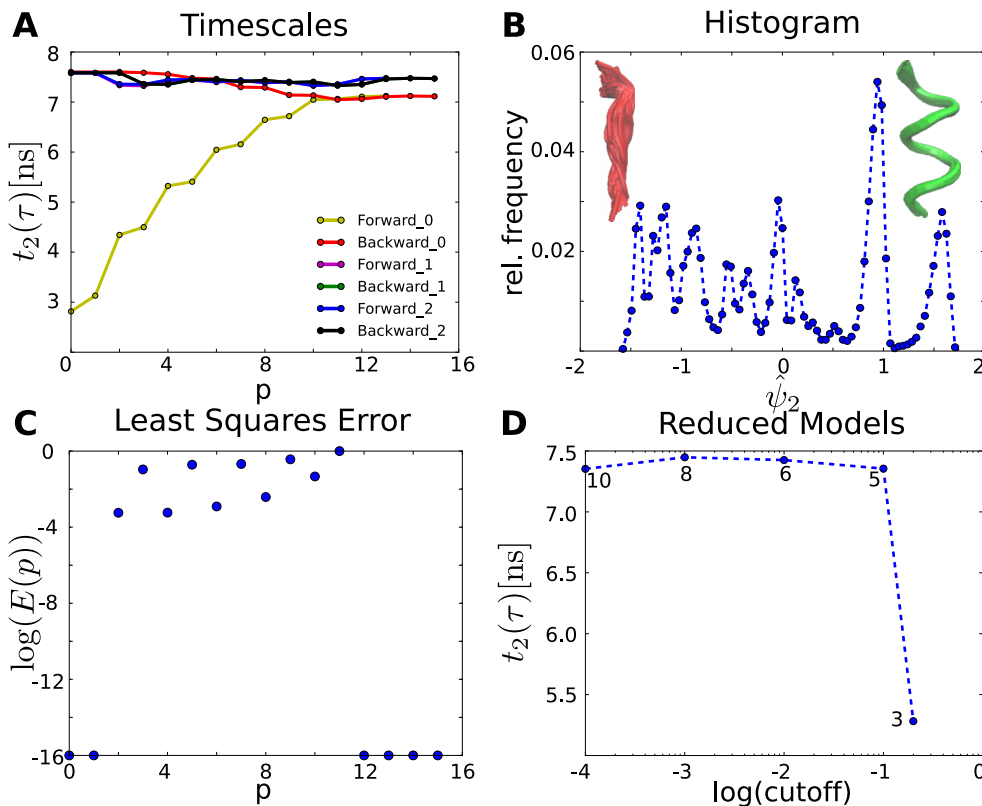


FIG. 5. Results for deca-alanine peptide. (a): Second implied time scale  $t_2$  in ns along the three forward and backward sweeps of the ALS-iteration. (b): Relative histogram of the simulation data along the  $\hat{\psi}_2$ -coordinate. We identify two peaks of the population corresponding to the most negative (around  $-1.3 \pm 0.3$ ) and the most positive values ( $1.6 \pm 0.2$ ) of the coordinate. Extracting 200 random frames from each of these peaks and superimposing their molecular structures show 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 time scale  $t_2$  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.

