The Journal of Chemical Physics ARTICLE pubs.aip.org/aip/jcp

Tensor-SqRA: Modeling the transition rates of interacting molecular systems in terms of potential energies

Cite as: J. Chem. Phys. 160, 104112 (2024); doi: 10.1063/5.0187792
Submitted: 16 November 2023 • Accepted: 19 February 2024 •
Published Online: 14 March 2024

Alexander Sikorski,[1,2,a] Amir Niknejad,[3,b] Marcus Weber,[2,c] and Luca Donati[1,2,d]

AFFILIATIONS

[1] Department of Mathematics and Computer Science, Freie Universität Berlin, 14195 Berlin, Germany
[2] Department Modeling and Simulation of Complex Processes, Zuse Institute Berlin, 14195 Berlin, Germany
[3] Department of Mathematics, College of Mount Saint Vincent, New York, New York 10471, USA

[a] **Author to whom correspondence should be addressed:** sikorski@zib.de
[b] amir.niknejad@mountsaintvincent.edu
[c] weber@zib.de
[d] donati@zib.de

ABSTRACT

Estimating the rate of rare conformational changes in molecular systems is one of the goals of molecular dynamics simulations. In the past few decades, a lot of progress has been done in data-based approaches toward this problem. In contrast, model-based methods, such as the Square Root Approximation (SqRA), directly derive these quantities from the potential energy functions. In this article, we demonstrate how the SqRA formalism naturally blends with the tensor structure obtained by coupling multiple systems, resulting in the tensor-based Square Root Approximation (tSqRA). It enables efficient treatment of high-dimensional systems using the SqRA and provides an algebraic expression of the impact of coupling energies between molecular subsystems. Based on the tSqRA, we also develop the projected rate estimation, a hybrid data-model-based algorithm that efficiently estimates the slowest rates for coupled systems. In addition, we investigate the possibility of integrating low-rank approximations within this framework to maximize the potential of the tSqRA.

I. INTRODUCTION

Classical molecular systems are modeled by a function $\widetilde{V}(x)$, which provides the potential energy of the system as a function of the D-dimensional coordinate vector x of the atoms of the system. The potential energy function $\widetilde{V}$ is calculated as a sum of interaction energies, and each summand depends on a small subset of the coordinates only. If we assume that parts of the molecular system are spatially far enough apart from each other, then these parts of the system move almost independently. According to this assumption, let the potential energy of the system, e.g., be written as

$$\begin{aligned}\widetilde{V}(x) := \widetilde{V}(x_1, x_2) &= V_1(x_1) + V_2(x_2) + V_c(x_1, x_2) \\ &= V(x_1, x_2) + V_c(x_1, x_2),\end{aligned} \tag{1}$$

where x_1 is a d-dimensional vector and x_2 is a $(\mathrm{D} - d)$-dimensional vector of disjoint subsets of coordinates of the entire D-dimensional system. The potential defined in Eq. (1) can be investigated from three different points of view: (i) V_1 and V_2 are analyzed individually as *isolated subsystems*; (ii) V_1 and V_2 are analyzed as a non-interacting *combined system*; and (iii) V_1 and V_2 interact by means of the potential V_c giving rise to a *coupled system*. In real applications, it is interesting to understand how the term V_c of the coupled system changes the rate of rare transitions of the combined system.

In order to answer this fundamental question, the typical approach (known as Markov state modeling[1]) is to produce classical Molecular Dynamics (MD) simulations at the atomistic level; then, the simulation data are used to construct transition probability

matrices, whose spectral analysis allows for the determination of the time scales of the "macroscopic movements."

As an alternative to the data-based approach of Markov state modeling, which is widely known and applied, the model-based Square Root Approximation (SqRA)[2–5] can directly derive transition rate matrices from the potential energy function of the molecular system without taking the detour of generating molecular simulation data.

In this article, we review the fundamentals of SqRA and show how its algebraic structure can be leveraged to calculate the rare event rates of molecular systems with potentials defined as in Eq. (1). In particular, we show how SqRA allows us to directly represent the kinetic properties of the non-interacting system $V(x_1, x_2)$ in terms of the rate matrices of the isolated subsystems $V_1(x_1)$ and $V_2(x_2)$ using the Kronecker formalism,[6] thereby circumventing the curse of dimensionality. Clearly, this simple decomposition breaks down when introducing the coupling term $V_c(x_1, x_2)$.

To alleviate this problem, we developed a tensor formulation of SqRA (tSqRA), which allows us to represent the coupling terms by Hadamard products. Since these act on the interacting particles only, this enables to inherit as much of the underlying decoupled structure as possible. This reduces the complexity while still providing exact results, avoiding the (computationally expensive) production of molecular simulation data to obtain transition rate matrices of the entire molecular system. Using a tensor algebra to analyze coupled Markov processes is not a new idea; there exist introductory texts to these kinds of approaches; see the work of Dayar[7] or Ludyk.[8] In addition, analytical and linear algebraic methods are well-established and widely known; see the work of Pollock[9] and Jokar and Mehrmann.[10] Not only in theory, but also in applications, these methods have been used intensively; see the work of Fernández[11] and Ching *et al.*[12]

However, our article goes one step further than the existing theory and methods by combining the algebraic form of SqRA, which provides an explicit link between transition rates and potential energy with the tensor algebra approach. This offers an interpretation of the kinetics of the coupled system as a perturbation of isolated subsystems and allows us to apply perturbation methods. In order to illustrate this idea, we introduce the Projected Rate Estimation (PRE), a hybrid data-model-based method to obtain the global transition rates of coupled systems as perturbation of the transition rates of non-interacting subsystems by carrying out only a few local simulations.

The presented methods do not exploit any low-rank structures in the problem yet and therefore cannot be applied directly to larger systems. However, we demonstrate how the provided formalism lends itself to future low-rank approximations via tensor-trains or tensor-networks, thereby facilitating computations on larger scales.[13–16]

II. THEORY

Originally, SqRA[2,17] was invented to solve the problem of finding a transition rate matrix $Q \in \mathbb{R}^{N\times N}$ of a molecular system with an Euclidean state space discretized into N subsets such that this matrix is reversible and has a predefined stationary distribution $\pi \in \mathbb{R}^N$. It turned out that a regular grid discretization of the state space guarantees a simple algebraic form of the matrix Q and that increasing the size of the grid $M \to \infty$ leads to a convergence of Q toward the Fokker–Planck operator of the underlying overdamped Langevin dynamics of the system.[18] Note that for the application of SqRA, a regular grid can be constructed by a hypercubic grid, which can be seen as a Voronoi tessellation with equidistant center points in each direction. In many applications of SqRA, it is assumed that the discretization of the state space is fine enough to approximate the continuous stationary distribution by the pointwise evaluation of the Boltzmann distribution at the centers of the cells. In Secs. II A–II D, we will show how the algebraic structure of the SqRA lends itself to the application to combined and even coupled systems.

In Sec. II A, we start by introducing the necessary foundations of the SqRA. Section II B then shows how the SqRA matrices for the combined system can be composed of the SqRA matrices of the isolated systems using Kronecker products and sums. In Sec. II C, we show how these matrices are changed by the introduction of a coupling between the subsystems. Section II D provides the most general formulation in terms of tensors and Hadamard products paving the way for future low-rank developments.

A. Linear algebraic form of SqRA

Consider a D-dimensional dynamical system governed by the potential energy function $V(x) : \mathbb{R}^D \to \mathbb{R}$, where the state space x is discretized by a hypercubic grid with N center points x_i with $i = 1, 2, \ldots, N$. The stationary distribution π is given by the Boltzmann distribution with entries

$$\pi_i := \pi(x_i) = \exp\left(-\frac{1}{k_B T} V(x_i)\right), \tag{2}$$

where k_B is the Boltzmann constant and T is the temperature. The SqRA defines the sparse rate matrix $Q \in \mathbb{R}^{N\times N}$ as

$$Q_{ij} = \begin{cases} \sqrt{\pi_j/\pi_i} & \text{if } x_i \text{ is adjacent to } x_j, \\ -\sum_{\substack{i=1 \\ i\neq j}}^{M} Q_{ij} & \text{if } x_i = x_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

or alternatively in the matrix form

$$\begin{aligned} Q &:= D^{-1}AD - \operatorname{diag}\left(D^{-1}AD\, e\right) \\ &= D^{-1}AD - E \\ &= Q^{\text{out}} - E. \end{aligned} \tag{4}$$

Here, D is a diagonal matrix with the square roots of the vector π on the diagonal, i.e., $D = \operatorname{diag}(\sqrt{\pi})$, A is the adjacency matrix of the grid, and $e_M = (1, 1, \ldots, 1)^T$ is a column vector with N entries. We denote the off-diagonal part of Q by $Q^{\text{out}} = D^{-1}AD$, and the diagonal part $E = \operatorname{diag}(D^{-1}ADe_M)$ is chosen in such a way that Q has row sum zero. Note that the stationary distribution in Eq. (2) does not need to be normalized since the quotient in the SqRA's formula cancels out the normalization constant. When defining Q this way, it has unit-less entries. To obtain a proper rate matrix, it is necessary to multiply each entry Q_{ij} by a flux term Φ_{ij} that depends on the grid geometry and the diffusion.[4] For simplicity, we assume a regular grid and constant diffusion such that we can omit $\Phi \propto 1\ (\text{time units})^{-1}$.

With this assumption, the matrix Q is a transition rate matrix. The rows of Q^{out} comprise the outgoing rates from one given state (subset) of the system to the adjacent states, and the diagonal entries of Q are negative and represent the "total exit rate" from a given state. Since these are also the entries of E, we will also refer to E as the "exit rate matrix." Note that Q can be seen as a similarity transform of $A - E$,

$$Q = D^{-1}(A - E)D. \tag{5}$$

Thus, eigenvalues of Q and $A - E$ coincide. As a consequence, the implied time-scales of molecular systems, which are derived from these eigenvalues, only depend on the "total exit rates" and on the adjacency of the discretized states of the system.

In this regard, $A - E$ is like a decomposition of the process into a "entropical" part (adjacency matrix A) and a "energetic" part (exit rate matrix E). The eigenvectors (or their sign structure) of Q on the other hand can be used to identify macro-states in the form of conformations or metastabilities by algorithms such as PCCA[19] or PCCA+.[20] If v is an eigenvector of $A - E$, then $D^{-1}v$ is the corresponding eigenvector of Q (sharing the same sign structure with v).

