



Committor functions via tensor networks

Yian Chen^{a,*}, Jeremy Hoskins^a, Yuehaw Khoo^{a,*}, Michael Lindsey^b

^a Department of Statistics, University of Chicago, Illinois, IL 60637, USA

^b Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720, USA

ARTICLE INFO

Article history:

Received 30 June 2021

Received in revised form 13 September 2022

Accepted 19 September 2022

Available online 4 October 2022

Keywords:

Committor function

Variational formulation

Tensor network

Tensor train

Matrix product state

Alternating least squares

ABSTRACT

We propose a novel approach for computing committor functions, which describe transitions of a stochastic process between metastable states. The committor function satisfies a backward Kolmogorov equation, and in typical high-dimensional settings of interest, it is intractable to compute and store the solution with traditional numerical methods. By parametrizing the committor function in a matrix product state/tensor train format and using a similar representation for the equilibrium probability density, we solve the variational formulation of the backward Kolmogorov equation with linear time and memory complexity in the number of dimensions. This approach bypasses the need for sampling the equilibrium distribution, which can be difficult when the distribution has multiple modes. Numerical results demonstrate the effectiveness of the proposed method for high-dimensional problems.

© 2022 Elsevier Inc. All rights reserved.

1. Introduction

Understanding rare transitions between metastable states of a high-dimensional stochastic processes is a problem of great importance in the applied sciences. Examples of interesting transition events include chemical reactions, nucleation events during phase transitions, and conformational changes of molecules [25,33,57,58,4,11]. In such complex systems, the dynamics linger near metastable states for long waiting periods, punctuated by sudden jumps from one metastable state to another. One important tool for describing transition events is transition path theory [54,30,48,52], where the committor function plays a central role. The committor function measures the probability that the process hits a certain metastable state of the system before another and can be viewed as the solution of a backward Kolmogorov equation.

Computing the committor function in high-dimensional settings is a formidable task. Traditional numerical methods such as finite difference and finite element methods become prohibitively expensive in even moderate dimensions. To overcome the curse of dimensionality, significant efforts have been expended to apply deep learning framework to solve for high dimensional partial differential equations [55,44,15,20,38]. Most recently [19,28,27] have suggested representing the committor function using neural networks. Some of these approaches rely on sampling from the equilibrium distribution and so work well when the transitions are easily observed. For, e.g., chemical systems at low temperature the committor function can change sharply between the two metastable states, and transitions are rare and difficult to sample. To address this problem, [41] has proposed an adaptive importance sampling scheme. Meanwhile, the dynamical Galerkin framework

* Corresponding authors.

E-mail addresses: yianc@uchicago.edu (Y. Chen), jeremyhoskins@uchicago.edu (J. Hoskins), ykhoo@uchicago.edu (Y. Khoo), lindsey@berkeley.edu (M. Lindsey).

for computing committor functions [45], which represents functions in a basis rather than as neural networks, approaches the sampling problem by initializing short trajectories uniformly according to known reaction coordinates.

Tensor network methods [8,35,51,34,9,22] have emerged as an alternative to neural networks as a tool for high-dimensional problems in modern quantum physics and beyond. Typical tensor decomposition methods include tensor trains [36] (also known as linear tensor networks or matrix product states [1,37,56]), the CP decomposition [16], and the Tucker/Hierarchical Tucker decomposition [16,47,14]. These methods approximate tensors in compressed, structured formats that enable efficient linear algebra operations. More details can be found in [23,13,12,21]. Moreover, tensor network methods have also been applied to solve for high-dimensional partial differential equations [22,2].

In this paper we propose a novel approach to computing committor functions based on matrix product states/tensor trains. Specifically, we approximate both the equilibrium probability distribution and the committor function using tensor trains, achieving good performance even for high-dimensional problems in the low-temperature regime. This new approach fully bypasses the aforementioned difficulties due to sampling and establishes an alternative method for studying rare transition events between metastable states in complex, high-dimensional systems.

The rest of the paper is organized as follows. The committor function and its properties are reviewed in Section 2. Therein we also explain how the boundary condition can be accommodated within our tensor format and provide a summary of relevant tensor network methods. We introduce the key ingredients of our proposed method in Section 3. Numerical experiments for two representative classes of examples are presented in Section 4, demonstrating the accuracy and efficiency of the proposed algorithm. Finally, in Section 5 we summarize our findings.

2. Background and preliminaries

In this section we first review the motivation for computing committor functions, and summarize challenges and recent advances relevant to this task. We then briefly discuss tensor train decompositions, introduce the basic tensor operations, and define relevant associated notations used in this work. Throughout the paper, we use MATLAB notation for multidimensional array indexing.

2.1. Committor functions

The underlying stochastic process of interest is the overdamped Langevin process, defined by

$$d\mathbf{X}_t = -\nabla V(\mathbf{X}_t) dt + \sqrt{2\beta^{-1}} d\mathbf{W}_t, \quad (2.1)$$

where $\mathbf{X}_t \in \Omega \subset \mathbb{R}^d$ is the state of the system, $V : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$ is a smooth potential energy function, $\beta = 1/T$ is the inverse of the temperature T , and \mathbf{W}_t is a d -dimensional Wiener process. If the potential energy function V is confining for Ω (see, e.g., [5, Definition 4.2]), then one can show that the equilibrium probability distribution of the Langevin dynamics (2.1) is the Boltzmann-Gibbs distribution

$$p(\mathbf{x}) = \frac{1}{Z_\beta} \exp(-\beta V(\mathbf{x})), \quad (2.2)$$

where $Z_\beta = \int_\Omega \exp(-\beta V(\mathbf{x})) d\mathbf{x}$ is the partition function. We are interested in the transition between two simply connected domains $A, B \subset \Omega$ with smooth boundaries. The associated committor function $q : \Omega \rightarrow [0, 1]$ is defined by

$$q(\mathbf{x}) = \mathbb{P}(\tau_B < \tau_A \mid \mathbf{X}_0 = \mathbf{x}), \quad (2.3)$$

where τ_A and τ_B are the hitting times for the sets A and B , respectively. The committor function q provides a useful statistical description of properties such as the density and probability of reaction trajectories [48,52,39]. However, computing the committor function can be a formidable task since it involves solving the following (possibly high-dimensional) backward Kolmogorov equation with Dirichlet boundary conditions:

$$-\beta^{-1} \Delta q(\mathbf{x}) + \nabla V(\mathbf{x}) \cdot \nabla q(\mathbf{x}) = 0 \text{ in } \Omega \setminus (A \cup B), \quad q(\mathbf{x})|_{\partial A} = 0, \quad q(\mathbf{x})|_{\partial B} = 1. \quad (2.4)$$

For high-dimensional problems, traditional methods such as finite difference and finite element discretization are intractable. Numerous alternative methods can effectively approximate the committor function under the assumption that the transition paths from A to B are localized in a quasi-one-dimensional reaction tube or low-dimensional manifold. For example, the finite temperature string method [53,49] approximates the isosurfaces of the committor function with hyperplanes normal to the most probable transition paths. The diffusion map approach [7] aims to obtain the committor function on a set of points by applying point cloud discretization to the generator $L = -\beta^{-1} \Delta + \nabla V(\mathbf{x}) \cdot \nabla$. The method presented in [24] improves the diffusion map approach by discretizing L using a finite element method on local tangent planes of the point cloud. Also an alternative approach was introduced by solving a potential function instead, see [31] for details.

To compute the committor function, a classic approach is to solve the variational problem

$$\arg \min_q \int_{\Omega} |\nabla q(\mathbf{x})|^2 p(\mathbf{x}) d\mathbf{x}, \quad q(\mathbf{x})|_{\partial A} = 0, \quad q(\mathbf{x})|_{\partial B} = 1, \quad (2.5)$$

for which (2.4) is the Euler-Lagrange equation, see for example [32,19,28,27,41]. Specifically [19,28,27] proposed to parametrize the committor function q using neural networks. In order to obtain an unconstrained optimization problem, the boundary conditions are enforced in [19,27,41] by adding two extra penalty terms, as in

$$\arg \min_q \int_{\Omega} |\nabla q(\mathbf{x})|^2 p(\mathbf{x}) d\mathbf{x} + \rho \int_{\partial A} q(\mathbf{x})^2 p_{\partial A}(\mathbf{x}) d\mathbf{x} + \rho \int_{\partial B} (q(\mathbf{x}) - 1)^2 p_{\partial B}(\mathbf{x}) d\mathbf{x}, \quad (2.6)$$

where $p_{\partial A}$ and $p_{\partial B}$ define probability measures supported on the boundaries ∂A and ∂B respectively. In all of these works, the objective is evaluated and optimized via stochastic sampling. By contrast, in this work, we propose to represent the committor function q in a tensor train format, which will allow for optimization via stable and efficient deterministic linear algebra operations.