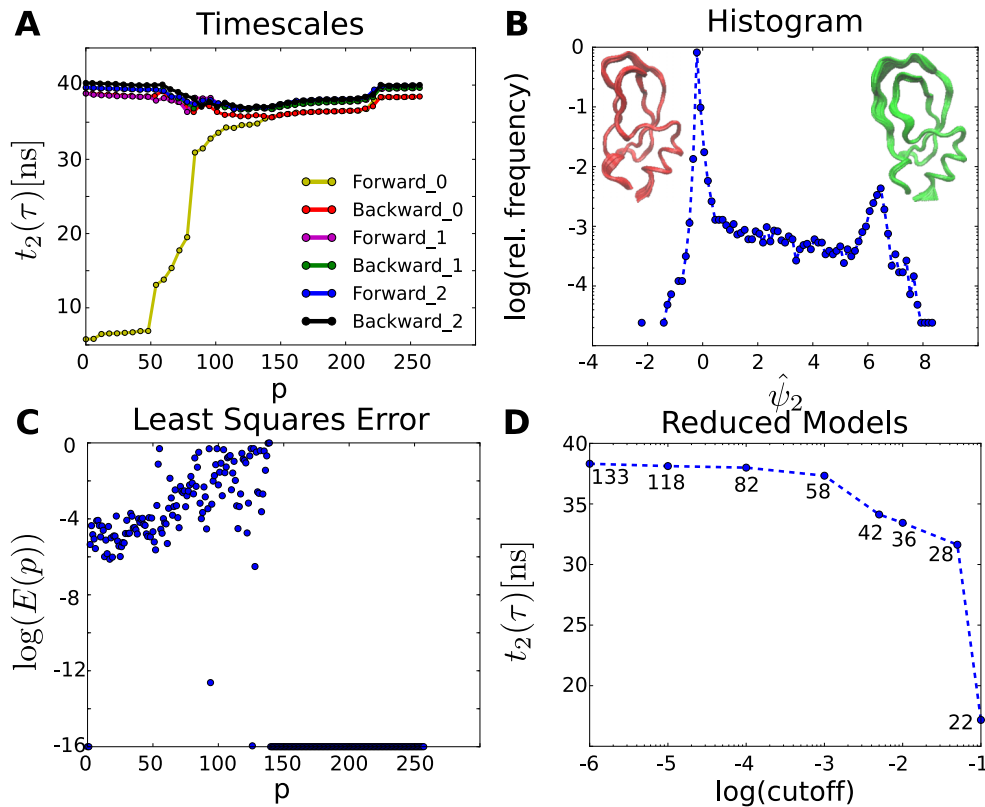


FIG. 6. Results for BPTI. (a): Second implied time scale  $t_2$  in  $\mu\text{s}$  along the three forward and backward sweeps of the ALS-iteration. (b): Relative histogram of the simulation data along the  $\hat{\psi}_2$ -coordinate. We identify two peaks of the population. Extracting 200 random frames 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 time scale  $t_2$  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.

$$f_2^p(x_p) = \frac{1 - (x_p/r_0)^{64}}{1 - (x_p/r_0)^{96}}, \quad (43)$$

where  $x_p$  is the  $C_\alpha$  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 dataset, 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  $\tau = 5 \mu\text{s}$ .

Figure 6(a) shows that again, the second implied time scale  $t_2$  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  $-0.2 \pm 0.5$  and  $6.5 \pm 1.0$ ). 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 folding of the loop on the N-terminal side of the backbone.<sup>8,41</sup>

The results of the least squares approximations are not as clear as in the previous example. It is apparent from Fig. 6(c) 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. 6(d) 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.

## VI. CONCLUSIONS

We have proposed a new approach to approximate eigenvalues and eigenfunctions of molecular dynamics operators by products of one-coordinate basis functions. The one-coordinate basis functions are used to indicate local changes in internal coordinates of the molecular system, such as rotamer changes in the dihedrals or formation/dissociation of contacts in the distances. Combining these one-coordinate functions by products allows us to encode complex couplings of coordinates (e.g., concerted changes of torsions or

simultaneous dissociation of multiple contacts). In order to avoid a combinatorial explosion of the products, we select a small subset of them using a sparse tensor product approach. Specifically, the TT-format and the method of linear variation are used in order to iteratively select a suitable subspace, from which the eigenvalues and eigenfunctions are approximated. We make use of the ALS as a learning algorithm, in a version which we have adapted to the problem setting.

Our results suggest that the TT-approach is suitable for selecting a sparse set of products of one-coordinate basis functions in order 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 postprocessing 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.