Describing transitions in terms of rate matrices is only one possible notation. A very common way of analyzing Markov processes is the use of transition probability matrices instead of rate matrices. Our approach can be transferred to the framework of transition matrices. A Markov chain that is observed for a specific lag time τ [the time unit is defined by the flux in Eq. (5)] is represented by a conditional probability matrix

$$K^{\tau} = \exp(\tau Q), \tag{6}$$

where $\exp(\cdot)$ denotes the matrix exponential. Equation (6) defines the discrete version of the Koopman operator that transports observable functions $f \in L^{\infty}$ forward in time. Correspondingly, its transpose is a discretization of the propagator P^{τ}, which transports probability densities $\rho \in L^{1}$. The entries of the ith row of K^{τ} quantify the conditional probability for a system that starts in this state i to end up in the respective states in the time span τ. In contrast to Q, the matrix K^{τ} is usually a dense matrix. However, K^{τ} has the same eigenvectors, such as Q, and if λ is an eigenvalue of Q, then $\exp(\tau\lambda)$ is an eigenvalue of K^{τ}. In this regard, many results and methods developed for K^{τ} or P^{τ} directly carry over to Q.

B. The Kronecker formalism of SqRA for combined subsystems

When considering two isolated systems consisting of N_1 and N_2 possible states, respectively, the combined system can be in one of $N = N_1 \times N_2$ possible states and the corresponding Q matrix is of size $N_1 \cdot N_2 \times N_1 \cdot N_2$. However, since they do not interact, they should still be described by the two individual system's rates matrices Q_1, Q_2, which are of sizes $N_1 \times N_1$ and $N_2 \times N_2$ only, therefore providing a way more compact representation. In this section, we will show how the SqRA allows for such a compact representation in the form of Kronecker products and sums.

We first consider the case of $n = 2$ combined subsystems, respectively, of dimensions D_1 and D_2 such that $D_1 + D_2 = D$, before stating the more general result for arbitrary n. The combined potential energy is the sum of the subsystems potentials $V_1 : \mathbb{R}^{D_1} \to \mathbb{R}$ and $V_2 : \mathbb{R}^{D_2} \to \mathbb{R}$,

$$V(x) := V(x_1, x_2) = V_1(x_1) + V_2(x_2), \tag{7}$$

where $x_1 \in \mathbb{R}^{D_1}$ and $x_2 \in \mathbb{R}^{D_2}$ are the state vectors of the subsystems. Similarly, due to its exponential form, the overall stationary distribution defined in Eq. (2) factors into the product of the marginal distributions of the subsystems

$$\pi(x) := \pi(x_1, x_2) = \pi_1(x_1)\pi_2(x_2). \tag{8}$$

The hyper-cubic grid discretizing the Euclidean state space is given by the combination of the grids along the two system's sets of coordinates, each consisting of N_1 and N_2 cells, respectively, for a total number of $N = N_1 \cdot N_2$ cells. Let us now see how the matrices A, D, E, Q^{out}, and Q introduced in Eq. (4) are rewritten in terms of the smaller matrices associated with the two subsystems.

1. Adjacency matrix

The matrix A of the coupled system is given by the Kronecker sum of the corresponding adjacency matrices of the subsystems,

$$\begin{aligned} A &= A_1 \oplus A_2 \\ &= A_1 \otimes I_2 + I_1 \otimes A_2, \end{aligned} \tag{9}$$

where I_1 and I_2 are, respectively, two identity matrices of sizes $(N_1 \times N_1)$ and $(N_2 \times N_2)$.

2. Diagonal matrix

The stationary distributions π_1 and π_2 of the subsystems are used to build the diagonal matrices $D_1 = \text{diag}(\sqrt{\pi_1})$ and $D_2 = \text{diag}(\sqrt{\pi_2})$, and the matrix D is given by Kronecker product of the corresponding subsystems matrices,

$$D = D_1 \otimes D_2. \tag{10}$$

3. Off-diagonal matrix

Inserting Eqs. (9) and (10) into the off-diagonal part of the SqRA [Eq. (4)], we obtain

$$\begin{aligned} Q^{\text{out}} = D^{-1}AD &= (D_1^{-1} \otimes D_2^{-1})(A_1 \otimes I_2 + I_1 \otimes A_2)(D_1 \otimes D_2) \\ &= (D_1^{-1} \otimes D_2^{-1})(A_1 \otimes I_2)(D_1 \otimes D_2) \\ &\quad + (D_1^{-1} \otimes D_2^{-1})(I_1 \otimes A_2)(D_1 \otimes D_2) \\ &= (D_1^{-1}A_1 \otimes D_2^{-1}I_2)(D_1 \otimes D_2) \\ &\quad + (D_1^{-1}I_1 \otimes D_2^{-1}A_2)(D_1 \otimes D_2) \\ &= D_1^{-1}A_1D_1 \otimes I_2 + I_1 \otimes D_2^{-1}A_2D_2, \end{aligned}$$

and we see that the off-diagonal part indeed decomposes into the Kronecker sum of the individual systems,

$$Q^{\text{out}} = Q_1^{\text{out}} \oplus Q_2^{\text{out}}. \tag{11}$$

4. Rate matrix

Since transition probabilities are given by the product probabilities for the respective transitions of the subsystems, the discretization of the Koopman operator of the full system can be decomposed as[7]

$$\begin{aligned} K^\tau &= K_1^\tau \otimes K_2^\tau \\ &= \exp(\tau Q_1) \otimes \exp(\tau Q_2) \\ &= \exp(\tau(Q_1 \oplus Q_2)), \end{aligned} \tag{12}$$

where we applied the Kronecker sum rule for exponential matrices in the last line. This implies

$$Q = Q_1 \oplus Q_2. \tag{13}$$

5. Exit rate matrix

Applying Eqs. (11) and (13) yields

$$\begin{aligned} Q &= Q_1 \oplus Q_2 \\ &= (Q_1^{\text{out}} - E_1) \oplus (Q_2^{\text{out}} - E_2) \\ &= (Q_1^{\text{out}} - E_1) \otimes I_2 + I_1 \otimes (Q_2^{\text{out}} - E_2) \\ &= Q_1^{\text{out}} \oplus Q_2^{\text{out}} - E_1 \oplus E_2, \end{aligned}$$

from which one derives

$$E = E_1 \oplus E_2. \tag{14}$$

For an arbitrary number n of subsystems, we summarize this finding in the following proposition:

Proposition 1. Consider a D*-dimensional system with potential* $V : \mathbb{R}^{\text{D}} \to \mathbb{R}$ *and stationary distribution* $\pi : \mathbb{R}^{\text{D}} \to \mathbb{R}$ *defined on a state space that can be discretized by a* D*-dimensional hypercubic grid.*

Assume that the coordinates can be partitioned into n subsets of sizes $\text{D}_1, \text{D}_2, \ldots, \text{D}_n$ *each, with* $\sum_{i=1}^n \text{D}_i = \text{D}$*, such that the potential V is written as*

$$V(x) = \sum_{i=1}^{n} V_i(x_i),$$

where each potential $V_i : \mathbb{R}^{\text{D}_i} \to \mathbb{R}$ is a D_i-dimensional function that depends only on the ith subset of coordinates.

Correspondingly, the stationary distribution is rewritten as

$$\pi(x) = \prod_{i=1}^{n} \pi_i(x_i).$$

For each subsystem i, we define the adjacency matrix A_i*, the diagonal matrix* $D_i = \text{diag}\left(\pi_i^{1/2}\right)$*, the off-diagonal rate matrix* $Q_i^{\text{out}} = D_i^{-1} A_i D_i$*, the exit rate matrix* $E_i = \text{diag}\left(Q_i^{\text{out}} e_{M_i}\right)$*, as well as the rate matrix* $Q_i = Q_i^{\text{out}} - E_i$*. Then, the SqRA matrices for the entire system are given in terms of the subsystem matrices,*

$$\boxed{\begin{aligned} A &= \bigoplus_{i=1}^{n} A_i, \\ D &= \bigotimes_{i=1}^{n} D_i, \\ Q^{\text{out}} &= \bigoplus_{i=1}^{n} Q_i^{\text{out}} = D^{-1} A D, \\ E &= \bigoplus_{i=1}^{n} E_i, \\ Q &= \bigoplus_{i=1}^{n} Q_i = Q^{\text{out}} - E. \end{aligned}}$$

C. Matrix representation of the SqRA with a global coupling term

Section II B showed that the structure of the SqRA naturally leads to a low-rank representation for the coupled case. We will now study how the results change when adding a coupling between the systems.

As in Sec. II B, we start with the case of $n = 2$ subsystems first. Given the coupling potential V_c, the total potential energy of the coupled system is

$$\widetilde{V}(x_1, x_2) = V_1(x_1) + V_2(x_2) + V_C(x_1, x_2). \tag{15}$$

The unnormalized stationary distribution is decomposed as

$$\widetilde{\pi}(x_1, x_2) = \pi_1(x_1)\pi_2(x_2)\,\pi_C(x_1, x_2), \tag{16}$$

where $\pi_C(x_1, x_2) = \exp(-1/k_B T V_C(x_1, x_2))$. The adjacency relations are not affected by the coupling term, so the adjacency matrix A is the same as in Eq. (9). On the other hand, each entry of the diagonal matrix D is reweighted by the coupling, leading to

$$\widetilde{D} = D_C\,(D_1 \otimes D_2), \tag{17}$$

where $D_C = \text{diag}\left(\pi_C^{1/2}\right)$ is the $(N \times N)$ diagonal matrix built with the stationary distribution of the coupling potential V_C, while D_1 and D_2 are defined as for the combined system. Similarly, the inverse satisfies

$$\widetilde{D}^{-1} = D_C^{-1}\left(D_1^{-1} \otimes D_2^{-1}\right). \tag{18}$$

Note that, given that D_C is a diagonal matrix, the calculation in Eq. (17) can be interpreted as either a matrix–matrix multiplication or an elementwise multiplication. This section focuses on the matrix formalism, but the elementwise interpretation will play a central role in Sec. II D.

The off-diagonal matrix $\widetilde{Q}^{\text{out}}$ of the coupled system is written as

$$\begin{aligned} \widetilde{Q}^{\text{out}} &= \widetilde{D}^{-1} A \widetilde{D} \\ &= D_C^{-1}\left(D_1^{-1} \otimes D_2^{-1}\right)(A_1 \oplus A_2) D_C\,(D_1 \otimes D_2) \\ &= D_C^{-1}\left(Q_1^{\text{out}} \oplus Q_2^{\text{out}}\right) D_C \\ &= D_C^{-1} Q^{\text{out}} D_C, \end{aligned} \tag{19}$$

and thus also $\widetilde{E} = \mathrm{diag}\left(D_C^{-1} Q^{\mathrm{out}} D_C\, e_M\right)$. According to Eq. (4), the SqRA rate matrix for the coupled system then reads

$$\begin{aligned}\widetilde{Q} &= \widetilde{D}^{-1} A \widetilde{D} - \widetilde{E} \\ &= D_C^{-1} Q^{\mathrm{out}} D_C - \widetilde{E}.\end{aligned} \tag{20}$$

In the general case for n subsystems coupled by a single coupling term, results of Proposition 1 are modified as follows.