Since the potential function V is confining, we can effectively restrict our domain of interest to a bounded subset of Ω . Outside of this subset, the density is small and contributes only negligibly to the variational cost (2.5). For simplicity we shall identify Ω with this subset and assume $\Omega = \Omega_1 \times \Omega_2 \times \dots \times \Omega_d$ where each $\Omega_i \in \mathbb{R}$ is a bounded subset.

2.2. Soft boundary condition

Unfortunately, the formulation (2.6) is not immediately amenable to optimization within a tensor format for q . The reason is that the surface measures on ∂A and ∂B cannot themselves be identified with functions on Ω , much less than functions that can be compressed in tensor format, so the penalty terms cannot simply be viewed as inner products of tensor trains.

Therefore we instead consider an objective of the form

$$\arg \min_q \int_{\Omega} |\nabla q(\mathbf{x})|^2 p(\mathbf{x}) d\mathbf{x} + \rho \int_{\Omega} q(\mathbf{x})^2 p_A(\mathbf{x}) d\mathbf{x} + \rho \int_{\Omega} (q(\mathbf{x}) - 1)^2 p_B(\mathbf{x}) d\mathbf{x}. \quad (2.7)$$

Here p_A and p_B are probability densities that are absolutely continuous with respect to the Lebesgue measure on Ω .

In fact, we show in Appendix A that the exact optimizer of (2.7) admits a probabilistic interpretation similar to that of the usual committor function. As such we call the optimizer a ‘soft committor function.’ Specifically, the interpretation is based on a modification of the Langevin dynamics (2.1) in which one augments the state space Ω with two ‘cemetery states’ c_A and c_B . The process jumps randomly to these two states with instantaneous jump rates $\frac{\rho \cdot p_A}{\beta \cdot p}$ and $\frac{\rho \cdot p_B}{\beta \cdot p}$, respectively. The soft committor function evaluates the probability that the modified process hits c_B before c_A . This formulation is rather similar to the Poisson type equation introduced in [31] with some differences in the right hand side of the equation.

From a different perspective, when ρ is large and p_A and p_B concentrate near ∂A and ∂B , respectively, the soft committor function can be viewed as an approximation of the ordinary committor function. In fact, if p_A and p_B are Gaussian densities, then in high dimensions [6], p_A and p_B each weakly approximates a uniform measure on a suitable hypersphere. This is convenient because A and B are often chosen to be balls and the Gaussian densities have exact tensor train representations. Section 3.3 provides two examples on the construction of p_A and p_B using Gaussian densities. In practice we choose p_A and p_B such that they are well-representable in tensor train format (introduced in Section 2.3). More details on the construction of p_A and p_B will be provided below in Section 3.

For simplicity, in what follows we will simply refer to soft committor functions as committor functions.

2.3. Tensors and tensor networks

In this subsection we summarize the basic tensor operations used in this work. In particular, for ease of exposition we introduce tensor network diagram notation, which provides a convenient way of visually describing tensor operations. We also introduce the matrix product state/tensor train format for parametrizing high-dimensional functions.

In tensor diagrams, a tensor is represented by a node, where the number of incoming legs indicates the dimensionality of the tensor, i.e., the number of indices/arguments. There are two types of leg: legs indicating continuous arguments are denoted by dashed lines, and legs indicating discrete indices are denoted by solid lines. For example, Fig. 2.1 (a) shows the tensor diagram for a 3-tensor \mathcal{A} and a 2-tensor \mathcal{B} , which can be viewed as two functions

$$\mathcal{A}(x_1, i_2, i_3), \quad \mathcal{B}(j_1, x_2), \quad (2.8)$$

respectively, where x_1, x_2 are continuous variables and i_2, i_3, j_1 are discrete variables.

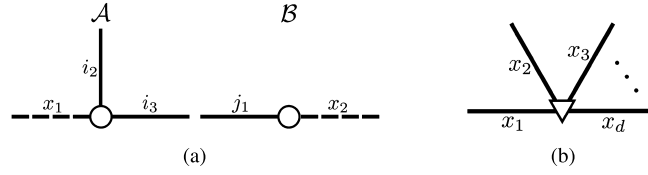


Fig. 2.1. (a) Tensor diagrams for a 3-tensor \mathcal{A} and a 2-tensor \mathcal{B} . (b) Tensor diagram for a d -dimensional Kronecker Delta node. Solid lines correspond to discrete variables, and dashed lines correspond to continuous variables.

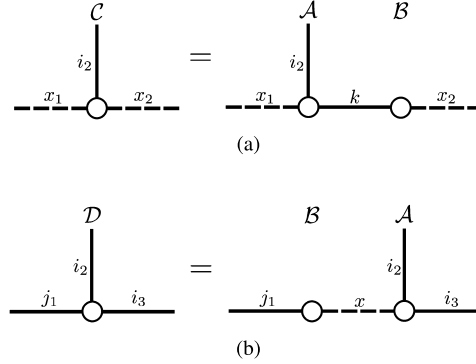


Fig. 2.2. (a) Tensor contraction of discrete legs. (b) Tensor contraction of continuous legs.

We also define the multi-dimensional Kronecker delta tensor (depicted by an inverted triangular node as in Fig. 2.1 (b)):

$$\delta(x_1, x_2, \dots, x_d) = \begin{cases} 1 & \text{if } x_1 = x_2 = \dots = x_d. \\ 0 & \text{otherwise.} \end{cases} \quad (2.9)$$

By a slight abuse of notation, we will use the same symbol to represent the appropriate Dirac delta function when the legs represent continuous variables.

Next we describe a key operation called tensor contraction. This operation is indicated visually by joining legs from different tensors. For example, in Fig. 2.2 (a), the third leg of \mathcal{A} is joined with the first leg of \mathcal{B} . This corresponds to the computation

$$\mathcal{C}(x_1, i_2, x_2) = \sum_k \mathcal{A}(x_1, i_2, k) \mathcal{B}(k, x_2), \quad (2.10)$$

where it is implicitly assumed that the indices of the joined legs have the same range. Here x_1, x_2 are continuous variables, and i_2 is a discrete variable.

Tensor contraction can be defined for continuous legs as well. For example, in Fig. 2.2 (b), continuous legs of \mathcal{A} and \mathcal{B} are contracted, corresponding to the operation

$$\mathcal{D}(j_1, i_2, i_3) = \int_{\Omega_0} \mathcal{B}(j_1, x) \mathcal{A}(x, i_2, i_3) dx, \quad (2.11)$$

for some suitable domain Ω_0 , which is implicitly assumed to be the domain of both joined legs. The resulting tensor \mathcal{D} is a 3-tensor with only discrete legs.

A tensor network diagram consists of a collection of individual tensor diagrams with some pairs of legs joined, i.e., contracted. The contracted legs correspond to the so-called ‘internal indices’ for the tensor network, while the uncontracted legs correspond to ‘external indices,’ which are the indices remaining after all of the indicated contractions have been performed.

Next we introduce several low-complexity tensor networks and their corresponding diagrams. A matrix product state (MPS) or tensor train (TT) is a factorization of a d -tensor into a chain-like product of 3-tensors. Such a factorization allows one to approximate high-dimensional tensors and manipulate them efficiently, typically with $\mathcal{O}(d)$ time and memory complexity.

Definition 1. Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ be a d -tensor, with entries indexed by (i_1, i_2, \dots, i_d) . Then we say that \mathcal{A} is a MPS/TT with ranks $\mathbf{r} = (r_0, \dots, r_d)$, where we fix $r_0 = r_d = 1$ by convention, if one can write

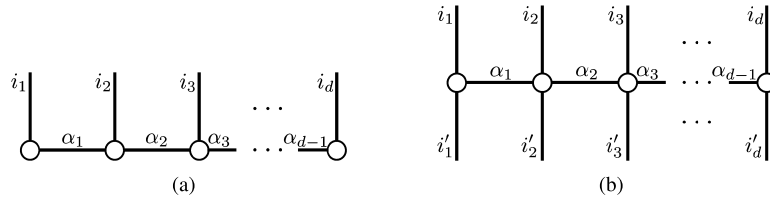


Fig. 2.3. (a) d -dimensional TT/MPS. (b) d -dimensional MPO. The indices depicted match the expressions in (2.12) and (2.14). Note that we omit the legs for the trivial indices α_0 and α_d .

$$\begin{aligned} \mathcal{A}(i_1, i_2, \dots, i_d) &= \sum_{\alpha_0=1}^{r_0} \cdots \sum_{\alpha_d=1}^{r_d} \mathcal{G}_1(\alpha_0, i_1, \alpha_1) \mathcal{G}_2(\alpha_1, i_2, \alpha_2) \cdots \mathcal{G}_d(\alpha_{d-1}, i_d, \alpha_d) \\ &= \mathcal{G}_1(:, i_1, :) \mathcal{G}_2(:, i_2, :) \cdots \mathcal{G}_d(:, i_d, :) \end{aligned} \quad (2.12)$$

for all (i_1, i_2, \dots, i_d) . Here $\mathcal{G}_k(:, i_k, :) \in \mathbb{R}^{r_{k-1} \times r_k}$ is viewed as a matrix for each $k = 1, \dots, d$, and the matrix product in (2.12) is a 1×1 matrix, i.e., a scalar value. The 3-tensor $\mathcal{G}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$ is called the k -th tensor core of \mathcal{A} .