Although this research is still in its infancy, our work suggests that approximating eigenfunctions by a sparse set of products of one-coordinate basis functions may be a promising direction to compute conformation dynamics of macromolecules. 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. We will have to study the effects of non-equilibrium data on our current methods, and we will need to develop a framework where statistical uncertainty and discretization error can be balanced. However, we think that the methods presented in this work can be a useful foundation for the study of these problems.

## ACKNOWLEDGMENTS

This work was funded by the Einstein foundation Berlin through ECMath, by Deutsche Forschungsgemeinschaft

through SFB 1114, and by the European Commission through ERC starting grant “pcCell.” We thank Benjamin Trendelkamp-Schroer for helpful discussions and Cecilia Clementi for suggesting the basis set used for BPTI.

## APPENDIX A: SIMULATION SETUP OF 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,<sup>42</sup> 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  $V = 232.6 \text{ nm}^3$ , 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 500 ps NPT equilibration run. The temperature was set to  $T = 300 \text{ K}$ . The equilibration run was followed by a 500 ns production run, again at  $T = 300 \text{ K}$ . 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  $z$  directions. 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 were 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  $3 \mu\text{s}$  total data.

## APPENDIX B: RELATION TO THE BLOCK-TT-FORMAT

Our optimization method shown in Algorithm 1 is built on the modification of the ALS (Ref. 31) for the block-TT-format, see Refs. 38 and 39. 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  $U_p$  which carries an additional index  $m$ , enumerating the different functions, while all remaining components retain their structure as before. Eqs. (29) and (34) then turn into

$$\hat{\psi}_{\mathbf{m}} = \sum_{i_1, \dots, i_d} \left[ \sum_{k_1=1}^{r_1} \dots \sum_{k_{d-1}=1}^{r_{d-1}} U_1(i_1, k_1) \dots U_p(k_{p-1}, i_p, k_p, \mathbf{m}) \dots U_d(k_{d-1}, i_d) \right] f_{i_1}^1(x_1) \dots f_{i_d}^d(x_d), \quad (\text{B1})$$

$$\hat{\psi}_{\mathbf{m}} = \sum_{k_{p-1}, i_p, k_p} U_p(k_{p-1}, i_p, k_p, \mathbf{m}) f_{i_p}^p(x_p) \cdot g_{k_{p-1}}^p(x_1, \dots, x_{p-1}) \cdot h_{k_p}^p(x_{p+1}, \dots, x_d), \quad (\text{B2})$$

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  $U_p$ , while all others are fixed. We observe that we can solve

the eigenvalue problem Eq. (19) for the three-fold product basis in Eq. (B2), and we can update every slice  $U_p(:, :, :, m)$  by the  $m$ th eigenvector thus obtained. In order to proceed to the optimization of the next component  $U_{p+1}$ , however, the index  $m$  needs to be moved into  $U_{p+1}$  first. Otherwise,  $U_p$

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  $U_p$  into a matrix in  $\mathbb{R}^{r_{p-1} \times r_p \times M}$  and compute a low-rank decomposition, e.g., by singular value decomposition (SVD) or QR-decomposition,

$$U_p(k_{p-1}i_p, k_p m) = \sum_{k'_p=1}^{r'_p} V_p(k_{p-1}i_p, k'_p) W_p(k'_p, k_p m). \quad (\text{B3})$$

- Contract the arrays  $W_p$  and  $U_{p+1}$  by summing over  $k_p$ ,

$$\begin{aligned} \tilde{U}_{p+1}(k'_p, i_{p+1}, k_{p+1}, m) \\ = \sum_{k_p=1}^{r_p} W_p(k'_p, k_p m) U_{p+1}(k_p, i_{p+1}, k_{p+1}). \end{aligned} \quad (\text{B4})$$