Proposition 2. Consider the situation of Proposition 1 but with a global coupling potential $V_C : \mathbb{R}^n \to \mathbb{R}$. The potential thus decomposes into

$$\widetilde{V}(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} V_i(x_i) + V_C(x_1, x_2, \ldots, x_n), \tag{21}$$

with additional coupling term $V_C : \mathbb{R}^n \to \mathbb{R}$, and the stationary distribution factorizes as

$$\widetilde{\pi}(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} \pi_i(x_i) \cdot \pi_C(x_1, x_2, \ldots, x_n). \tag{22}$$

The SqRA matrices for the coupled system are then given by

$$\boxed{\begin{aligned} D_C &= \mathrm{diag}\left[\sqrt{\pi_C}\right], \\ \widetilde{Q}^{\mathrm{out}} &= D_C^{-1} Q^{\mathrm{out}} D_C, \\ \widetilde{E} &= \mathrm{diag}\left[D_C^{-1} Q^{\mathrm{out}} D_C\, e_M\right], \\ \widetilde{Q} &= \widetilde{Q}^{\mathrm{out}} - \widetilde{E}. \end{aligned}} \tag{23}$$

D. Generalization to the tensor formulation for arbitrary interactions

Using the tensor formalism, previous results can be easily generalized to the case where the potential $\widetilde{V}$ is given by a sum of lower order potentials, i.e., potentials that act only on a subset of coordinates, leading to a flexible decomposition in terms of Hadamard products and paving the way for low-rank tensor computations.

To this end, let us consider each coordinate as an individual subsystem, i.e., $n = \mathrm{D}$, and introduce the space of tensors of order D, $T(\mathrm{D}) = \mathbb{R}^{N_1 \times \cdots \times N_{\mathrm{D}}}$. Each individual state of the (discretized) system can be understood as a single entry of this tensor; then, elements $x \in T(\mathrm{D})$ represent distributions or functions over all states. We furthermore introduce the symbolic multi-indices $I \in \mathcal{J} \subset \mathcal{P}(\{1, \ldots, N\})$, where $\mathcal{P}$ denotes the power set, i.e., the set of all possible subsets of indices that can appear. We use these multi-indices in subscript to denote the coordinates (modes) upon which the individual lower-order potential contributions V_I depend,

$$\widetilde{V} = \sum_{I \in \mathcal{J}} V_I(x_I). \tag{24}$$

Furthermore, we use Greek superscript letters to denote the individual grid positions (indices) of those respective coordinates. For example, the tensor $V_{1,2} \in \mathbb{R}^{N_2 \times N_3}$ of order 2 with components $V_{1,2}^{\alpha\beta} = V_{1,2}(x^{\alpha\beta})$, $\alpha = 1, \ldots, N_1, \beta = 1, \ldots, N_2$, holds the evaluation of all potential contributions of the combinations of first and second coordinates at the respective product grid. To each set of indices I corresponds a tensor D_I of order $|I|$ consisting of the square roots of the stationary distribution [Eq. (2)] of the corresponding potential

$$D_I^{\alpha\beta\gamma\cdots} = \exp\left(-\frac{1}{2}\frac{1}{k_B T} V_I^{\alpha\beta\gamma\cdots}\right). \tag{25}$$

This tensor holds all the information about the interaction between the indices I. We combine the individual interactions to the whole interaction tensor D of order D by means of the (widened) *Hadamard product*, which is composed of the elements selected by their shared indices along their shared modes, for example, for two three-tensors sharing two modes, we obtain the four-tensor,

$$\left(a_{ijk} \odot b_{jkl}\right)^{\alpha\beta\gamma\delta} := a_{ijk}^{\alpha\beta\gamma} \cdot b_{jkl}^{\beta\gamma\delta}, \tag{26}$$

where we "broadcast" or "widen" the elementwise multiplication along all dimensions appearing only on one side. It can also be understood as the element-wise product of the two tensors after padding them to the same shape, essentially by replicating/copying them along their non-shared modes. The SqRA tensor $D \in T(\mathrm{D})$ then decomposes into the Hadamard factors corresponding to the respective lower order potentials,

$$D_{ijkl\ldots} = D_i \odot D_j \odot \cdots \odot D_{ij} \odot \cdots \odot D_{ijkl\ldots}, \tag{27}$$

or shortly

$$D = \bigodot_{I \in \mathcal{J}} D_I. \tag{28}$$

In Secs. II A and II B, we introduced the adjacency matrix A. In the case of a one-dimensional system, it consists of a sparse matrix with two off-diagonals. For the flattened representation of higher-dimensional systems, it is a multibanded-matrix with 2D bands. However, the *Kronecker sum* representation defined in Eq. (9) directly translates to a tensor representation with entries

$$\begin{aligned} A^{\alpha\beta\gamma\ldots,\alpha'\beta'\gamma'\cdots} &= A_1^{\alpha\alpha'} \oplus A_2^{\beta\beta'} \oplus \ldots \\ &= A_1^{\alpha\alpha'} \delta^{\beta\beta'} \delta^{\gamma\gamma'} \ldots + \delta^{\alpha\alpha'} A_2^{\beta\beta'} \delta^{\gamma\gamma'} \ldots + \cdots, \end{aligned} \tag{29}$$

with δ being the usual Kronecker delta: $\delta^{\alpha\alpha'} = 1$ if $\alpha = \alpha'$ and 0 otherwise. In this regard, we will think of A as a linear map, mapping tensors of order D to tensors of order D. The action of the SqRA tensor $Q : T(\mathrm{D}) \to T(\mathrm{D})$ on a state $x \in T(\mathrm{D})$ can then be computed via

$$Qx = D^{-1} \odot A(D \odot x) - E \odot x, \tag{30}$$

where $E = D^{-1} A(D) \in T(\mathrm{D})$.

Let us now discuss the practical implications of this formulation in terms of computational complexity. Let N denote the size of a state vector in $T(\mathrm{D})$. The action of the adjacency operator A on a tensor state $x \in T(\mathrm{D})$ can be computed by $2\mathrm{D}N$ floating point operations. The regular grid leads to a banded matrix-representation for flattened states, which allows for a very cache-efficient implementation (cf. Appendix B). Since $D \in T(\mathrm{D})$, it requires just as much memory as we need to hold the state x in memory. Note here that when computing D according to Eq. (28), we evaluate the potential functions only on grids up to the order of the interaction.

For example, consider a system of D particles in 1D space with pairwise interactions and a grid of m cells for each particle. Using the tensor representation [Eq. (28)], each of the $D \cdot (D-1)$ pairwise potentials gets evaluated on a grid of the size m^2, resulting in $O(D^2 m^2)$ potential evaluations, compared to $O(D^2 m^D)$ evaluations for a naïve application on the whole grid. Similarly, a system of $p = D/3$ particles in three-dimensional space with bond (pair-wise) and angle (triplet-wise) interactions, discretized to m cells in each of the $D = 3p$ coordinates, requires $O(p^3 m^3)$ evaluations instead of $O(p^3 m^D)$.

Put differently, let d denote the order of the highest order interactions and assume that the number of these interactions is fixed. The computational effort for computing the D tensor then scales with $O(m^d)$ compared to $O(m^D)$ for the classical evaluation on the whole grid, i.e., it does not depend on the full system dimension. Note, however, that the number of cells $N = m^D$ still grows exponentially in the dimension. Therefore, when using a dense state representation, the application of D still scales badly.

However, since each D_I acts only on a few modes, low-rank representations of the state, in conjunction with approximate low-rank computations of the Hadamard product, allow for a closed low-rank representation of Q. In cases where the dynamics do indeed permit low-rank representations, as would be expected for weakly interacting systems, this method promises to break the curse of dimensionality (see also the discussion in Sec. V).

Finally, note that for the iterated application, just as with the similarity transform in the matrix case, we have

$$Q^n(x) = D^{-1} \odot (A - E)^n (D \odot x). \tag{31}$$

Thus, in order to compute the spectrum of Q by an iterative solver, we merely need the repeated evaluation of $A - E$, resulting in $O((2D+1)N)$ floating point operations. Considering that A is inherently low-rank [cf. Eq. (9)], using a low-rank state representation promises an exponential speedup.

To summarize, using the tensor SqRA (denoted as tSqRA) allows us to alleviate the (explicit) exponential dependence of the computation cost for Q in the system dimension, replacing it by the maximal order of interactions. Even though the scalability is still influenced by the state-size (implying an exponential relationship for dense representation), the tensor formalism should naturally facilitate the integration of low-rank techniques to manage this aspect.

E. Coupling of the eigenvalues and the eigenvectors

We noted previously that the eigenvalues and eigenvectors of Q are of special interest to understand the slow time-scale dynamics of the process. Let us therefore investigate what we can say about the spectrum of the composed and coupled system. Since Q is similar to $A - E$, which in turn is symmetric, it follows that all eigenvalues are real numbers. Furthermore, the leading eigenvalue of Q matrices is 0 (with a constant eigenvector) and all other eigenvalues are negative.

Let $0 = \lambda_i^{(1)} > \lambda_i^{(2)} \geq \cdots \geq \lambda_i^{(N_i)}$ denote the eigenvalues with respective eigenfunctions $X_i^{(1)}, \ldots, X_i^{(N_i)}$ of the rate matrix Q_i of the ith isolated subsystem. Defining $\Lambda_i = \mathrm{diag}(\lambda_i)$ such that $Q_i X_i = \Lambda_i X_i$, for the combined system Q, we have

$$QX = \left(\bigoplus_i Q_i\right)\left(\bigotimes_i X_i\right) = \left(\bigoplus_i \Lambda_i\right)\left(\bigotimes_i X_i\right) = \Lambda X, \tag{32}$$

where $\Lambda := \bigoplus_i \Lambda_i$ is a diagonal matrix with eigenvalues as diagonal entries and $X := \bigotimes_i X_i$ is a matrix with the eigenvectors of the combined system. In terms of individual eigenvalues $\lambda^{(j)}$, indexed by the multi-index $j \in \{1, \ldots, N_1\} \times \cdots \times \{1, \ldots, N_n\}$, this comes down to the eigenvalues being a sum of one of each of the isolated system's eigenvalues,

$$\lambda^{(j)} = \sum_{i=1}^{n} \lambda_i^{j(i)}, \tag{33}$$

where $j(i)$ is the corresponding index to the eigenvalue of subsystem i. The corresponding eigenvector $X^{(j)}$ is then given by the product of the respective subsystem's eigenvectors,

$$X^{(j)} = \bigotimes_{i=1}^{n} X_{(i)}^{j(i)}. \tag{34}$$

When introducing a coupling term, V_C, the spectrum is perturbed from $\lambda^{(j)}$ to $\widetilde{\lambda}^{(j)}$ as well as from $X^{(j)}$ to $\widetilde{X}^{(j)}$.