In tensor diagrams, an MPS/TT is represented by a chain of 3-tensors, as in Fig. 2.3 (a). Note that the 0-th and last tensor cores can be viewed as 2-tensors since $r_0 = r_d = 1$, and as such the corresponding legs can be omitted from the diagram.

A matrix product operator (MPO) is a tensor network in which each constituent tensor has two external, uncontracted legs as well as two internal indices contracted with neighboring tensors in a chain-like fashion. Concretely, an MPO is a tensor $\mathcal{O} \in \mathbb{R}^{(m_1 \times m_2 \times \cdots \times m_d) \times (n_1 \times n_2 \times \cdots \times n_d)}$ that can be written in the form

$$\mathcal{O}(i_1, \dots, i_d; i'_1, \dots, i'_d) = \sum_{\alpha_0, \dots, \alpha_d} \mathcal{G}_1(\alpha_0, i_1, i'_1, \alpha_1) \cdots \mathcal{G}_d(\alpha_{d-1}, i_d, i'_d, \alpha_d) \quad (2.13)$$

$$= \mathcal{G}_1(:, i_1, i'_1, :) \cdots \mathcal{G}_d(:, i_d, i'_d, :), \quad (2.14)$$

where the sum over α_k has a range defined by a corresponding rank r_k , as in Definition 1, and $r_0 = r_d = 1$ by convention. In this case we similarly say that the MPO has ranks $\mathbf{r} = (r_0, \dots, r_d)$. The 4-tensor $\mathcal{G}_k \in \mathbb{R}^{r_{k-1} \times m_k \times n_k \times r_k}$ is called the k -th tensor core. A corresponding tensor network diagram is shown in Fig. 2.3 (b).

For further background on tensor networks and diagrams, see [35].

3. Proposed method

In this work we obtain the committor function by solving the variational problem (2.7) within a MPS/TT parametrization for the committor function q . We demonstrate that by approximating the equilibrium probability density p in MPS/TT format, this optimization problem can be solved using basic tensor operations. In particular the minimization is accomplished using a standard alternating least squares approach.

3.1. Discretizing the variational problem

We will represent the unknown committor function in a tensor product basis according to the product structure of the domain $\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_d$. Within this basis, we will approach the variational problem (2.7) by Galerkin approximation.

To begin, suppose that we have an orthogonal basis for each $L^2(\Omega_k)$, denoted by $\{\phi_j^{(k)}\}_{j=1}^\infty$. In order to obtain a finite-dimensional problem we consider the subspace of $L^2(\Omega_k)$ spanned by only the first $L^{(k)}$ basis functions. Here the $L^{(k)}$, $k = 1, \dots, d$, are a set of positive integers which are either fixed or determined adaptively. Then given the finite basis $\{\phi_j^{(k)}\}_{j=1}^{L^{(k)}}$, which spans a subspace of $L^2(\Omega_k)$ for each $k = 1, \dots, d$, we can consider an expansion of q in the corresponding tensor product basis:

$$\begin{aligned} q(\mathbf{x}) &= \sum_{i_1, \dots, i_d} \mathcal{Q}(i_1, \dots, i_d) \phi_{i_1}^{(1)}(x_1) \cdots \phi_{i_d}^{(d)}(x_d), \\ &:= \sum_{\mathbf{i}} \mathcal{Q}(\mathbf{i}) \phi_{i_1}^{(1)}(x_1) \cdots \phi_{i_d}^{(d)}(x_d), \end{aligned} \quad (3.1)$$

where, for notational convenience, we have defined $\mathbf{i} := (i_1, i_2, \dots, i_d)$ and $\mathbf{x} := (x_1, x_2, \dots, x_d)$. Additionally, we set $\phi^{(k)} := (\phi_j^{(k)})_{j=1}^{L^{(k)}}$. Each $\phi^{(k)}$ can be viewed as a 2-tensor via $\phi^{(k)}(j, x) = \phi_j^{(k)}(x)$, where the first index is the basis function index and the second (continuous) index is a spatial coordinate. Then the decomposition (3.1) for q can be depicted graphically as in Fig. 3.1.

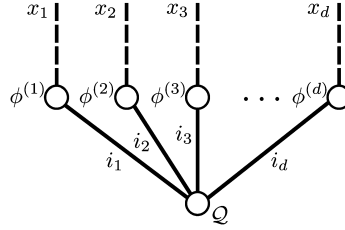


Fig. 3.1. Tensor diagram for the decomposition (3.1) of the committor function.

To determine the committor function q within the truncated tensor product basis, we want to determine the coefficient tensor \mathcal{Q} that is optimal in the sense of (2.7). By inserting the parameterization (3.1) into the variational problem (2.7), we can rewrite the optimization problem as follows,

$$\arg \min_{\mathcal{Q}} \underbrace{\sum_{k=1}^d \sum_{\mathbf{i}, \mathbf{j}} H^k(\mathbf{i}; \mathbf{j}) \mathcal{Q}(\mathbf{i}) \mathcal{Q}(\mathbf{j})}_{\int_{\Omega} |\nabla q(\mathbf{x})|^2 p(\mathbf{x}) d\mathbf{x}} + \underbrace{\rho \sum_{\mathbf{i}, \mathbf{j}} H^A(\mathbf{i}; \mathbf{j}) \mathcal{Q}(\mathbf{i}) \mathcal{Q}(\mathbf{j})}_{\rho \int_{\Omega} q(\mathbf{x})^2 p_A(\mathbf{x}) d\mathbf{x}} + \underbrace{\rho \sum_{\mathbf{i}, \mathbf{j}} H^B(\mathbf{i}; \mathbf{j}) \mathcal{Q}(\mathbf{i}) \mathcal{Q}(\mathbf{j}) - 2\rho \sum_{\mathbf{i}} \mathcal{Q}(\mathbf{i}) h^B(\mathbf{i}) + \rho}_{\rho \int_{\Omega} (q(\mathbf{x}) - 1)^2 p_B(\mathbf{x}) d\mathbf{x}}, \quad (3.2)$$

where

$$H^k(\mathbf{i}; \mathbf{j}) = \int_{\Omega} \frac{\partial}{\partial \mathbf{x}_k} [\phi_{i_1}^{(1)}(x_1) \dots \phi_{i_d}^{(d)}(x_d)] \frac{\partial}{\partial \mathbf{x}_k} [\phi_{j_1}^{(1)}(x_1) \dots \phi_{j_d}^{(d)}(x_d)] p(\mathbf{x}) d\mathbf{x} \quad (3.3)$$

$$H^A(\mathbf{i}; \mathbf{j}) = \int_{\Omega} \phi_{i_1}^{(1)}(x_1) \dots \phi_{i_d}^{(d)}(x_d) \phi_{j_1}^{(1)}(x_1) \dots \phi_{j_d}^{(d)}(x_d) p_A(\mathbf{x}) d\mathbf{x} \quad (3.4)$$

$$H^B(\mathbf{i}; \mathbf{j}) = \int_{\Omega} \phi_{i_1}^{(1)}(x_1) \dots \phi_{i_d}^{(d)}(x_d) \phi_{j_1}^{(1)}(x_1) \dots \phi_{j_d}^{(d)}(x_d) p_B(\mathbf{x}) d\mathbf{x} \quad (3.5)$$

$$h^B(\mathbf{i}) = \int_{\Omega} \phi_{i_1}^{(1)}(x_1) \phi_{i_2}^{(2)}(x_2) \dots \phi_{i_d}^{(d)}(x_d) p_B(\mathbf{x}) d\mathbf{x}. \quad (3.6)$$

Here $(\mathbf{i}; \mathbf{j}) = (i_1, \dots, i_d; j_1, \dots, j_d)$ is a concatenation of multi-indices. We can simply ignore the last constant ρ since it does not affect the minimizer. Computing the tensors $\{H^k\}_{k=1}^d$, H^A , H^B , and h^B is *prima facie* intractable as it requires us to perform integration over the d -dimensional domain Ω , in addition to storing tensors of exponential size in d . Moreover, the number of unknown tensor entries of \mathcal{Q} is also exponential in d . Traditional approaches are therefore prohibitively expensive for d of even moderate size.

In the next two sections we show how to use MPS/TT approximations to obtain $\{H^k\}_{k=1}^d$, H^A , H^B , h^B , allowing us to solve the optimization problem (3.2) with computational and storage complexities of $\mathcal{O}(d)$.