After this, the  $p$ th component can be updated by  $V_p$ , which carries no more than three indices, while the  $p+1$ -st component can be updated by  $\tilde{U}_{p+1}$ , 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 decomposition Eq. (B3) 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 weighted inner product Eq. (15). Even for large ranks  $r'_p$ , yielding close approximations to the full matrix  $U_p$ , 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  $U_p$  provides a new left interface  $g_{k_p}^{p+1}(U_p)$ . The eigenvectors of generalized eigenvalue problem Eq. (19) parametrize  $M$  eigenfunctions in terms of reduced basis Eq. (40), yielding a component  $U_{p+1} \in \mathbb{R}^{r_p \times n \times r_{p+1} \times M}$ . 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.

## APPENDIX C: OPTIMIZATION PROBLEM FOR THE COMPONENTS $U_p$

Here, we formulate the optimization problem which needs to be solved for increasing ranks  $r_p$  in every iteration step of Algorithm 1. We seek to determine the optimal component  $U_p \in \mathbb{R}^{r_{p-1} \times n \times r_p}$ , s.t. eigenvalue sum Eq. (38) for reduced basis Eq. (40) 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  $\hat{\lambda}_m(U_p)$  with respect to the entries of  $U_p$ . These derivatives can be obtained as follows: The eigenvalues  $\hat{\lambda}_m(U_p)$  solve generalized eigenvalue problem Eq. (19) using the reduced correlation matrices  $\mathbf{C}^\tau(U_p)$ ,  $\mathbf{C}^0(U_p)$  between basis functions Eq. (40). These correlation matrices can be computed from the larger correlation matrices  $\mathbf{C}_{p,p+1}^\tau$ ,  $\mathbf{C}_{p,p+1}^0$  of four-fold product basis Eq. (41),

$$\mathbf{C}_{p,p+1}^\tau = \langle \mathcal{T}(\tau) g_{k_{p-1}}^{p-1} f_{i_p}^p f_{i_{p+1}}^{p+1} h_{k_{p+1}}^{p+1}, g_{l_{p-1}}^{p-1} f_{j_p}^p f_{j_{p+1}}^{p+1} h_{l_{p+1}}^{p+1} \rangle_\pi, \quad (\text{C1})$$

$$\mathbf{C}_{p,p+1}^0 = \langle g_{k_{p-1}}^{p-1} f_{i_p}^p f_{i_{p+1}}^{p+1} h_{k_{p+1}}^{p+1}, g_{l_{p-1}}^{p-1} f_{j_p}^p f_{j_{p+1}}^{p+1} h_{l_{p+1}}^{p+1} \rangle_\pi, \quad (\text{C2})$$

by the formulas

$$[\mathbf{C}^\tau(U_p)]_{l_{p-1} j_p, l_{p+1} j_{p+1}}^{k_{p-1} i_p, k_{p+1} i_{p+1}} = \sum_{k_{p-1}, i_p, l_{p-1}, j_p} U_p(k_{p-1}, i_p, k_p) [\mathbf{C}_{p,p+1}^\tau]_{l_{p-1}, j_p, j_{p+1}, l_{p+1}}^{k_{p-1}, i_p, i_{p+1}, k_{p+1}} U_p(l_{p-1}, j_p, l_p), \quad (\text{C3})$$

$$[\mathbf{C}^0(U_p)]_{l_{p-1} j_p, l_{p+1} j_{p+1}}^{k_{p-1} i_p, k_{p+1} i_{p+1}} = \sum_{k_{p-1}, i_p, l_{p-1}, j_p} U_p(k_{p-1}, i_p, k_p) [\mathbf{C}_{p,p+1}^0]_{l_{p-1}, j_p, j_{p+1}, l_{p+1}}^{k_{p-1}, i_p, i_{p+1}, k_{p+1}} U_p(l_{p-1}, j_p, l_p). \quad (\text{C4})$$