The easiest case is the coupling of two subsystems. One can observe a repeating algebraic pattern comparing the construction of the rate matrix Q, from the adjacency matrix A, with the construction of coupled systems from uncoupled systems. In Eq. (4), we have a conversion from A, which represents transition rates with regard to a constant potential energy function into a matrix Q of exit rates with regard to a non-constant potential V. Interestingly, this is the same kind of linear algebra like the transition from the positive exit rates of the combined system (matrix Q^{out}) with a constant coupling energy to the rate matrix of the coupled system $\widetilde{Q}$ with a non-constant V_I. It is given by a similarity transform of a positive rate matrix using a diagonal matrix followed by subtracting the diagonal matrix of row sums. Although this is a simple linear algebraic correspondence between the matrices, it also shows that a transition from Q to $\widetilde{Q}$ can change the solution of the eigenproblem largely, as the transition from a pure adjacency A to a molecular system (by regarding the potential V) does. However, by the structure of the equation

$$\widetilde{Q} = D_C^{-1}(Q - \Delta E) D_C, \tag{35}$$

where $\Delta E = \widetilde{E} - E$ is the difference of "total leaving rates," one can see that the eigenvalues of $\widetilde{Q}$ are identical to the eigenvalues of $(Q - \Delta E)$. Furthermore, the eigenvectors of $\widetilde{Q}$ are the eigenvectors of $(Q - \Delta E)$ except for a componentwise rescaling using the diagonal matrix D_C^{-1}. This rescaling does not have an effect on the sign structure of the eigenvectors.

Lemma 1. The matrix $\widetilde{Q}$ is a similarity transform of $(Q - \nabla E)$; the whole change of the timescales is therefore driven by a perturbation of the diagonal only.

In conclusion, for a PCCA-based analysis of the influence of a coupling energy term on the slowest processes, one has to analyze how changing the diagonal of Q influences the result of the eigenproblem.

III. METHODS

We applied our theoretical framework to two illustrative examples in Sec. IV. Here, we provide the corresponding algorithmic and implementation details. First, we show how we can use the tSqRA to compute the application of $\widetilde{Q}$ and thereby also its spectrum without the large memory overhead of the representations of Q in a dense or sparse format, leading to exponential, respectively, linear, memory savings in the number of dimensions. Then, we introduce the *Projected Rate Estimation* (PRE), an efficient algorithm to estimate the macroscopical transition rates of a coupled system from comparatively few simulations using the tSqRA of the combined system as a prior.

A. Exploiting the tensor formulation to solve the eigenvalue problem of the coupled system

The use of the tensor formulation does not offer any particular advantage in solving, e.g., the eigenvalue problem of the combined matrix Q, where it is more convenient to compose the individual eigenvalues, Eq. (33) and eigenvectors (34). However, in the case of the coupled system $\widetilde{Q}$, it permits us to avoid the explicit construction of a matrix for the application of $\widetilde{Q}$ to a vector, thus enabling the use of matrix-free methods.

Using the results for a global coupling (Sec. II C), we write the rate matrix applied to a column vector v of size $(M \times 1)$ as

$$\widetilde{Q}v = D_C^{-1} Q^{\text{out}} D_C v - \text{diag}\left[D_C^{-1} Q^{\text{out}} D_C\, e\right] v. \tag{36}$$

Making use of the definition $D_C = \text{diag}\left[\sqrt{\pi_C}\right]$ (23) and the fact that multiplying a diagonal matrix from the left is equivalent to the element-wise product $\circ : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$,

$$(x \circ y)_i := x_i y_i = (\text{diag}\,[x] y)_i, \tag{37}$$

the left hand term of the matrix-vector product (36) becomes

$$\left[D_C^{-1} Q^{\text{out}} D_C\right] v = \pi_C^{-1/2} \circ Q^{\text{out}}\left(\pi_C^{1/2} \circ v\right). \tag{38}$$

Likewise, for the second term, we can write

$$\text{diag}\left[D_C^{-1} Q^{\text{out}} D_C\, e\right] v = \left(\pi_C^{-1/2} \circ Q^{\text{out}} \pi_C^{1/2}\right) \circ v. \tag{39}$$

In this way, we have transformed the application of $\widetilde{Q}$ (36) to a series of element-wise operations as well as the application of Q^{out},

$$\widetilde{Q}v = \pi_C^{-1/2} \circ Q^{\text{out}}\left(\pi_C^{1/2} \circ v\right) - \left(\pi_C^{-1/2} \circ Q^{\text{out}} \pi_C^{1/2}\right) \circ v. \tag{40}$$

Note that Q^{out} is the result of the Kronecker sum of the matrices Q_i and as such can be efficiently stored and computed by applying the individual Q_i to the relevant components of the respective input.

In Appendix A 1, we illustrate how this can be achieved by a series of rearrangements of the input followed by a batched matrix-vector product and provide a corresponding python implementation.

Alternatively to this approach of pulling back the computation to matrix algebra, the banded structure of the Kronecker sum can be used to devise a direct cache efficient implementation of its action. We provide a self-contained Julia implementation of this approach, together with an implementation of the tensor formalism from Sec. II D in Appendix B.

In any case, both approaches allow us to calculate the eigenvalues of $\widetilde{Q}$ using matrix-free solvers, such as the ARPACK algorithm[21] without ever creating the whole matrix $\widetilde{Q}$. It follows that we can write the vectors v and e as tensors of order n and shape $(N_1, N_2, \ldots, N_n)$ and the two matrix-vector multiplications as a sum of tensor dot products as in Sec. II D.

This has strong implications in terms of memory consumption. Consider a system with n dimensions, discretized into m bins each: A state vector in that case has a size of $8 \cdot m^n$ bytes, and a dense representation of $\widetilde{Q}$ would need $8 \times m^{2n}$ bytes, but even when using a sparse representation, it would have $\sim 8 \times 2\,\text{nm}^n$ bytes. For a system in $n = 9$ dimensions with $m = 10$ bins each, we would need ~8 GB for the state, 8×10^9 GB for a dense, and 134 GB for a sparse $\widetilde{Q}$ matrix. Using the tensor decomposition of coupled systems, the biggest part of storing $\widetilde{Q}$ is the storage of D_C, which has the same size as a single state vector.

B. The projected rate estimation

When analyzing a coupled system, one is not necessarily interested in the whole matrix $\widetilde{Q}$. In practice, it may be sufficient to study the impact of the coupling on the characteristic time-scales, such as the rates between metastable macrostates.

We therefore propose an algorithm that can be understood as hybrid between the formal SqRA and the data-based Koopman estimation by trajectory simulation. We use the SqRA on the isolated systems to compute their spectrum and obtain the eigenfunctions of the combined systems by means of Eq. (34). Using the PCCA+ algorithm,[22,23] we transform these into membership functions that characterize the slow-scale dynamics and span an invariant subspace of the combined system. These membership functions inform us about the location of the transition regions and provide the basis for a projection to obtain the macroscopic rates of the combined system.[24] By projecting simulated trajectories of the coupled system onto these combined membership functions, we then estimate the slowest rates of the interacting coupled system.

To this end, let us quickly introduce PCCA+, an algorithm to compute the characteristic time-scales, a process also known as "coarse-graining." The fundamental object of PCCA+ are the membership functions $\chi \in \mathbb{R}^{N \times n_C}$ obtained by a linear combination of the n_C dominant eigenfunctions X of Q via the matrix $A \in \mathbb{R}^{n_C \times n_C}$,

$$\chi = XA. \tag{41}$$

PCCA+ finds a matrix A such that χ is non-negative and all its row sums equal to one. The projection of Q onto this subspace leads to the coarse-grained generator $Q_c = A^{-1} \Lambda A \in \mathbb{R}^{n_C \times n_C}$ that provides the rates between the macroscopic states characterized by χ. One defining property of Q_c and χ is that time propagation of the system (via Q or Q_c) and the projection onto the subspace χ commute, i.e., $Q\chi = \chi Q_c$ or using the (Moore–Penrose) pseudoinverse $Q_c = \chi^+ Q \chi$. Analogous results hold for the Koopman operator $K = K^\tau = \exp(\tau Q)$ with lag time $\tau \in \mathbb{R}$ (and the coarse grained K_c), which

has the advantage that it allows for a Monte Carlo approximation via simulations of the system,

$$K^\tau \chi(x) \approx \frac{1}{N_R} \sum_{i=1}^{N_R} \chi(x_i^\tau), \tag{42}$$

where x_i^τ are the end points of N_R sampled trajectories of length τ started in x. The guiding idea of this algorithm is that

$$K\chi = \chi K_c, \tag{43}$$

and similarly, for the coupled system, $\widetilde{K}\widetilde{\chi} = \widetilde{\chi}\widetilde{K}_c$ if $\widetilde{\chi}$ was computed with respect to $\widetilde{K}$, which in practice is not known.

However, for small coupling terms, the dominant eigenspace and therefore the membership functions are perturbed only slightly (see Sec. II E), i.e., $\chi \approx \widetilde{\chi}$. We therefore have

$$\widetilde{K}\chi \approx \chi \widetilde{K}_c, \tag{44}$$

which in turn allows us to approximate the coarse grained coupled dynamics via

$$\hat{K}_c = \chi^+ \widetilde{K}\chi \approx \widetilde{K}_c \tag{45}$$

and the Monte Carlo approximation (42) of $\widetilde{K}\chi$ using the coupled dynamics. Taking the matrix logarithm then allows us to recover the approximated rates of the coupled system $\hat{Q}_c = \operatorname{logm} \hat{K}_c$.

In other words, we estimate the rates of the interacting system using a few simulations projected onto memberships or metastabilities obtained from the tSqRA for the coupled systems.