3.2. Constructing H^k

In this subsection we detail the construction of H^k , which corresponds to the variational energy term $\int_{\Omega} |\nabla q(\mathbf{x})|^2 p(\mathbf{x}) d\mathbf{x}$ of (2.7). As mentioned above, in order to obtain each H^k in (3.3), one needs to evaluate a d -dimensional integral and store the resulting high-dimensional tensor. To circumvent the exponential complexity in d , we assume that the equilibrium density p can be approximated as an MPS/TT as follows:

$$p(\mathbf{x}) = \sum_{\substack{m_1, \dots, m_d \\ \alpha_0, \dots, \alpha_d}} \mathcal{P}_1(\alpha_0, m_1, \alpha_1) \dots \mathcal{P}_d(\alpha_{d-1}, m_d, \alpha_d) \psi_{m_1}^{(1)}(x_1) \dots \psi_{m_d}^{(d)}(x_d), \quad (3.7)$$

where $\psi^{(k)} := (\psi_j^{(k)})_{j=1}^{K^{(k)}}$ is a vector of univariate basis functions $\Omega_i \rightarrow \mathbb{R}$. Fig. 3.2 illustrates the structure of the equilibrium density p that we assume in this paper.

The construction of the MPS/TT format for a given equilibrium density p will be described in Section 4 in the contexts of specific example problems.

Such an approximation of p amounts to changing the tensor representation of H^k depicted graphically in Fig. 3.3 (a) to the representation in Fig. 3.3 (b). Note that these calculations involve the derivatives of our univariate basis functions $\phi^{(k)}$. In our figures, we use a hollow node to represent the vector of basis functions $\phi^{(k)}$ and a filled node to represent its vector

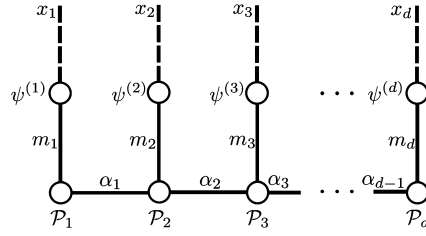


Fig. 3.2. Tensor diagram for the approximate equilibrium density p (3.7).

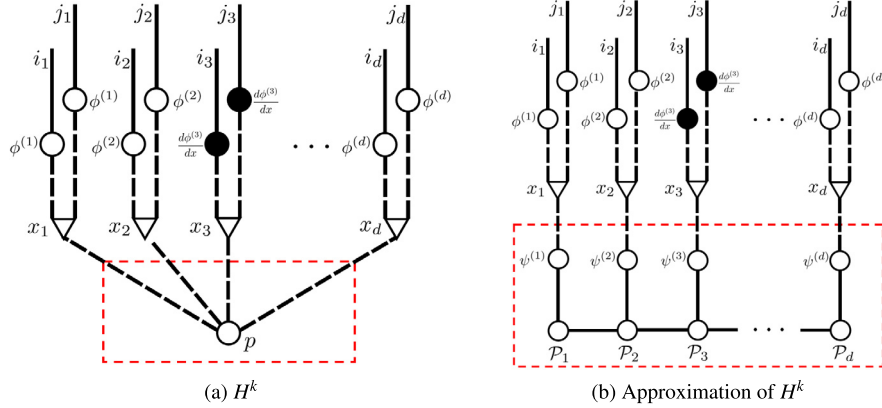


Fig. 3.3. (a) Tensor diagram for H^k , $k=3$, as defined in (3.3). (b) Approximation of H^k obtained by replacing p with its MPS/TT approximation (3.7). We use red dashed boxes to indicate the region of replacement. The original basis functions $\phi^{(l)}$, $l \neq k$, are represented using hollow nodes, and the derivative $d\phi^{(k)}/dx$ is distinguished using a filled node.

of derivatives, i.e., $d\phi^{(k)}/dx$. We observe that the $\{H^k\}_{k=1}^d$ are naturally viewed as MPOs, and moreover the construction of these MPOs can be performed using basic tensor algebra in $\mathcal{O}(d)$ complexity. More precisely:

- (1) To construct an MPO for H^k following Fig. 3.3 (b), note that we need to perform two types of tensor contraction: one involving the original basis functions $\phi^{(l)}$, $l \neq k$, and the other involving the derivatives $d\phi^{(k)}/dx$. Therefore we precompute these two contractions, which can be recycled to form MPOs for the H^k , $k=1, \dots, d$. First, we form I_l , $l=1, \dots, d$ by contracting three tensors $\psi^{(l)}$, $\phi^{(l)}$, $\phi^{(l)}$ and form \tilde{I}_l , $l=1, \dots, d$ by contracting three tensors $\psi^{(l)}$, $d\phi^{(l)}/dx$, $d\phi^{(l)}/dx$. The tensors I_l and \tilde{I}_l are defined graphically in Fig. 3.4 (a) and (b). These contractions can be performed by univariate numerical integration. Next we contract each I_l with the corresponding tensor core \mathcal{P}_l to obtain H_l for $l=1, \dots, d$. Similarly we contract \tilde{I}_l with \mathcal{P}_l to obtain \tilde{H}_l . These constructions are illustrated in Fig. 3.4 (a) and (b), respectively.
- (2) Next we assemble H^k by substituting H_l , \tilde{H}_l , $l=1, \dots, d$ into the red boxes in Fig. 3.4 (c) as needed. This yields an MPO as shown in Fig. 3.4 (c) on the right. We denote l -th tensor core of H^k by H_l^k .
- (3) Finally we repeat step (2) for all $k=1, \dots, d$.

The algorithm outputs core tensors H_1^k, \dots, H_d^k for the MPO H^k , $k=1, \dots, d$. The computational complexity of forming each pair I_l, \tilde{I}_l is $\mathcal{O}(K^{(l)}L^{(l)2})$ since a univariate numerical integration is performed for each entry. Hence the cost of computing all of the I_l, \tilde{I}_l is $\mathcal{O}(dKL^2)$, where we define $K := \max_l K^{(l)}$, $L := \max_l L^{(l)}$.

If we assume that the MPS/TT format for p has ranks $\mathbf{r} = (r_0, r_1, r_2, \dots, r_d)$, and set $r = \max_l r_l$, then the contraction steps with the \mathcal{P}_l altogether cost $\mathcal{O}(dr^2KL^2)$. Therefore to construct the tensor cores H_1^k, \dots, H_d^k , the total computational cost is $\mathcal{O}(dr^2KL^2)$. The memory complexity, including that of storing the intermediate tensors $I_l, \tilde{I}_l, H_l, \tilde{H}_l$, is $\mathcal{O}(dKL^2 + dr^2L^2)$.

Remark 1. Our approach relies on the assumption that the equilibrium density of the system can be efficiently represented or approximated in MPS/TT format. The MPS/TT structure makes it possible to perform numerical integration for each individual tensor core, thus resulting in a computational complexity that is linear in d . When the assumptions are violated, the TT rank of the equilibrium density can grow with d , indicating the true computational complexity can be higher than our analysis.

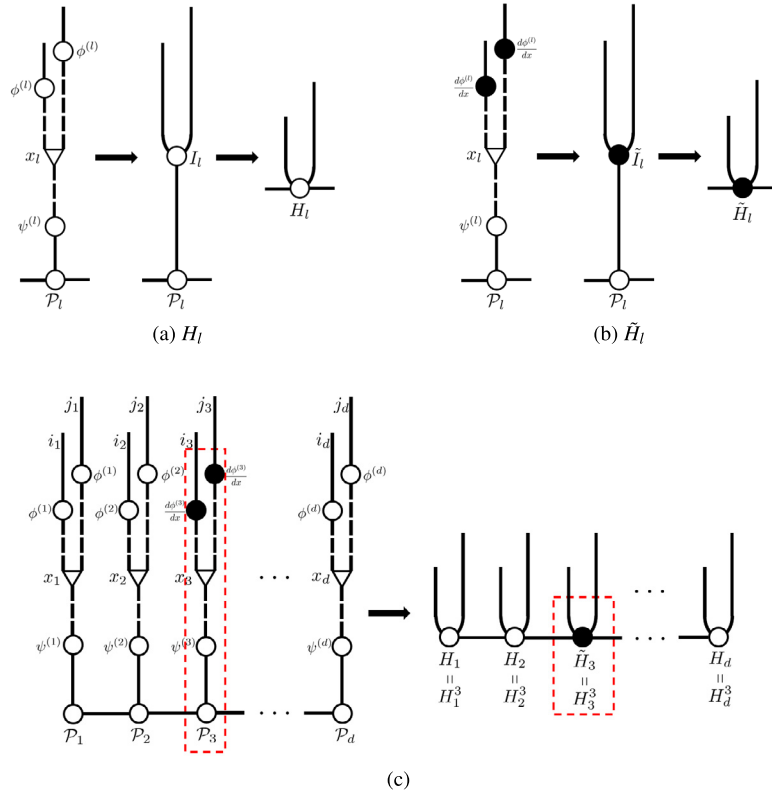


Fig. 3.4. Diagrammatic illustration of the construction of an MPO format for H^k . (a) Precompute tensors H_l . (b) Precompute tensors \tilde{H}_l . (c) Assemble H^k , $k = 3$, by substituting tensors H_l and \tilde{H}_l . We use red dashed boxes to indicate the region of replacement. A similar construction is repeated for other k . (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3.3. Constructing H^A and H^B

The tensors H^A and H^B derive from the two soft boundary penalty terms $\rho \int_{\Omega} q(\mathbf{x})^2 p_A(\mathbf{x}) d\mathbf{x}$ and $\rho \int_{\Omega} (q(\mathbf{x}) - 1)^2 p_B(\mathbf{x}) d\mathbf{x}$ in (2.7). The construction of the MPO format for H^A and H^B is similar to that of H^k detailed above in Section 3.2. However, we now further need to represent the soft boundary measures p_A and p_B as MPS/TT. Recall our motivation that p_A and p_B weakly approximate surface measures on the boundaries ∂A and ∂B , though for any choice of p_A and p_B we may still interpret the soft committor function probabilistically following Appendix A. The construction of approximate surface measures varies depending on the specific geometry of A and B . In many applications, A and B are balls or half-spaces, so ∂A and ∂B are spheres or hyperplanes. We discuss these two cases in detail presently.