Using these formulas, we can differentiate the matrix entries of  $\mathbf{C}^\tau(U_p)$  and  $\mathbf{C}^0(U_p)$  with respect to the variables  $U_p$ ,

$$\begin{aligned} \frac{\partial [\mathbf{C}^\tau(U_p)]_{l_{p-1} j_p, l_{p+1} j_{p+1}}^{k_{p-1} i_p, k_{p+1} i_{p+1}}}{\partial U_p(k'_{p-1}, i'_p, k'_p)} &= \sum_{l_{p-1}, j_p} [\mathbf{C}_{p,p+1}^\tau]_{l_{p-1}, j_p, j_{p+1}, k_{p+1}}^{k'_{p-1}, i'_p, i_{p+1}, k_{p+1}} U_p(l_{p-1}, j_p, l_p) \delta_{k_p, k'_p} \\ &+ \sum_{k_{p-1}, i_p} [\mathbf{C}_{p,p+1}^\tau]_{k'_{p-1}, i'_p, j_{p+1}, k_{p+1}}^{k_{p-1}, i_p, i_{p+1}, k_{p+1}} U_p(k_{p-1}, i_p, k_p) \delta_{l_p, k'_p}, \end{aligned} \quad (\text{C5})$$

$$\begin{aligned} \frac{\partial [\mathbf{C}^0(U_p)]_{l_{p-1} j_p, l_{p+1} j_{p+1}}^{k_{p-1} i_p, k_{p+1} i_{p+1}}}{\partial U_p(k'_{p-1}, i'_p, k'_p)} &= \sum_{l_{p-1}, j_p} [\mathbf{C}_{p,p+1}^0]_{l_{p-1}, j_p, j_{p+1}, k_{p+1}}^{k'_{p-1}, i'_p, i_{p+1}, k_{p+1}} U_p(l_{p-1}, j_p, l_p) \delta_{k_p, k'_p} \\ &+ \sum_{k_{p-1}, i_p} [\mathbf{C}_{p,p+1}^0]_{k'_{p-1}, i'_p, j_{p+1}, k_{p+1}}^{k_{p-1}, i_p, i_{p+1}, k_{p+1}} U_p(k_{p-1}, i_p, k_p) \delta_{l_p, k'_p}. \end{aligned} \quad (\text{C6})$$

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

$$\frac{\partial \hat{\lambda}_m(U_p)}{\partial \mathbf{C}^\tau(U_p)(i,j)} = \mathbf{v}_m(i)\mathbf{v}_m(j)(2 - \delta_{ij}), \quad (\text{C7})$$

$$\frac{\partial \hat{\lambda}_m(U_p)}{\partial \mathbf{C}^0(U_p)(i,j)} = -\hat{\lambda}_m(U_p)\mathbf{v}_m(i)\mathbf{v}_m(j)(2 - \delta_{ij}), \quad (\text{C8})$$

where  $\mathbf{v}_m$  is the  $m$ th eigenvector corresponding to  $\hat{\lambda}_m(U_p)$ . Combining Eqs. (C7) and (C8) with Eqs. (C5) and (C6), we find the derivatives of  $\hat{\lambda}_m(U_p)$  with respect to the variables  $U_p$ . Eqs. (C7) and (C8) can be obtained from perturbation theory. Consider an analytic perturbation of  $\mathbf{C}^\tau = \mathbf{C}^\tau(U_p)$  and  $\mathbf{C}^0 = \mathbf{C}^0(U_p)$ ,

$$\tilde{\mathbf{C}}^\tau = \mathbf{C}^\tau + \epsilon \mathbf{C}_1^\tau + \dots, \quad (\text{C9})$$

$$\tilde{\mathbf{C}}^0 = \mathbf{C}^0 + \epsilon \mathbf{C}_1^0 + \dots. \quad (\text{C10})$$

Then, the proof of Ref. 43 [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  $\mathbf{C}^\tau$ ,  $\mathbf{C}^0$ ,

$$\tilde{\lambda}_m = \hat{\lambda}_m(U_p) + \epsilon \hat{\lambda}_m^1 + \dots. \quad (\text{C11})$$

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

$$\hat{\lambda}_m^1 = (\mathbf{v}_m)^T (\mathbf{C}_1^\tau - \hat{\lambda}_m(U_p)\mathbf{C}_1^0) \mathbf{v}_m. \quad (\text{C12})$$

Eqs. (C7) and (C8) 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  $\mathbf{C}^\tau(U_p)$ ,  $\mathbf{C}^0(U_p)$ .