In this regard, the algorithm to estimate $\widetilde{Q}_c$ is summarized as follows:

1. Solve the eigenproblems for the isolated sub-systems,

$$Q_i X_i = \lambda_i X_i, \tag{46}$$

   where X_i are the matrices containing the eigenvectors of the ith subsystem and λ_i are the vectors containing the corresponding eigenvalues.
2. Identify the number c_i of metastabilities for each ith subsystem from the analysis of the eigenspectrum and compute the membership functions χ_i via PCCA+.
3. Compute the membership functions of the combined system

$$\chi = \bigotimes_{i=1}^{n} \chi_i. \tag{47}$$

   Note that χ_i is a matrix of size $N_i \times c_i$, while χ is a matrix of size $\prod_i N_i \times \prod_i c_i$.
4. Compute the (partial) $\widetilde{K}\chi \in \mathbb{R}^{n \times N}$. This can be done by matrix exponentiation, classical trajectory simulation, or Gillespie's algorithm on the SqRA rates.
5. Compute the pseudoinverse to estimate $\hat{K}_c = \chi^+ \widetilde{K}\chi$.
6. Normalize the rows of $\hat{K}_c$.
7. Estimate the approximated coarse-grained rate matrix as

$$\hat{Q}_c = \frac{1}{\tau} \operatorname{logm} \hat{K}_c. \tag{48}$$

   Let us furthermore add the following remarks to the respective steps:

2. Consider, for example, a receptor–ligand system. Typically, one is interested in the metastabilities of the receptor, which would correspond to choosing the number of metastabilities for the ligand as one, obtaining the constant-one function for the ligand.
3. Similar to Eq. (32), the tensor product of membership functions is the membership function for the product system. This allows us to compute the membership functions on the individual subsystems to obtain the membership function of the combined system.
4. Concerning the computation of $\widetilde{K}\chi$, we propose three distinct approaches. If the system's size permits, we can explicitly compute the matrix exponential $\widetilde{K}$ and apply it to the memberships χ or employ a matrix-free implementation of the matrix-exponential vector product to obtain the precise result of $\widetilde{K}\chi$. Interestingly, it is not required to compute all entries of $\widetilde{K}\chi$ to determine the macroscopic transition rates, as previously observed in Refs. 25 and 26. Given that $n_C < N$, the problem of solving for $\hat{K}_C$ is inherently overdetermined. This situation can be leveraged by estimating only a few rows of $K\chi$ (and multiplying with the corresponding columns of χ^+). To this end, we can employ the Monte Carlo approximation of $\widetilde{K}\chi$ (42). The individual rows of $\widetilde{K}\chi$ correspond to initial conditions for simulating trajectories of the entire system. These trajectories can be computed using classical methods, such as Euler–Maruyama integration or molecular dynamics simulations. Alternatively, we can stay within the discrete tSqRA regime by employing the Gillespie algorithm on $\widetilde{Q}$. Since we only need to compute a subset of rows of $\widetilde{K}\chi$, there is no need to precompute the entire $\widetilde{Q}$; instead, we can compute the necessary rows along the trajectory on demand. The crucial question remains as to which rows of $\widetilde{K}\chi$, i.e., which initial states, are essential for accurately estimating the change in macroscopic rates. We posit that samples should ideally be taken from transition regions, where $0 \ll \chi_i \ll 1$, while also ensuring their physical relevance, characterized by a low potential energy. However, addressing the intricate question of whether and how χ can effectively guide the sampling process—of interest in various approaches, e.g., the stratified-χ sampling in Ref. 27—poses a substantial challenge. Acknowledging the depth of this issue, we have postponed a thorough investigation to the future. For our illustrative experiments, we opted for a pragmatically chosen randomly uniform sampling strategy, recognizing its inherent limitation to lower-dimensional systems.
6. We obtained better results by re-normalizing the rows of $\hat{K}_c$ to have row sum one, corresponding to the interpretation of a probability matrix. We expect that the results could be even further improved by directly restricting the solution of the linear system (45) to the unit simplex using constrained optimization [with optimization objective (58)].
7. The computation of the matrix-logarithms [$\operatorname{logm}(\cdot)$] can be rather involved in general. However, since all eigenvalues of K_c are positive, $0 \ll \lambda_i \leq 1$, we expect this as well for the estimated $\hat{K}_c$, resulting in a unique solution. Moreover, the coarse-grained matrix is of small size, alleviating concerns about stability.

IV. RESULTS

A. Two-dimensional example

This two-dimensional example illustrates our theoretical elaborations. No special increases in efficiency are required to recalculate these examples in a comprehensible manner.

1. Numerical experiment parameters

As an illustrative example, we considered the overdamped Langevin dynamics of a particle of mass $m = 1$ amu and friction $\xi = 1$ ps^{-1}, which moves in a two-dimensional space (D = 2) under the action of the potential energy function,

$$\begin{aligned}\widetilde{V}(x_1, x_2) &= V_1(x_1) + V_2(x_2) + cV_{12}(x_1, x_2)\\ &= (x_1^2 - 1)^2 + x_1 + 2x_2^2 + c\,x_1x_2 \quad (\text{kJ mol}^{-1}).\end{aligned} \tag{49}$$

The function is made of two potentials V_1 and V_2 describing the dynamics of two non-interacting subsystems along the coordinates x_1 and x_2, respectively. Additionally, a coupling term V_{12} can be activated ($c \neq 0$) or deactivated (c = 0) by the parameter c. The two-dimensional function, illustrated in Fig. 1(a) for $c = 0$ and $c = 1$, describes a surface with two wells of different heights separated by a barrier.

For our numerical experiments, we assumed standard thermodynamic parameters: the temperature of the system was $T = 300$ K; the molar Boltzmann constant was $k_B = 8.31 \times 10^{-3}$ kJ mol^{-1} K^{-1} such that the diffusion constant is 2.49 nm^2 ps^{-1} in each direction. Note that the use of units here is optional, but we include them for reference when computing a physical system. When performing these computations without units, one obtains the same results up to a scaling of the eigenvalues by the diffusion constant (see also Ref. 4).

2. The rate matrix

In order to build the rate matrix by SqRA, we discretized the x_1-range $[-3.4, 3.4]$ nm and x_2-range $[-3.4, 3.4]$ nm, respectively, in $N_1 = N_2 = 50$ subsets of the same length $\Delta x_1 = \Delta x_2 = 0.13$ nm for a total of $M = N_1 \cdot N_2 = 2500$ square subsets of the two-dimensional space.

For the combined system ($c = 0$), we built, respectively, the rate matrices Q_1 and Q_2 (size 50×50); then, we estimated the first four eigenvalues and right eigenvectors solving the eigenvalue problems

$$Q_1X_1 = \lambda_1X_1 \tag{50}$$

and

$$Q_2X_2 = \lambda_2X_2, \tag{51}$$

where X_1 and X_2 are matrices of size 50×4 containing the first four eigenvectors and λ_1 and λ_2 are four-dimensional vectors containing the corresponding eigenvalues. For the combined system ($c = 0$), the rate matrix Q of the entire system can be estimated from the Kronecker sum of the rate matrices Q_1 and Q_2. Exploiting this property (see Ref. 6 for more details), we estimated sixteen eigenvalues λ of Q as the sum of the eigenvalues of Q_1 and Q_2 [Eq. (33)],

$$\lambda_k = \lambda_{1,i} + \lambda_{2,j} \quad \forall i,\ j = 0, 1, 2, 3, \tag{52}$$

and the corresponding eigenvectors X as the Kronecker product [Eq. (34)],

$$X_k = X_{1,i} \otimes X_{2,j} \quad \forall i,\ j = 0, 1, 2, 3. \tag{53}$$

In Eqs. (52) and (53), we introduce the index $k = i + 4j$. Note that the eigenvectors X_1 and X_2 have size $N_1 = N_2 = 50$, while X has size $M = 2500$.

The eigenvalues and the corresponding eigenvectors are reported, respectively, in Figs. 1(b) and 1(c). The first eigenvector is constant and is associated with the eigenvalue $\lambda = 0$. This eigenvector represents the stationary state of the system when the other eigenmodes have decayed. The other eigenvectors X_k, with $k > 0$, represent the dominant kinetic processes between regions of the space with negative (blue color) and positive values (red color). The associated eigenvalues λ_k are negative and represent the timescales $-1/\lambda_k$ at which the dominant processes decay. We conclude that the slowest process is a transition between the regions of the space with $x_1 < 0$ and $x_1 > 0$, the second slowest process is a transition between the regions of the space with $x_2 < 0$ and $x_2 > 0$, and the third slowest process is a mix transition between the quadrants of the plane.

For the coupled system ($c \neq 1$), the relations in Eqs. (52) and (53) do not hold, i.e., the eigenvectors and eigenvalues of the full system are not directly derived from those of the individual subsystems. In this example, we built the full matrix $\widetilde{Q}$ applying Eq. (20), which requires to provide the diagonal matrix D_{12} with $50 \times 50 = 2500$ entries. The main effect of the coupling term is to distort the eigenvectors and to break the symmetries of the combined system. With regard to the eigenvalues, we observe a rise in the second eigenvalue λ_1 and a decrease in the third eigenvalue λ_2. From a physical perspective, this corresponds to an acceleration of the transition along the x_2 coordinate and a slowdown along the x_1 coordinate. In Fig. 1(b), we also plot the eigenvalues of the off-diagonal matrices Q^{out} and $D_{12}^{-1}Q^{\text{out}}D_{12}$. The eigenvalues of the two matrices perfectly match due to a similarity transform. Thus, it is the matrix $\widetilde{E}$ of the total leaving rates defined in Eq. (23) that affects the timescales of the system.

3. Coarse-grained rate matrix

The analysis of the system's eigenvectors and eigenvalues indicates that the system oscillates between two metastable states, i.e., two regions of the state space where the system resides for most of the time, with the transition from one state to the other occurring rarely. It is therefore convenient to reduce the matrices Q and $\widetilde{Q}$ to matrices Q_c and $\widetilde{Q}_c$ of size 2×2, containing the rates k_{12} and k_{21} between the two metastable regions. For this purpose, we used PCCA+, which provides the membership functions χ containing the probabilities that a state belongs to the identified metastable states. The membership functions can then be used to build the coarse-grained matrix Q_c by projecting the fine-grained matrix Q,

$$Q_c = (\chi^\top \operatorname{diag}(\pi)\chi)^{-1}\chi^\top \operatorname{diag}(\pi)Q\chi. \tag{54}$$

With regard to the combined system ($c = 0$), in practice, it is more convenient to exploit the algebraic structure of the rate matrix Q and to apply PCCA+ to Q_1 and Q_2 to obtain the membership functions χ_1 (size 50×2) and χ_2 (size 50×1) and the rate matrices $Q_{c,1}$ (size

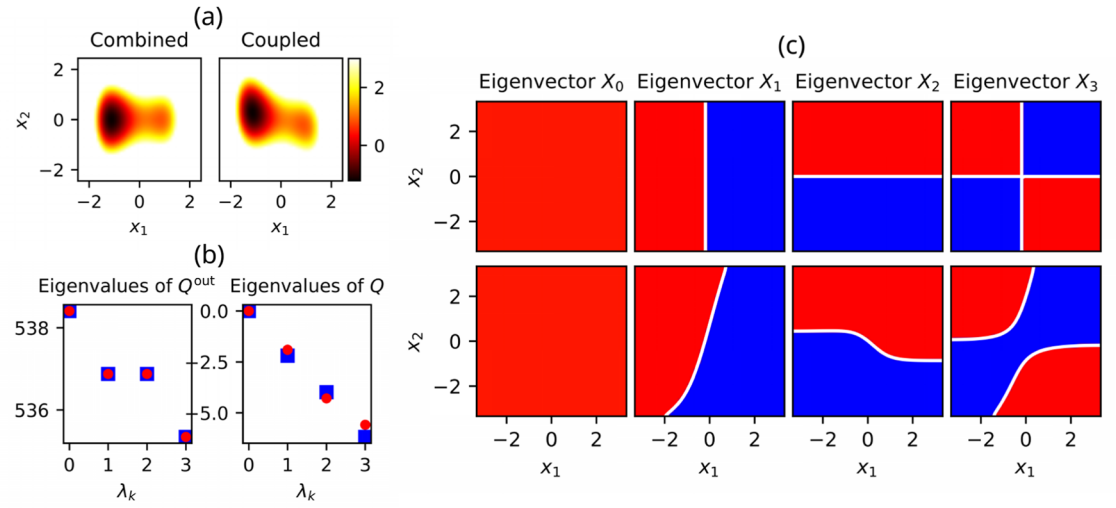

FIG. 1. 2D model. (a) Potential energy function for the combined (left) and coupled system with $c = 1$ (right). (b) Eigenvalues of the matrix Q^{out} (left) and Q (right) for the combined (blue) and coupled system (red). (c) First four eigenvectors for combined (top) and coupled system (bottom); the red and blue colors denote positive and negative values, respectively, and the white lines represent the zero-values.