For the case of a sphere, the Gaussian annulus theorem [6, Theorem 2.8] indicates that most of the mass of a high-dimensional Gaussian distribution concentrates on a shell. If we assume that region A is a d -dimensional ball with center \mathbf{x}_A and radius R_A , we can approximate the uniform measure on ∂A by a Gaussian density,

$$p_A(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\sigma^d} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_A\|^2}{2\sigma^2}\right), \text{ where } \sigma = \frac{R_A}{\sqrt{d}}. \quad (3.8)$$

More precisely, under this probability measure, we have that

$$\text{Prob}\left(\|\mathbf{x}\|_2 - \sqrt{d}\sigma \geq t\sigma\right) \leq \exp\left(-\frac{t^2}{\kappa}\right), \text{ for all } t > 0, \quad (3.9)$$

where $\kappa > 0$ is a constant [50]. This bound indicates that the mass of p_A concentrates on a shell with radius R_A and thickness $O(1/\sqrt{d})$. It is straightforward to convert (3.8) into MPS/TT format since it is in fact a pure tensor product of univariate functions of each scalar variable x_k . As such our resulting MPS/TT should have ranks all equal to 1.

For the case of a hyperplane, suppose in particular that $\partial A = \{\mathbf{x} \in \mathbb{R}^d : x_i = c\}$. In this case consider

$$p_A(\mathbf{x}) = \frac{1}{(2\pi)^{1/2}\sigma} \exp\left(-\frac{(x_i - c)^2}{2\sigma^2}\right), \quad (3.10)$$

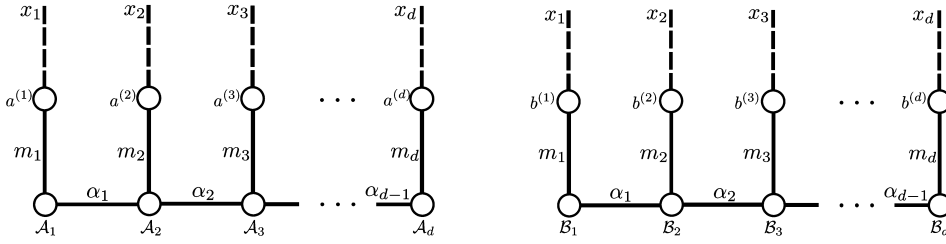


Fig. 3.5. Tensor diagrams for (a) p_A as in (3.11) and (b) p_B as in (3.12).

where σ controls the sharpness of the approximation. This choice of p_A is again a pure tensor product of univariate functions.

Now suppose that we have p_A and p_B in MPS/TT form. Specifically, assume that we have tensor cores $\{\mathcal{A}_i\}_{i=1}^d$ with associated ranks $\mathbf{s} = (s_0, s_1, \dots, s_d)$, together with a vector of basis functions $a^{(k)} := (a_j^{(k)})_{j=1}^{K_A^{(k)}}$ for each $k = 1, \dots, d$. Also assume that we have tensor cores $\{\mathcal{B}_i\}_{i=1}^d$ with ranks $\mathbf{t} = (t_0, t_1, \dots, t_d)$ and a vector of basis functions $b^{(k)} := (b_j^{(k)})_{j=1}^{K_B^{(k)}}$. Then we assume that we can write p_A and p_B as

$$p_A(\mathbf{x}) = \sum_{\substack{m_1, \dots, m_d \\ \alpha_0, \dots, \alpha_d}} \mathcal{A}_1(\alpha_0, m_1, \alpha_1) \dots \mathcal{A}_d(\alpha_{d-1}, m_d, \alpha_d) a_{m_1}^{(1)}(x_1) \dots a_{m_d}^{(d)}(x_d). \quad (3.11)$$

$$p_B(\mathbf{x}) = \sum_{\substack{m_1, \dots, m_d \\ \alpha_0, \dots, \alpha_d}} \mathcal{B}_1(\alpha_0, m_1, \alpha_1) \dots \mathcal{B}_d(\alpha_{d-1}, m_d, \alpha_d) b_{m_1}^{(1)}(x_1) \dots b_{m_d}^{(d)}(x_d). \quad (3.12)$$

In Fig. 3.5 we illustrate these formats graphically.

Now in light of the resemblance among (3.3), (3.4), and (3.5), we can use the same procedure described in Section 3.2 to approximate H^A and H^B as MPOs. To wit, we simply replace p in Fig. 3.3 with the MPS/TT approximations of $p_{\partial A}$ or $p_{\partial B}$ and replace all derivatives $d\phi^{(l)}/dx$ by $\phi^{(l)}$ since there are no derivatives in (3.4) and (3.5). Ultimately we obtain MPO formats for H_A and H_B with ranks \mathbf{s} and \mathbf{t} and cores H_k^A and H_k^B , $k = 1, \dots, d$ respectively. The computational complexities of constructing H^A and H^B are $\mathcal{O}(ds^2 K_A L^2)$ and $\mathcal{O}(dt^2 K_B L^2)$, respectively, where we define $s := \max_i s_i$, $t := \max_i t_i$, $K_A := \max_i K_A^{(i)}$, and $K_B := \max_i K_B^{(i)}$. The memory complexities are $\mathcal{O}(dK_A L^2 + ds^2 L^2)$ and $\mathcal{O}(dK_B L^2 + dt^2 L^2)$, respectively.

3.4. Constructing h^B

In this subsection we focus on constructing h^B , which comes from the cross term in the second penalty term $\rho \int_{\Omega} (q(\mathbf{x}) - 1)^2 p_B(\mathbf{x}) d\mathbf{x}$ within (2.7). The ideas are again very similar to Section 3.2. The tensor diagram for h^B is shown in Fig. 3.6 (a). By plugging in the MPS/TT approximation of the soft boundary measure p_B (3.12), we obtain the approximation of h^B illustrated in Fig. 3.6 (b). One can further bring h^B to a standard MPS/TT form, using the contractions shown in Fig. 3.7. In detail, the procedure is as follows:

- (1) In Fig. 3.7 (a), we contract the two connected basis function nodes in the red box. This results in tensors J_k , $k = 1, \dots, d$, seen in Fig. 3.7 (a) on the right. This contraction requires univariate numerical integrations.
- (2) Next we merge the computed tensor J_k and the tensor core \mathcal{B}_k for $k = 1, \dots, d$. The resulting 3-tensors, which are the tensor cores for h^B , are denoted h_k^B . This step is shown in Fig. 3.7 (b).

This procedure yields h^B in MPS/TT format with tensor cores h_1^B, \dots, h_d^B . The computational and memory complexities of constructing h^B are $\mathcal{O}(dt^2 K_B L)$ and $\mathcal{O}(dK_B L + dt^2 L)$, respectively.

3.5. Optimization

We have discussed how the MPS/TT format can be used to compress the tensors $\{H^k\}_{k=1}^d$, H^A , H^B , and h^B . In order to obtain a tractable algorithm for computing the committor function, it is natural to represent the unknown tensor \mathcal{Q} in a compatible format. Indeed, without imposing some additional structure on the parameterization (3.1), the unknown tensor core \mathcal{Q} is still of size exponential in d . Thus we approximate \mathcal{Q} as in MPS/TT format as

$$\mathcal{Q}(\mathbf{i}) := \sum_{\alpha_0, \dots, \alpha_d} \mathcal{Q}_1(\alpha_0, i_1, \alpha_1) \mathcal{Q}_2(\alpha_1, i_2, \alpha_2) \dots \mathcal{Q}_d(\alpha_{d-1}, i_d, \alpha_d). \quad (3.13)$$

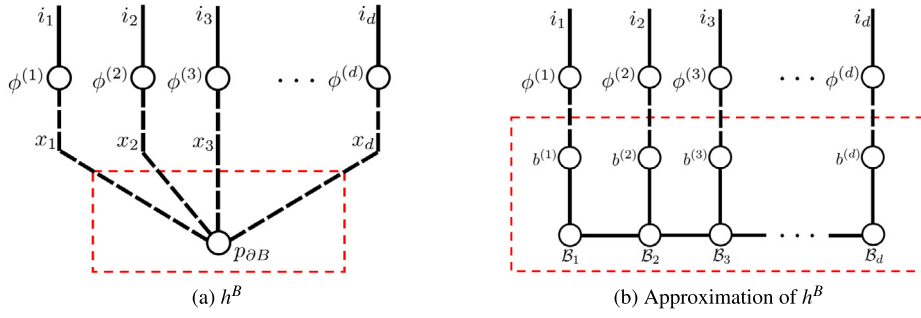


Fig. 3.6. (a) Tensor diagram for h^B as in (3.6). (b) Approximation of h^B obtained by replacing the soft boundary measure p_B with its MPS/TT approximation (3.12). We use red dashed boxes to indicate the region of replacement.