#### APPENDIX D: 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  $g_{k_p}^{p+1}$ ,

$$f_{k_p} = \sum_{l_{p-1}} \mathbf{u}^{k_p}(l_{p-1}) g_{l_{p-1}}^p, \quad (\text{D1})$$

is found as the solution of the linear system

$$\mathbf{A}^p \mathbf{u}^{k_p} = \mathbf{b}^{k_p}, \quad (\text{D2})$$

$$\mathbf{A}^p(k_{p-1}, l_{p-1}) = \langle g_{k_{p-1}}^p, g_{l_{p-1}}^p \rangle_\pi, \quad (\text{D3})$$

$$\mathbf{b}^{k_p}(k_{p-1}) = \langle g_{k_{p-1}}^p, g_{k_p}^{p+1} \rangle_\pi. \quad (\text{D4})$$

These quantities can be obtained from the correlation matrix  $\mathbf{C}_{p,p+1}^0$  in Eq. (C2). 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

$$\mathbf{b}^{k_p}(k_{p-1}) = \sum_{l_{p-1}, j_p} U_p(l_{p-1}, j_p, k_p) \langle g_{k_{p-1}}^{p-1}, g_{l_{p-1}}^{p-1} f_{j_p}^p \rangle_\pi, \quad (\text{D5})$$

where we have used recursion formula Eq. (35). Next, we can compute the approximation error for  $g_{k_p}^{p+1}$  via

$$(E(p)_{k_p})^2 = \langle g_{k_p}^{p+1} - f_{k_p}, g_{k_p}^{p+1} - f_{k_p} \rangle_\pi \quad (\text{D6})$$

$$= 1 - 2\langle g_{k_p}^{p+1}, f_{k_p} \rangle_\pi + \langle f_{k_p}, f_{k_p} \rangle_\pi \quad (\text{D7})$$

$$= 1 - 2(\mathbf{b}^{k_p})^T \mathbf{u}^{k_p} + (\mathbf{u}^{k_p})^T \mathbf{A}^p \mathbf{u}^{k_p}. \quad (\text{D8})$$

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  $x_p$ .

<sup>1</sup>W. van Gunsteren, J. Dolenc, and A. Mark, "Molecular simulation as an aid to experimentalists," *Curr. Opin. Struct. Biol.* **18**, 149–153 (2008).

<sup>2</sup>C. Schütte, A. Fischer, W. Huisinga, and P. Deuffhard, "A direct approach to conformational dynamics based on hybrid Monte Carlo," *J. Comput. Phys.* **151**, 146–168 (1999).

<sup>3</sup>N. Singhal and V. S. Pande, "Error analysis and efficient sampling in Markovian state models for molecular dynamics," *J. Chem. Phys.* **123**, 204909 (2005).

<sup>4</sup>F. Noé, I. Horenko, C. Schütte, and J. C. Smith, "Hierarchical analysis of conformational dynamics in biomolecules: Transition networks of metastable states," *J. Chem. Phys.* **126**, 155102 (2007).

<sup>5</sup>J. D. Chodera, K. A. Dill, N. Singhal, V. S. Pande, W. C. Swope, and J. W. Pitera, "Automatic discovery of metastable states for the construction of Markov models of macromolecular conformational dynamics," *J. Chem. Phys.* **126**, 155101 (2007).