2×2) and $Q_{c,2}$ (size 1). Note that $\chi_2 = \mathbf{1}$ (size 50) because the system has one metastability along the x_2 direction and $Q_{c,2} = 0$. Then, the membership functions of the coupled system are obtained by the Kronecker product,

$$\chi = \chi_1 \otimes \chi_2, \tag{55}$$

while the rate matrix is given by the Kronecker sum,

$$Q_c = Q_{c,1} \oplus Q_{c,2} = Q_{c,1} = \begin{pmatrix} -1.45 & 1.45 \\ 0.74 & -0.74 \end{pmatrix}. \tag{56}$$

Because the system has two metastabilities, the Kronecker product simply results in an increase in the size of χ_1 from 50 entries to $50 \times 50 = 2500$ entries since χ_2 is a constant function, and the rate matrix Q_c is equal to $Q_{c,1}$. However, if the second subsystem has two metastabilities as well, then the coupled system would have four metastabilities and the corresponding Q_c matrix would have the size 4×4.

4. Coupled system

For the coupled system ($c = 1$), we applied the PRE algorithm derived in Sec. III B. The algorithm is based on the assumption that the Koopman operator $\widetilde{K}^\tau$ of the coupled system applied to membership functions χ of the combined system well approximates the resultant of the coarse-grained Koopman operator $\widetilde{K}_c^\tau$ applied to χ [see Eq. (44)]. First of all, we verified the validity of this assumption by constructing the discretized coupled Koopman operator as

$$\widetilde{K}^\tau = \mathrm{expm}(\tau \widetilde{Q}), \tag{57}$$

where $\widetilde{Q}$ is the rate matrix of the coupled system built by SqRA and $\mathrm{expm}(\cdot)$ denotes a matrix exponential. The matrix $\widetilde{K}^\tau$ has been multiplied to the membership functions χ of the combined system, and we solved the linear regression problem

$$\min_{\hat{K}_c} \|\widetilde{K}^\tau \chi - \chi \hat{K}_c^\tau\|, \tag{58}$$

i.e., we looked for the matrix $\hat{K}_c^\tau$ (size 2×2) that minimizes the two-norm $\|\widetilde{K}^\tau \chi - \chi \hat{K}_c^\tau\|$. The matrix $\hat{K}_c^\tau$ approximates the exact coarse-grained Koopman operator $\widetilde{K}_c^\tau$ for the coupled system and yields the approximated coarse-grained rate matrix as

$$\hat{Q}_c = \frac{1}{\tau} \mathrm{logm}\, \hat{K}_c^\tau. \tag{59}$$

In Fig. 2(b), we plotted the off-diagonal entries of $\hat{Q}_c$ (dashed lines) for several τ values in the range [0,1.5] ps and compared with the exact values of $\widetilde{Q}_c$ (solid lines) obtained by applying PCCA+ to $\widetilde{Q}$. We observe that while the exact results do not depend on the choice of lag time τ, the approximate results tend to converge only for large values of τ. However, the approximate rates do not reach the exact values.

If building the matrix $\widetilde{K}^\tau$ is not feasible due to the high dimensionality, the action of the Koopman operator on the membership functions χ can only be approximated for some points x in the state space. Provided N_R trajectories of length τ, starting in x and reaching the points x_i^τ, with $i = 1, 2, \ldots, N_R$, yield

$$\widetilde{K}^\tau \chi(x) \approx \frac{1}{N_R} \sum_i^{N_R} \chi(x_i^\tau). \tag{60}$$

The trajectories could be generated by solving the underlying equations of motion of the coupled system, for example, by using the Euler–Maruyama integration scheme. Instead, we exploited

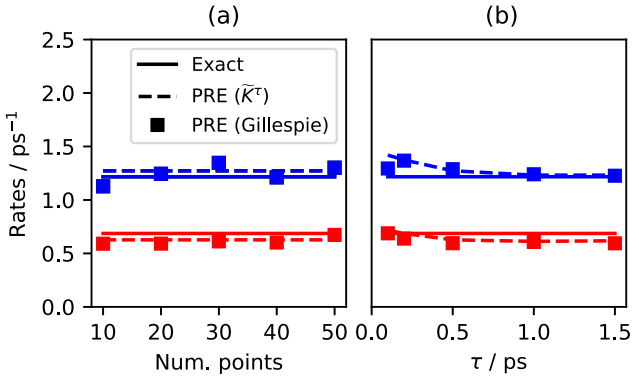

FIG. 2. 2D model. Transition rates k_{12} (blue) and k_{21} (red) of the coarse-grained model as functions of the number of starting points (a) and the lag time τ (b). The solid lines represent the exact results using $\widetilde{Q}$, the dashed lines represent the projected rate estimation (PRE) using the exact $\widetilde{K}^\tau = \exp(Q\tau)$, and the squares represent the PRE using Gillespie's algorithm to estimate $\widetilde{K}^\tau$.

the knowledge of the transition rates between adjacent subsets of the state space, approximated by SqRA, and applied the Gillespie algorithm.

In order to explain the procedure, consider the following example. Let us assume that we know the matrices Q_1 and Q_2 of the one-dimensional individual subsystems and that the system is in a state $x = (x_1, x_2)$ belonging to a subset of the state space with indices (i_1, i_2). The system, in an infinitesimal interval of time, can evolve its state in five ways: two transitions along the x_1 direction, two along the x_2 direction, or no transition. The four transition rates are contained in the Q_1 and Q_2 matrices, but these must be reweighted according to the coupling term V_{12} in Eq. (49). For example, the forward transition rate along the x_1 direction becomes

$$\widetilde{Q}_1^{i_1,i_1+1} = Q_1^{i_1,i_1+1} \cdot \frac{\sqrt{\pi_{12}^{i_1+1,i_2}}}{\sqrt{\pi_{12}^{i_1,i_2}}}. \tag{61}$$

Likewise, we obtain the rates $\widetilde{Q}_1^{i_1,i_1-1}$, $\widetilde{Q}_2^{i_2,i_2+1}$, and $\widetilde{Q}_2^{i_2,i_2-1}$ and the total leaving rate given by minus the sum of the former. Thus, we have estimated only a row of the matrix $\widetilde{Q}$, which can be used within a standard Gillespie algorithm to estimate (i) the infinitesimal time interval at which the next transition occurs and (ii) which of the five possible events (four transitions and no transition) takes place. The reweighting of rates for the coupled system occurs at each simulation step; however, for highly dimensional systems, this is the only solution since it is not possible to estimate the entire matrix $\widetilde{Q}$.

Since the Gillespie algorithm provides the solution in terms of jumps between the subsets of the state space, it provides the indices (j_1, j_2) of the arrival subset of x^τ and the calculation of the membership functions can be approximated by calculating the value that χ assumes in the target subsets,

$$\chi(x^\tau) \approx \chi^{j_1,j_2}. \tag{62}$$

Repeating this calculation for N_R trajectories, one can approximate the Koopman operator according to Eq. (60) and solve the linear regression problem in Eq. (58). However, unlike the situation where we have the entire matrix $\widetilde{K}^\tau$, after solving the linear regression problem, it is necessary to normalize the rows of the matrix $\widetilde{K}_c^\tau$.

We studied the sensitivity of the method to the initial points and the lag time τ of the Koopman operator and reported the results in Fig. 2. First of all, we studied the dependence of the method on the number of initial points x, randomly drawn from a uniform distribution that covers the state space, setting a lag time equal to $\tau = 0.5$ ps and a number of replicas per point $N_R = 100$. We do not observe any particular improvement in the results by increasing the number of initial points, so for the next analysis, we considered ten initial points and increased the number of replicas to 500 to reduce the stochastic error. Excluding the first point, which corresponds to a lag time of 0.1 ps, the agreement with the expected results (dashed line) is excellent.

B. *n*-dimensional example

Without special mathematical tricks, one could not calculate the SqRA for high-dimensional examples. In this example, we heavily use the algebraic structure of tSqRA to rigorously calculate the eigenvalues of a large rate "matrix." That this becomes possible is shown in Fig. 3.

1. Numerical experiment parameters

As a second example, we studied a "metastable chain" made of n bimetastable systems interacting via harmonic oscillators. The potential energy function describing the dynamics of this system is written as

$$\begin{aligned}\widetilde{V}(x_1, x_2, \ldots, x_n) &= V_1(x_1) + V_2(x_2) + \cdots + V_n(x_n) \\ &\quad + c \cdot [V_{12}(x_1, x_2) + V_{23}(x_2, x_3) + \cdots \\ &\quad + V_{n-1,n}(x_{n-1}, x_n)] \\ &= \sum_{i=1}^{n} V_i(x_i) + c \cdot \sum_{i=1}^{n-1} V_{i,i+1}(x_i, x_{i+1}),\end{aligned} \tag{63}$$

where

$$V_i(x_i) = (x_i^2 - 1)^2 \tag{64}$$

and

$$V_{i,i+1}(x_i, x_{i+1}) = \frac{1}{2}|x_i - x_{i+1}|^2. \tag{65}$$

The combined system $(c = 0)$ is characterized by n^2 metastable states, while the coupled system $(c = 1)$ has only two metastable states, as illustrated in Fig. 3(a) for $n = 2$ and $n = 3$, respectively. For the numerical experiments, we used the same thermodynamic parameters as in the previous examples. To build the rate matrix Q_i of each bimetastable system by SqRA, we discretized the range $[-2.5, 2.5]$ nm of each coordinate x_i into $N_i = 5$ subsets of length $\Delta x_i = 1.0$ nm; then, the total number of subsets of the system is 5^n.