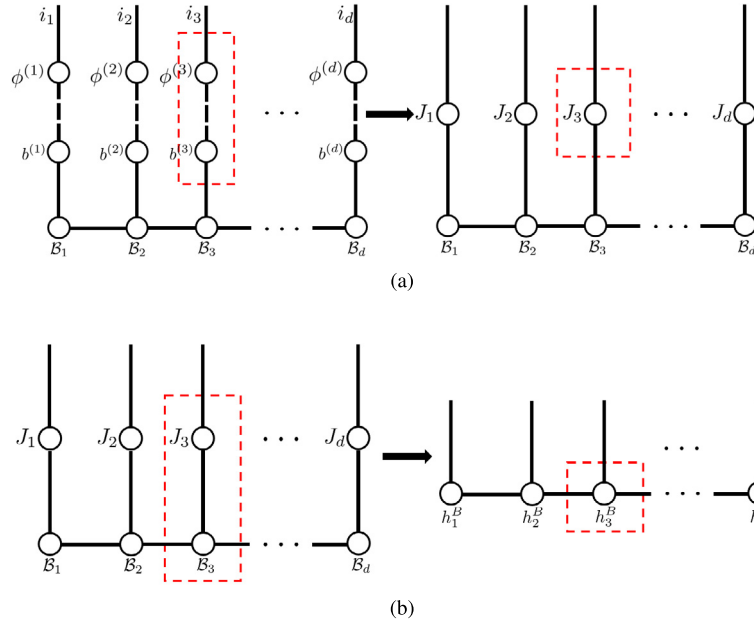


Fig. 3.7. Illustration of the construction of h^B in MPS/TT format. (a) Contract tensors $\phi^{(l)}$ and $b^{(l)}$ to get J_l . (b) Contract tensors J_l and B_l to get h_l^B . We use red dashed boxes emphasize the contractions.

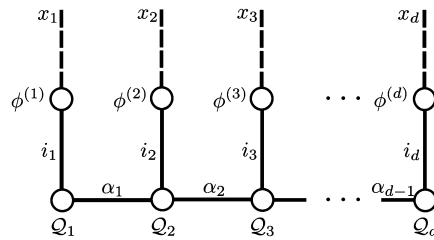


Fig. 3.8. Tensor diagram for the parametrization of q following (3.1) and (3.13).

The tensor diagram for q is shown in Fig. 3.8. Empirically we observe that this format is able to capture the structure of q accurately, i.e., without growth of the ranks of the tensor cores. The MPS/TT format (3.13) for q greatly simplifies the solution of the variational problem (3.2). In Fig. 3.9 we compare the tensor diagram depictions of the original variational problem and the new simplified problem by replacing Q with its MPS/TT approximation. We note that all terms in the simplified form (Fig. 3.9 (b)) can be computed with standard MPO-MPS or MPS-MPS contractions in $\mathcal{O}(d)$ time.

In Fig. 3.9, the unknown tensor cores of MPS Q are marked in red. A standard approach for optimization problems of the form Fig. 3.9 (b) is alternating least squares (ALS). In each ALS iteration, we loop over the dimensions $k = 1, \dots, d$. For each k , we treat all coefficient tensor cores but Q_k as constant. This yields an unconstrained least squares problem for Q_k . Naively the computational complexity is $\mathcal{O}(d^2)$ since the bottleneck is the summation of d terms in Fig. 3.9 (b) and each

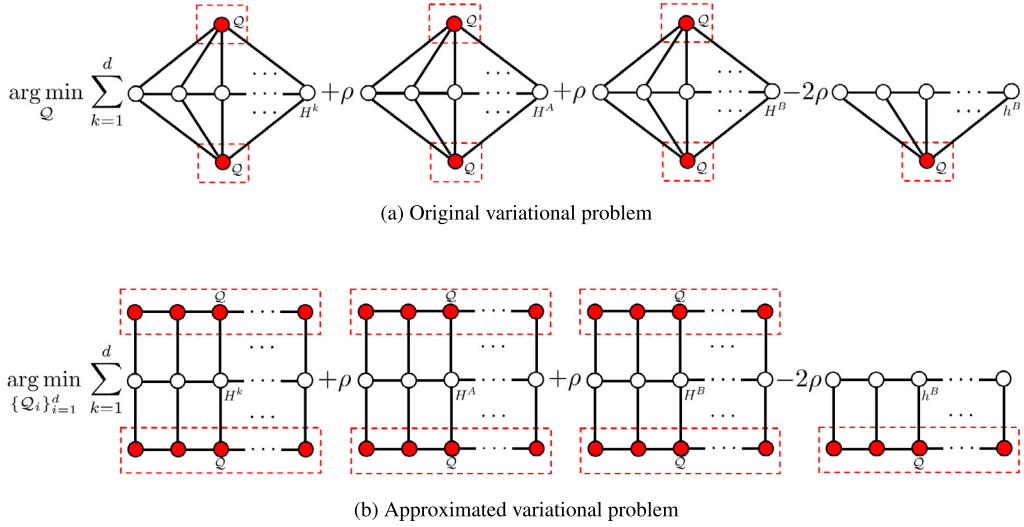


Fig. 3.9. (a) Tensor diagram representation of the variational problem corresponding to (3.2). (b) Approximate variational problem obtained by replacing \mathcal{Q} with its MPS/TT approximation (3.13). We use red dashed boxes to indicate the region of replacement. Specifically the red filled tensor cores are unknown variables in the optimization.

term requires at least $\mathcal{O}(d)$ tensor contractions. However by using the same trick as in the construction of H^k and carefully reusing the computed nodes, one can bring the computational complexity down to $\mathcal{O}(d)$.

Remark 2. Similar to the discussions for the equilibrium density, our complexity analysis relies crucially on the assumption that the TT ranks of the committor function are bounded by a constant as the dimension grows. Here we parametrize the committor function by an MPS/TT with fixed TT rank. In practice we can tune the rank parameter and monitor the numerical rank between the cores. If the rank grows rapidly with the dimensions, our complexity analysis can underestimate the true computational complexity and MPS/TT may not be the most efficient format of parametrizing the true committor function.

4. Numerical experiments

In this section, we present numerical results that demonstrate the accuracy and efficiency of the proposed method.

4.1. Double-well potential

In the first numerical experiment, we consider the following potential

$$V(\mathbf{x}) = (x_1^2 - 1)^2 + 0.3 \sum_{i=2}^d x_i^2, \quad (4.1)$$

and we let A, B be the half-spaces

$$A = \{\mathbf{x} \in \mathbb{R}^d | x_1 \leq -1\}, \quad B = \{\mathbf{x} \in \mathbb{R}^d | x_1 \geq 1\}. \quad (4.2)$$

Now (4.1) is a double-well potential along dimension x_1 , and the two boundaries ∂A and ∂B are located in the potential wells. When the temperature $T = 1/\beta$ is low, the equilibrium density $p \propto e^{-\beta V}$ is concentrated within the two wells. Meanwhile, in this case q is mostly flat with a sharp transition from 0 to 1 at $x_1 = 0$.