<sup>6</sup>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," *J. Chem. Phys.* **134**, 174105 (2011).

<sup>7</sup>W. C. Swope, J. W. Pitera, F. Suits, M. Pitman, and M. Eleftheriou, "Describing protein folding kinetics by molecular dynamics simulations. 2. Example applications to alanine dipeptide and beta-hairpin peptide," *J. Phys. Chem. B* **108**, 6582–6594 (2004).

<sup>8</sup>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," *J. Chem. Theory Comput.* **11**, 5525–5542 (2015).

<sup>9</sup>K. A. Beauchamp, G. R. Bowman, T. J. Lane, L. Maibaum, I. S. Haque, and V. S. Pande, "MSMBuilder2: Modeling conformational dynamics at the picosecond to millisecond scale," *J. Chem. Theory Comput.* **7**, 3412–3419 (2011).

<sup>10</sup>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," *Proc. Natl. Acad. Sci. U. S. A.* **106**, 19011–19016 (2009).

<sup>11</sup>A. Berezhkovskii, G. Hummer, and A. Szabo, "Reactive flux and folding pathways in network models of coarse-grained protein dynamics," *J. Chem. Phys.* **130**, 205102 (2009).

<sup>12</sup>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," *J. Phys. Chem. B* **115**, 5415–5424 (2011).

<sup>13</sup>F. Noé, S. Doose, I. Daidone, M. Löllmann, J. D. Chodera, M. Sauer, and J. C. Smith, "Dynamical fingerprints for probing individual relaxation processes in biomolecular dynamics with simulations and kinetic experiments," *Proc. Natl. Acad. Sci. U. S. A.* **108**, 4822–4827 (2011).