Differently from the previous example, constructing the rate matrices Q and $\widetilde{Q}$ of the coupled system (combined and coupled)

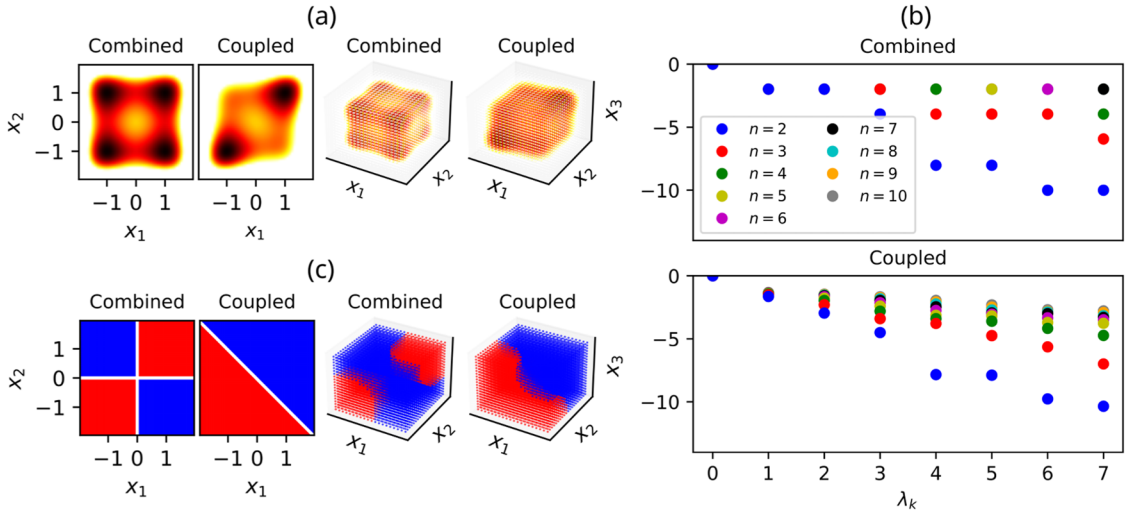

FIG. 3. Metastable chain. (a) Combined and coupled potential energy function for $n = 2$ and $n = 3$; dark and bright colors represent, respectively, low and high energy values. (b) First eight eigenvalues for $n = 2, 3, \ldots, 10$ of the combined and coupled system. (c) Second eigenvectors for $n = 2$ and $n = 3$ of the combined and coupled system.

is not feasible for high values of n. Thus, we used the approach described in Sec. III A to calculate the eigenvalues and eigenvectors without building the full rate matrices but providing to the eigensolver a function that represents their action on a vector of size $(N \times 1)$. Figure 3(b) shows the first eight eigenvalues of the rate matrix Q and $\widetilde{Q}$. For the combined case (a), we observe that the number of eigenvalues associated with the slowest processes is equal to n; then, the slowest transitions occur along the coordinates x_i. On the contrary, the effect of the coupling term is to change the symmetry of the system and to create a hierarchy of sub-processes arising at different time scales. Thus, we observe a decreasing scale of eigenvalues for each n-dimensional system. The first non-zero eigenvalue corresponds to the slowest process occurring along the diagonal of the n-dimensional circle, as shown in the Fig. 3(c) by the corresponding eigenvectors for the case with $n = 2$ and $n = 3$.

V. CONCLUSION AND OUTLOOK

This article shows how the algebraic structure of SqRA can be exploited in order to systematically analyze the macroscopic timescales of slow processes in coupled molecular systems from first principles. Our article is a step toward understanding the origins of these timescales and the influence of interaction energies between subsystems.

Using the tensor approach (tSqRA), we can describe the direct relation between energies and rates of coupled systems, which allows for its mathematical analysis based on first-principles models of molecular processes. We demonstrate how this allows us to treat an high-dimensional system as a set of interacting subsystems of lower dimensions. Potentially, this offers a method to calculate the transition rates of complex systems without the need for simulations. We have shown this in the second example where we studied a ten-dimensional system.

Going beyond the pure formal tensor-computations by exploiting the interpretation of a coupled system as a perturbed combined system, we bring together tensor methods with classical simulation. In combining the tSqRA with PCCA+, we suggest a new hybrid method, the projected rate estimation (PRE), which uses the combined non-interacting subsystems (model-based) as prior for an effective estimation of the macroscopical rates of a coupled system by a comparatively small number of simulations (data-based). So far, its efficacy is limited to low-dimensions due to the yet open question of the choice of initial states for the simulations. We, nevertheless, believe that the algorithm shows in how far the developed tensor view enables new approaches to classical problems.

On the computational side, the tSqRA improves the applicability and efficacy of the SqRA for coupled systems even without the use of low-rank approximations. However, we also explain how tSqRA can be extended to incorporate low-rank approximations. Currently, the limiting factor for an efficient tensor-based calculation is the dense representation of the system's state, which is succumbed to the curse of dimensionality. However, a low rank representation, e.g., in the form of tensor networks, could alleviate that problem.[28] The lacking piece so far is the support for truncated Hadamard-products.

Note in this regard that the adjacency matrix A, being a Kronecker sum, is inherently low rank. Similarly, the matrix D is composed of Hadamard products, such as in Eq. (28). Assuming that the system under consideration admits a low-rank representation, as to be expected for two weakly coupled systems, we expect the exploitation of these low-rank structures to dramatically improve both memory and compute costs.

From our perspective, it is worthwhile to better comprehend the computational advantage of tensor-based complexity reduction methods for the future analysis of molecular systems without the need for molecular simulation.

ACKNOWLEDGMENTS

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through the Cluster of Excellence MATH+, Project No. AA1-15 "Math-powered drug-design," DFG Project No. 390685689, as well as by Project Nos. B03 "Multilevel coarse graining of multiscale problems" and B05 "Origin of scaling cascades in protein dynamics" of the Collaborative Research Center CRC 1114 "Scaling Cascades in Complex Systems," DFG Project No. 235221301.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Alexander Sikorski and Luca Donati contributed equally to this paper.

Alexander Sikorski: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Amir Niknejad**: Conceptualization (supporting); Project administration (supporting); Resources (supporting); Supervision (supporting); Writing – original draft (supporting). **Marcus Weber**: Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal). **Luca Donati**: Conceptualization (supporting); Data curation (supporting); Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (lead); Visualization (lead); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The scripts for recreating the experiments as well as from the appendices can be found at https://github.com/zib-cmd/article-tsqra.

APPENDIX A: PYTHON IMPLEMENTATION OF A GLOBAL COUPLING TERM

1. From matrix-vector multiplication to sum of tensor dot products

When computing with tensors, one difficulty is how to actually represent them for computations, either as n-dimensional arrays or in their "flattened" representation, where states are vectors and operator matrices. In this section, we describe how we can compute the action of a direct sum of matrices, interpreted as a flattened matrix, as the sum of matrix products along the individual tensor modes. We will then use this approach in the following Python implementation.

Consider n matrices Q_i each of same size $(N_i \times N_i)$ and the matrix Q of size $(N_i^n \times N_i^n)$ obtained as the Kronecker sum of Q_i as in Eq. (13). Given an arbitrary column vector v of size $(N_i^n \times 1)$, the result of the matrix-vector multiplication is the column vector,

$$u = Qv = \left(\bigoplus_{i=1}^{n} Q_i\right) v, \tag{A1}$$

of size $(N_i^n \times 1)$. This operation requires the explicit construction of the matrix Q. Instead, we now reformulate the same operation as a sum of tensor dot products applied to the smaller matrices Q_i.

First, we reshape the vector v into a tensor of order n with shape $(N_1, N_2, \ldots, N_n)$ and entries $v^{\gamma_1,\gamma_2,\ldots,\gamma_n}$, where each index γ_i specifies the ith dimension of the tensor. Next, we perform the following two operations on v to build n tensors u_i of order n:

1. Tensor contraction between the matrix Q_i and the tensor v,

$$u_i^{\alpha,\gamma_1,\gamma_2,\ldots,\gamma_n} = \sum_{\kappa_i=1}^{N_i} Q_i^{\alpha\kappa_i}\, v^{\gamma_1,\gamma_2,\ldots,\kappa_i,\ldots,\gamma_n}, \tag{A2}$$

   where κ_i is the contraction index, that is, the index on which to sum the products between the entries of the matrix Q_i and the entries of the tensor v. In other words, the entries of the resulting tensor u_i are the dot products between the k_ith column of Q_i and the k_ith dimension of v. Here, we added the subscript i to κ_i to remark that it replaces the ith index of v in Eq. (A2). It follows that u_i is a tensor of order n, which contains all the indexes of Q_i and v but the second index of Q_i and the ith index of v. The operation in Eq. (A2) is repeated $\forall i = 1, 2, \ldots, n$.
2. Index swapping,

$$u_i^{\alpha,\gamma_1,\gamma_2,\ldots,\gamma_n} \to u_i^{\gamma_1,\gamma_2,\ldots,\alpha_{(i)},\ldots\gamma_n}, \tag{A3}$$

   where the notation $\alpha_{(i)}$ remarks that the tensor must be reordered such that α occupies the ith position.

Once we computed the n tensors u_i, one for each matrix Q_i, the tensor u representing the matrix multiplication between Q and v is the sum of tensors u_i with entries

$$u^{\gamma_1,\gamma_2,\ldots,\gamma_n} = \sum_{i=1}^{n} u_i^{\gamma_1,\gamma_2,\ldots,\alpha_{(i)},\ldots\gamma_n}. \tag{A4}$$

The final result is a tensor of order n, which can be flattened into a one-dimensional vector equal to the result of the matrix-vector multiplication defined in Eq. (A1). We implement this approach in the following Python code.