For this example, we can compute a ground truth solution. By symmetry, we can obtain the committor function by solving the backward Kolmogorov equation in the first dimension, i.e., setting $q_{\text{true}}(\mathbf{x}) = f(x_1)$, where

$$\frac{d^2 f(x_1)}{dx_1^2} - 4x_1(x_1^2 - 1) \frac{df(x_1)}{dx_1} = 0, \quad f(-1) = 0, \quad f(1) = 1. \quad (4.3)$$

We can solve this ODE numerically using a finite difference method on a very fine grid to produce q_{true} . The performance of our proposed method is evaluated by the following relative error metric

$$E_1 = \frac{\|q - q_{\text{true}}\|_{L^2(\Omega \setminus (A \cup B))}}{\|q_{\text{true}}\|_{L^2(\Omega \setminus (A \cup B))}}, \quad (4.4)$$

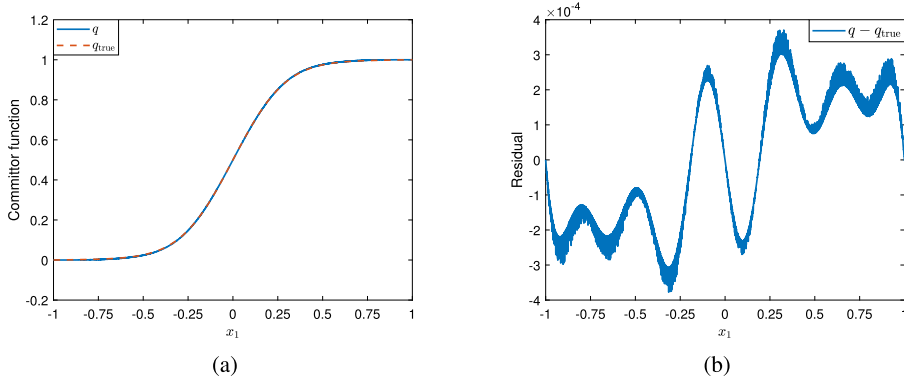


Fig. 4.1. Numerical results for the double-well potential, $T = 0.2$. (a) The numerical solution of the committor function q , compared with the ground truth q_{true} , plotted along the x_1 dimension. (b) The residual plot $q - q_{\text{true}}$ along the x_1 dimension.

where $\|\cdot\|_{L^2(\Omega \setminus (A \cup B))}$ denotes the L^2 -norm with respect to uniform measure over the domain $\Omega \setminus (A \cup B)$. The other error metric we use is

$$E_2 = \frac{\|q - q_{\text{true}}\|_{L^2(\Omega \setminus (A \cup B), p)}}{\|q_{\text{true}}\|_{L^2(\Omega \setminus (A \cup B), p)}}, \quad (4.5)$$

where $\|\cdot\|_{L^2(\Omega \setminus (A \cup B), p)}$ denotes the L^2 -norm with respect to the equilibrium density p defined in (2.2) over the domain $\Omega \setminus (A \cup B)$.

We enforce the boundary conditions by constructing soft boundary measures p_A and p_B in MPS/TT format following (3.10). Meanwhile, we can exactly treat the equilibrium density p in MPS/TT format since it factorizes as a pure tensor product

$$p(\mathbf{x}) = \prod_{k=1}^d p_k(x_k)$$

of univariate functions, given the choice of potential (4.1).

It remains to fix a univariate basis for each dimension of the committor function q . One could of course choose a generic basis such as Chebyshev polynomials, Legendre polynomials, or Fourier series. For this example, however, a better choice is to construct an appropriate truncated orthogonal polynomial basis for each dimension k according to the univariate density p_k .

We solve for the committor function at two representative temperatures $T = 0.2$ and $T = 0.05$ in $d = 20$ dimensions. For $T = 0.2$, we use the first 30 orthogonal polynomials for all dimensions. For the lower temperature $T = 0.05$, we use 60 orthogonal polynomial basis functions since the true committor function changes more sharply near $x_1 = 0$. We show q and q_{true} for $T = 0.2$ in Fig. 4.1 (a) and the corresponding residual $q - q_{\text{true}}$ in Fig. 4.1 (b). Numerical results for $T = 0.05$ are illustrated similarly in Fig. 4.2.

We compute the relative error E_1 in (4.4) using 10^5 uniformly distributed samples between $[-1, 1]$ for x_1 . The relative error is $E_1 = 2.36 \times 10^{-4}$ for $T = 0.2$ and $E_1 = 1.67 \times 10^{-3}$ for $T = 0.05$. With 10^7 samples, we obtain relative error $E_2 = 1.60 \times 10^{-4}$ defined in (4.5) for $T = 0.2$ and $E_2 = 6.77 \times 10^{-4}$ for $T = 0.05$. Additional tests were performed with other bases such as Chebyshev polynomials and Fourier series and the behavior was similar.

We show the numerical convergence of the solution with respect to the number of basis functions by examining the coefficient tensor of the committor function. Take $T = 0.05$ as an example where we use 60 orthogonal polynomial basis in each dimension, all basis coefficients form a 60^d coefficient tensor $\mathcal{Q} = \mathcal{Q}(i_1, \dots, i_d)$ for $i_1, \dots, i_d = 1, \dots, 60$. Specifically, the first dimension is most meaningful and the rest of the dimensions are equivalent due to permutation symmetry.

We visualize the numerical convergence by a slice of the committor's coefficient tensor in the first dimension (Fig. 4.3 (a)) and the second dimension (Fig. 4.3 (b)). We can observe that the basis coefficients for the first dimension decay roughly exponentially as i_1 increases. Since the committor function value should be a constant in x_2, \dots, x_d , we observe the coefficient associated with $i_2 = 1$ is large while the rest of the coefficients are nearly zero.

4.2. Ginzburg-Landau potential

The Ginzburg-Landau theory was developed to provide a mathematical description of superconductivity [17]. In this numerical example, we consider a simplified Ginzburg-Landau model, in which the Ginzburg-Landau energy is defined for a one-dimensional scalar field $u : [0, 1] \rightarrow \mathbb{R}$ as follows:

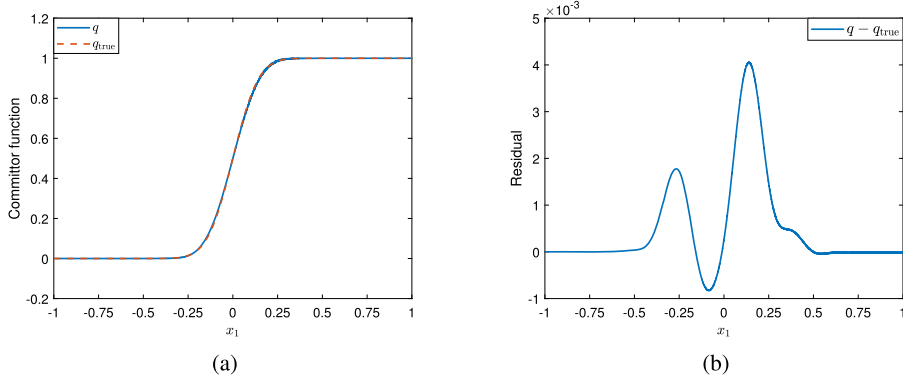


Fig. 4.2. Numerical results for the double-well potential, $T = 0.05$. (a) The numerical solution of the committor function q , compared with the ground truth q_{true} , plotted along the x_1 dimension. (b) The residual plot $q - q_{\text{true}}$ along the x_1 dimension.

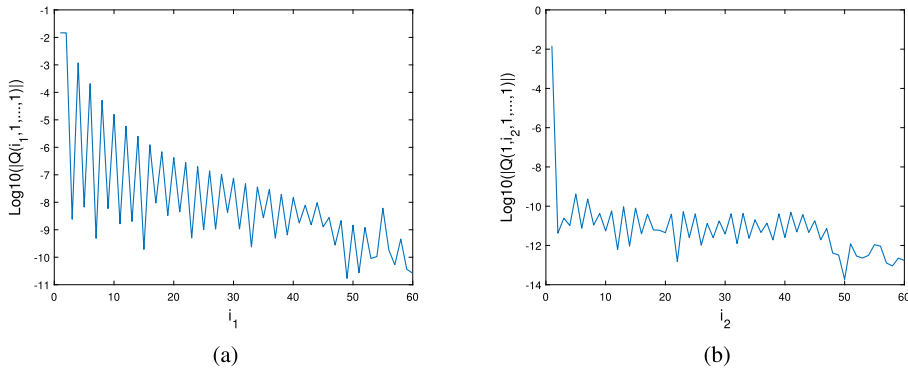


Fig. 4.3. Coefficients of basis function representation for the double-well potential model, $T = 0.05$. (a) Log magnitude of the coefficients for the x_1 dimension $\mathcal{Q}(i_1, 1, \dots, 1)$. (b) Log magnitude of the coefficients for the x_2 dimension $\mathcal{Q}(1, i_2, 1, \dots, 1)$.

$$\tilde{V}[u] = \int_0^1 \left[\frac{\lambda}{2} (u')^2 + \frac{1}{4\lambda} (1 - u^2)^2 \right] dx, \quad (4.6)$$

where λ is a small positive parameter and u satisfies the boundary conditions $u(0) = u(1) = 0$. We discretize u uniformly on $[0, 1]$ as $U = (U_1, U_2, \dots, U_d)$ with boundary conditions $U_0 = U_{d+1} = 0$. Then we approximate the continuous Ginzburg-Landau energy (4.6) with the discretization

$$V(U) := \sum_{i=1}^{d+1} \frac{\lambda}{2} \left(\frac{U_i - U_{i-1}}{h} \right)^2 + \frac{1}{4\lambda} (1 - U_i^2)^2, \quad (4.7)$$

where the grid spacing $h = 1/(d+1)$. We fix $d = 50$ and $\lambda = 0.03$. Note that $V(U)$ has two global minima U_{\pm} illustrated in Fig. 4.4. We let A and B be the balls $\{U : \|U - U_{\pm}\| \leq R\}$ centered at the global minima. The radius R is set to be 2.5, chosen such that the balls A and B roughly contain the regions of high equilibrium probability density around the two centers.