- <sup>14</sup>B. Lindner, Z. Yi, J.-H. Prinz, J. C. Smith, and F. Noé, “Dynamic neutron scattering from conformational dynamics. I. Theory and Markov models,” *J. Chem. Phys.* **139**, 175101 (2013).
- <sup>15</sup>K. A. Beauchamp, D. L. Ensign, R. Das, and V. S. Pande, “Quantitative comparison of villin headpiece subdomain simulations and triplet–triplet energy transfer experiments,” *Proc. Natl. Acad. Sci. U. S. A.* **108**, 12734–12739 (2011).
- <sup>16</sup>G. R. Bowman and P. L. Geissler, “Equilibrium fluctuations of a single folded protein reveal a multitude of potential cryptic allosteric sites,” *Proc. Natl. Acad. Sci. U. S. A.* **109**, 11681–11686 (2012).
- <sup>17</sup>K. A. Beauchamp, R. McGibbon, Y. S. Lin, and V. S. Pande, “Simple few-state models reveal hidden complexity in protein folding,” *Proc. Natl. Acad. Sci. U. S. A.* **109**, 17807–17813 (2012).
- <sup>18</sup>I. Buch, T. Giorgino, and G. de Fabritiis, “Complete reconstruction of an enzyme-inhibitor binding process by molecular dynamics simulations,” *Proc. Natl. Acad. Sci. U. S. A.* **108**, 10184–10189 (2011).
- <sup>19</sup>S. K. Sadiq, F. Noé, and G. de Fabritiis, “Kinetic characterization of the critical step in HIV-1 protease maturation,” *Proc. Natl. Acad. Sci. U. S. A.* **109**, 20449–20454 (2012).
- <sup>20</sup>M. Sarich, F. Noé, and C. Schütte, “On the approximation quality of Markov state models,” *SIAM Multiscale Model. Simul.* **8**, 1154–1177 (2010).
- <sup>21</sup>F. Noé and F. Nüske, “A variational approach to modeling slow processes in stochastic dynamical systems,” *SIAM Multiscale Model. Simul.* **11**, 635–655 (2013).
- <sup>22</sup>F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, “Variational approach to molecular kinetics,” *J. Chem. Theory Comput.* **10**, 1739–1752 (2014).
- <sup>23</sup>F. Vitalini, F. Noé, and B. G. Keller, “A basis set for peptides for the variational approach to conformational kinetics,” *J. Chem. Theory Comput.* **11**, 3992–4004 (2015).
- <sup>24</sup>I. Oseledets and E. Tyrtshnikov, “Breaking the curse of dimensionality, Or how to use SVD in many dimensions,” *SIAM J. Sci. Comput.* **31**, 3744–3759 (2009).
- <sup>25</sup>I. Oseledets, “Tensor-train decomposition,” *SIAM J. Sci. Comput.* **33**, 2295–2317 (2011).
- <sup>26</sup>S. Ostlund and S. Rommer, “Thermodynamic limit of density matrix renormalization,” *Phys. Rev. Lett.* **75**, 3537–3540 (1995).
- <sup>27</sup>S. R. White, “Density matrix formulation for quantum renormalization groups,” *Phys. Rev. Lett.* **69**, 2863–2866 (1992).
- <sup>28</sup>S. Szalay, M. Pfeffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, and Ö. Legeza, “Tensor product methods and entanglement optimization for *ab initio* quantum chemistry,” *Int. J. Quantum Chem.* **115**, 1342–1391 (2015).
- <sup>29</sup>W. Hackbusch and S. Kühn, “A new scheme for the tensor representation,” *J. Fourier Anal. Appl.* **15**, 706–722 (2009).
- <sup>30</sup>W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus* (Springer, Berlin Heidelberg, 2012).
- <sup>31</sup>S. Holtz, T. Rohwedder, and R. Schneider, “The alternating linear scheme for tensor optimization in the tensor train format,” *SIAM J. Sci. Comput.* **34**, A683–A713 (2012).
- <sup>32</sup>J. Chodera, W. Swope, J. Pitner, and K. Dill, “Long-time protein folding dynamics from short-time molecular dynamics simulations,” *SIAM Multiscale Model. Simul.* **5**, 1214–1226 (2006).
- <sup>33</sup>K. Fan, “On a theorem of Weyl concerning eigenvalues of linear transformations,” *Proc. Natl. Acad. Sci. U. S. A.* **35**, 652–655 (1949).
- <sup>34</sup>A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry* (Dover Publications, Mineola, NY, 1982).
- <sup>35</sup>H. Wu and F. Noé, “Gaussian Markov transition models of molecular kinetics,” *J. Chem. Phys.* **142**, 084104 (2015).
- <sup>36</sup>G. Perez-Hernandez, F. Paul, T. Giorgino, G. de Fabritiis, and F. Noé, “Identification of slow molecular order parameters for Markov model construction,” *J. Chem. Phys.* **139**, 015102 (2013).
- <sup>37</sup>C. R. Schwantes and V. S. Pande, “Improvements in Markov state model construction reveal many non-native interactions in the folding of NTL9,” *J. Chem. Theory Comput.* **9**, 2000–2009 (2013).
- <sup>38</sup>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,” *Comput. Phys. Commun.* **185**, 1207–1216 (2014).
- <sup>39</sup>D. Kressner, M. Steinlechner, and A. Uschmajew, “Low-rank tensor methods with subspace correction for symmetric eigenvalue problems,” *SIAM J. Sci. Comput.* **36**, A2346–A2368 (2014).
- <sup>40</sup>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).
- <sup>41</sup>F. Noé, H. Wu, J.-H. Prinz, and N. Plattner, “Projected and hidden Markov models for calculating kinetics and metastable states of complex molecules,” *J. Chem. Phys.* **139**, 184114 (2013).
- <sup>42</sup>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,” *J. Comput. Chem.* **24**, 1999–2012 (2003).
- <sup>43</sup>J. R. Magnus, “On differentiating eigenvalues and eigenvectors,” *Econometric Theory* **1**, 179–191 (1985).