2. Python codes for computations of Qv and eigenvalues of $\widetilde{Q}$ based on matrices

Listing 1. The code defines the function *sum_tensor_dots*() that calculates the matrix-vector multiplication Qv as a sum of tensor dot products; then, it solves the eigenvalue problem. In this example, the matrix Q_i and the vector v are randomly generated, and it is assumed that each dimension is discretized into the same number of subsets. The function *np.tensordot*() applies the tensor contraction to axis 1 of the matrix and the index i of the vector in its tensor formulation. Similar solutions can also be found with

the function *np.einsum*(). Note that in Python the indexing starts from 0.

```
import numpy as np
import scipy.sparse.linalg as spl
n      = 5
Ni     = 2
N      = Ni ** n
Qi     = np.random.normal( 0, 1, ( Ni, Ni ) )
v      = np.random.normal( 0, 1, ( N,  1  ) )

tensor_shape = [Ni for i in range(n)]

def Q_matmul(v):
    v            = v.reshape(tensor_shape)
    ui           = np.empty(n, dtype=object)
    for i in range(n):
        td   = np.tensordot( Qi, v, axes = [[1], [i]] )
        for j in range(i):
            td = td.swapaxes(j,j+1)
        ui[i] = td
    return np.sum(ui).flatten('C')

u = Q_matmul(v)
lin_op_Q     = spl.LinearOperator((N, N), matvec=Q_matmul)
evals, evecs = spl.eigs(lin_op_Q, k=4, which='LR')
```

Listing 2. The code calculates the eigenvalues of the coupled matrix $\widetilde{Q}$ exploiting the tensor formulation of its action on a vector v. The array p, of shape $(N_1, N_2, \ldots, N_n)$, contains the entries of $\pi_{12\ldots n}^{1/2}$.

```
def tildeQ_matmul(v):
    e     = np.ones(tensor_shape)
    term1 = p ** -1 * Q_matmul( p * v )
    term2 = Q_matmul( p * e ) * v
    out   = term1 - term2
    return out.flatten('C')

lin_op       = spl.LinearOperator((N, N), matvec = tildeQ_matmul)
evals, evecs = spl.eigs(lin_op, k=11, which='LR')
```

APPENDIX B: JULIA IMPLEMENTATION FOR MULTIPLE LOW-ORDER INTERACTIONS

The following code illustrates how to compute the application of Q using the tensor approach developed in Sec. II D. apply_A exploits the fact that A is banded (with entries 1) to copy and add the specific bands of the input x to the output y (see Fig. 4). compute_D then loops over a given list of potential functions, evaluates them at the grids of the specified dimensions, and multiplies them according to Eq. (28) along the corresponding modes using Julia's broadcasting capabilities. We finally compute D and E for the example system with potential $V(x,y) = x^2 + (xy)^2$ on a square grid of size 10×10. The last two functions implement the application of $A - E$ and Q, respectively.

Using this, we were able to compute a simplified nine-dimensional pentane molecule with ten cells in each dimension, resulting in a memory demand of $10^9 \times 64$ bit = 8 Gb per state. While the computation of D is a matter of minutes, the computation of the spectrum using an Arnoldi method requires computation time on the order of 1 day due to the dense state representation.

Listing 3. Julia implementation of the tSqRA for multiple low-order interactions as in Sec. II D.

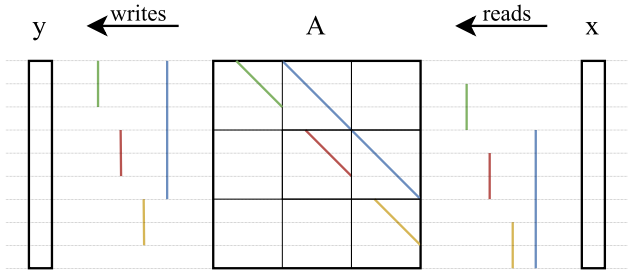

FIG. 4. Schematic illustration of the cache efficient application of $y = Ax$ exploiting the banded structure for linear memory accesses without storage of A. For simplicity, we only show the computations for the upper diagonal; the lower follows analogously.

```
Vec = Vector
Tensor = Array
Grid = Union{Vector,AbstractRange}

function apply_A(x::Tensor, dims::Tuple)
    y = zeros(size(x))
    len = length(x)
    off = 1 # offset
    for cd in dims # current dimension length
        bs = off * cd # blocksize
        for i in 1:bs:len-off # blockstart
            to = i:i+bs-off-1
            y[to] .+= x[to.+off]
            y[to.+off] .+= x[to]
        end
        off = bs
    end
    return y::Tensor
end

# take a list of potentials, a list of modes to apply to
# and a list of grids to evaluate
function compute_D(
        potentials::Vec{Function}, # list of potentials
        indices::Vec{<:Vec}, # list of dimensions each potential acts on
        grids::Vec{<:Grid}, # list of grids for each dimension
        beta=1,
    )

    y = ones(length.(grids)...)
    for (v, inds) in zip(potentials, indices)
        p = map(Iterators.product(grids[inds]...)) do x
            exp(-1 / 2 * beta * v(x))
        end
        dims = ones(Int, length(grids))
        dims[inds] .= size(p) # specify broadcasting dimensions
        y .*= reshape(p, dims...) # elementwise mult. of potential
    end
    return y::Tensor
end

apply_Q(x, D, E) = apply_A(x .* D, size(E)) ./D .- E .* x

# Example with V(x,y) = x^2 + x^2*y^2 on the domain [-1,1]^2
# For illustrative purposes we represent the potential
# as sum of a 1 and 2 dimensional potential
potentials = [x -> x[1] .^ 2, x -> x[1] .^ 2 .* x[2] .^ 2]
        indices = [[1], [1, 2]]
        grids = [range(-1, 1, 10), range(-1, 1, 10)]

        D = compute_D(potentials, indices, grids)
        E = apply_A(D, size(D)) ./ D
        x = rand(10, 10)
        Qx = apply_Q(x, D, E)
```

APPENDIX C: TABLE OF SYMBOLS

Table I shows the key variables and their symbols used in this article.

TABLE I. Table of symbols.

	Symbol	Size
Number of Cartesian coordinates	D	Scalar constant
Number of grouped coordinates/subsystems	n	Scalar constant
Number of coordinates of the ith subsystem	D_i	Scalar constant
Cartesian coordinates of the full system	x	N-dimensional vector
Indices of the subsystems	$i, j, k, \ldots$	Scalar index
Grouped coordinates of ith subsystem	x_i	Scalar variable
Observable function of the combined system	$f(x)$	N-dimensional function
Observable function of the coupled system	$\widetilde{f}(x)$	N-dimensional function
Observable function of the ith subsystem	$f_i(x_i)$	L_i-dimensional function
Observable function of the $i \sim j$ coupled system	$f_{ij}(x_i, x_j)$	$(\mathrm{D}_i + \mathrm{D}_j)$-dimensional function
Number of subsets in which x_i is discretized	N_i	Scalar constant
Total number of subsets in which x is discretized	N	Scalar constant
Indices of the subsets of a discretized coordinate	$\alpha, \beta, \gamma, \ldots$	Scalar index
Center of the αth subset of the discretized coordinate x_i	x_i^α	Scalar variable

REFERENCES

[1] C. Schütte and M. Sarich, *Metastability and Markov State Models in Molecular Dynamics*, *Courant Lecture Notes* (Providence, 2014), Vol. 24.

[2] H. C. Lie, K. Fackeldey, and M. Weber, "A square root approximation of transition rates for a Markov state model," SIAM. J. Matrix Anal. Appl. **34**, 738–756 (2013).

[3] L. Donati, M. Heida, B. G. Keller, and M. Weber, "Estimation of the infinitesimal generator by square-root approximation," J. Phys.: Condens. Matter **30**, 425201 (2018).

[4] L. Donati, M. Weber, and B. G. Keller, "Markov models from the square root approximation of the Fokker–Planck equation: Calculating the grid-dependent flux," J. Phys.: Condens. Matter **33**, 115902 (2021).

[5] L. Donati, M. Weber, and B. G. Keller, "A review of Girsanov reweighting and of square root approximation for building molecular Markov state models," J. Math. Phys. **63**, 123306 (2022).

[6] C. Loan, "The ubiquitous Kronecker product," J. Comput. Appl. Math. **123**, 85–100 (2000).

[7] T. Dayar, *Analyzing Markov Chains Using Kronecker Products: Theory and Applications* (Springer Publishing Company, Incorporated, 2012).

[8] G. Ludyk, *Quantum Mechanics in Matrix Form* (Springer Cham, 2018).

[9] D. Pollock, "On Kronecker products, tensor products and matrix differential calculus," Int. J. Comput. Math. **90**, 2462–2476 (2013).

[10] S. Jokar and V. Mehrmann, "Sparse solutions to underdetermined Kronecker product systems," Linear Algebra Appl. **431**, 2437–2447 (2009).

[11] F. Fernández, "The Kronecker product and some of its physical applications," Eur. J. Phys. **37**, 065403 (2016).

[12] W. Ching, M. K. Ng, and E. S. Fung, "Higher-order multivariate Markov chains and their applications," Linear Algebra Appl. **428**, 492–507 (2008).

[13] A. Ramanathan, P. K. Agarwal, and C. J. Langmead, "Using tensor analysis to characterize contact-map dynamics of proteins," Ph.D. thesis, Carnegie Mellon University Pittsburgh, PA, 2008.

[14] V. Khoromskaia and B. Khoromskij, "Tensor numerical methods in quantum chemistry: From Hartree–Fock to excitation energies," Phys. Chem. Chem. Phys. **17**, 31491 (2015).

[15] M. Lücke and F. Nüske, "tgEDMD: Approximation of the Kolmogorov operator in tensor train format," J. Nonlinear Sci. **32**, 44 (2022).

[16] P. Gelß, "The tensor-train format and its applications: Modeling and analysis of chemical reaction networks, catalytic processes, fluid flows, and Brownian dynamics," Doctoral thesis, FU Berlin, 2017.

[17] S. Kube and M. Weber, "A coarse graining method for the identification of transition rates between molecular conformations," J. Chem. Phys. **126**, 024103 (2007).

[18] M. Heida, "Convergences of the squareroot approximation scheme to the Fokker–Planck operator," Math. Models Methods Appl. Sci. **28**, 2599–2635 (2018).

[19] P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte, "Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains," Linear Algebra Appl. **315**, 39–59 (2000).

[20] P. Deuflhard and M. Weber, "Robust Perron cluster analysis in conformation dynamics," Linear Algebra Appl. **398**, 161–184 (2005).

[21] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK: Solution of large scale eigenvalue problems by implicitly restarted Arnoldi methods*, netlib@ornl.gov, 1997.

[22] M. Weber, "Meshless methods in conformation dynamics," Doctoral thesis, Freie Universität Berlin, 2006.

[23] S. Röblitz and M. Weber, "Fuzzy spectral clustering by PCCA+: Application to Markov state models and data classification," Adv. Data Anal. Classif. **7**, 147–179 (2013).

[24] M. Weber, "Implications of PCCA+ in molecular simulation," Computation **6**, 20 (2018).

[25] K. Fackeldey, A. Niknejad, and M. Weber, "Finding metastabilities in reversible Markov chains based on incomplete sampling," Spec. Matrices **5**, 73–81 (2017).

[26] A. Bittracher and C. Schütte, "A probabilistic algorithm for aggregating vastly undersampled large Markov chains," Physica D **416**, 132799 (2021).

[27] A. Sikorski, E. Ribera Borrell, and M. Weber, "Learning Koopman eigenfunctions of stochastic diffusions with optimal importance sampling and ISOKANN," J. Math. Phys. **65**, 013502 (2024).

[28] A. Cichocki, N. Lee, I. Oseledets, A. Phan, Q. Zhao, and D. P. Mandic, "Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions," Found. Trends Mach. Learn. **9**, 249–429 (2016).