We present numerical results for two representative temperatures $T = 8$ and $T = 16$. We enforce the boundary conditions by constructing soft boundary measures p_A and p_B in MPS/TT format following (3.8). We detail the approximation of the equilibrium probability density p in MPS/TT format in Appendix B.

We define the domain to be the hypercube $\Omega = [-\gamma, \gamma]^{50}$. Based on our choices of λ and β , we take $\gamma = 2.6$ since this choice guarantees that the equilibrium density has negligible mass outside of Ω . To represent the committor function q , we use the first 5 Fourier basis functions $\{1, \cos(\pi x/\gamma), \sin(\pi x/\gamma), \cos(2\pi x/\gamma), \sin(2\pi x/\gamma)\}$ for each dimension. The ranks of the coefficient MPS/TT \mathcal{Q} are all taken to be 6. We initialize all the entries of the unknown tensor cores of \mathcal{Q} with normal $\mathcal{N}(0, 1)$ random numbers and then perform ALS, gradually increasing the penalty parameter ρ to better enforce the boundary conditions. In practice we observe that different initializations have little effect on the output of the algorithm.

For problems of this size, traditional methods are intractable, making it difficult to obtain an exact reference q_{true} for comparison. Instead, as a proxy we study a ‘thickened isosurface’ around $q = 0.5$, defined as $\Gamma_{\epsilon} = \{U : \|q(U) - 0.5\| \leq \epsilon\}$, where $\epsilon > 0$ is a small threshold parameter. If the solution q is indeed a satisfactory approximation of the true committor

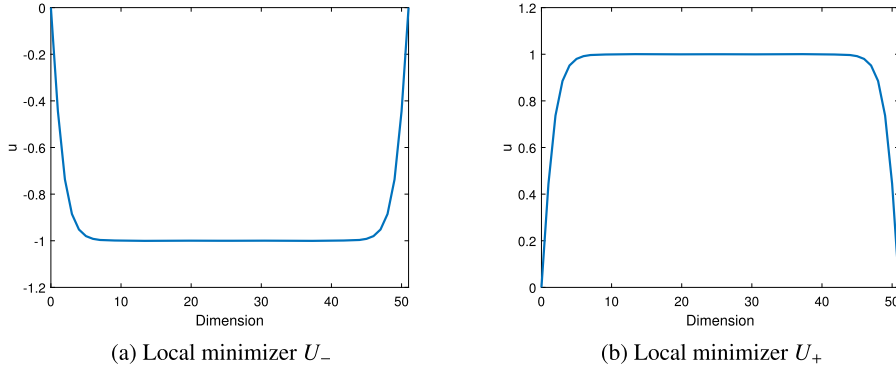


Fig. 4.4. The two global minima of the Ginzburg-Landau energy (4.7) with $d = 50$ and $\lambda = 0.03$.

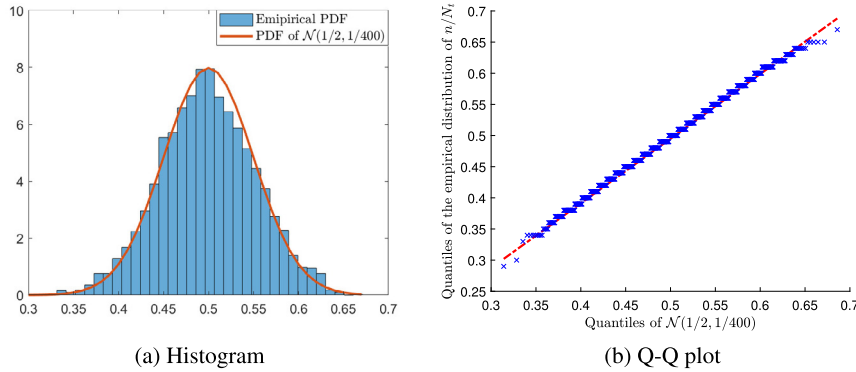


Fig. 4.5. Comparison of distributions for $T = 8$. (a) Empirical histogram for $\{n_j/N_t\}_{j=1}^{5000}$ compared with the density of $\mathcal{N}(\frac{1}{2}, 1/400)$. (b) Q-Q (quantile-quantile) plot of $\{n_j/N_t\}_{j=1}^{5000}$ compared with $\mathcal{N}(\frac{1}{2}, 1/400)$.

function, then for any trajectories given by (2.1) starting from points in Γ_ϵ , the probability of entering region B before A should be close to 0.5.

To verify this, we generate samples from the equilibrium distribution by simulating the process (2.1). Then we filter to keep samples on the isosurface Γ_ϵ using the computed committor function q and the threshold ϵ of our choice. Let us pick N_s points in Γ_ϵ , denoted $\{\tilde{U}_j\}_{j=1}^{N_s}$. For each point \tilde{U}_j , we generate N_t trajectories by simulating the Langevin process (2.1) and use n_j to denote the number of trajectories ending up in region B before A . By the central limit theorem, when N_t is large, the distribution of n_j/N_t should be well-approximated by the normal distribution $\mathcal{N}(\frac{1}{2}, (4N_t)^{-1})$. In our numerical tests, we set $\epsilon = 5 \times 10^{-3}$, $N_s = 5000$, and $N_t = 100$. The results for $T = 8$ and $T = 16$ are illustrated in Fig. 4.5 and Fig. 4.6, respectively. We compare the histogram of $\{n_j/N_t\}_{j=1}^{5000}$ with the normal distribution $\mathcal{N}(\frac{1}{2}, 1/400)$ on the left and show the Q-Q (quantile-quantile) plot of the distribution of $\{n_j/N_t\}_{j=1}^{5000}$ versus $\mathcal{N}(\frac{1}{2}, 1/400)$ on the right. These figures demonstrate that the distribution of $\{n_j/N_t\}_{j=1}^{5000}$ is indeed in good agreement with the normal distribution $\mathcal{N}(\frac{1}{2}, 1/400)$, which indicates that our solution q provides a good approximation of the true isosurface.

Next, consider the restriction of the equilibrium density to the $q = 0.5$ isosurface. Intuitively, the first term in the Ginzburg-Landau potential (4.7) encourages the configuration U to be as flat as possible. Therefore, to transition between the two boundary states U_- and U_+ , it is favorable in terms of energy to have only a single sign change in the discretized function U . We compute 10^7 samples from the equilibrium density by running the overdamped Langevin process (2.1), initialized at random states in Ω . We retain only the samples that fall in the thickened isosurface Γ_ϵ , $\epsilon = 5 \times 10^{-3}$. Then we perform 2-means clustering on these samples. In Fig. 4.7 (a) and Fig. 4.8 (a), we show the centroids $U^{(1)}$ and $U^{(2)}$ of the two clusters for $T = 8$ and $T = 16$, respectively. These configurations are symmetric with a single sign change.

Next we project all samples in the $q = 0.5$ isosurface to the line containing the two centroids, i.e., to points of the form $\theta U^{(1)} + (1 - \theta)U^{(2)}$. In Fig. 4.7 (b) and Fig. 4.8 (b) we plot the histograms of θ for all samples to demonstrate that these distributions are indeed bimodal. Observe that at higher temperature, the bimodality is less pronounced.

Finally, we study transition paths via the deterministic reactive flow [24]:

$$\frac{dU(t)}{dt} = \frac{1}{\beta} p(U(t)) \nabla q(U(t)). \quad (4.8)$$

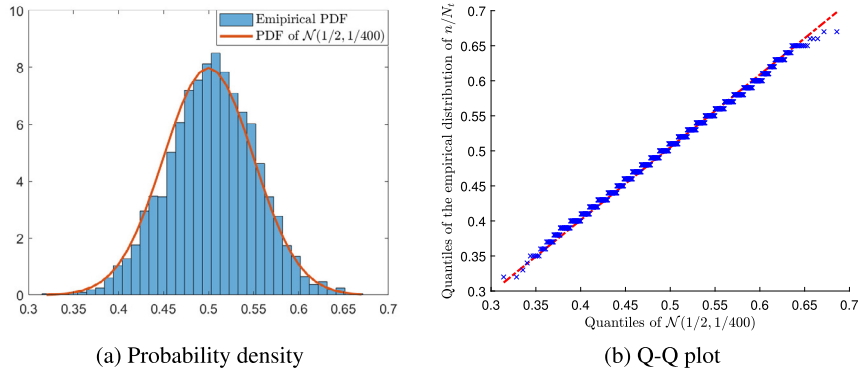


Fig. 4.6. Comparison of distributions for $T = 16$. (a) Empirical histogram for $\{n_j/N_t\}_{j=1}^{5000}$ compared with the density of $\mathcal{N}(\frac{1}{2}, 1/400)$. (b) Q-Q (quantile-quantile) plot of $\{n_j/N_t\}_{j=1}^{5000}$ compared with $\mathcal{N}(\frac{1}{2}, 1/400)$.

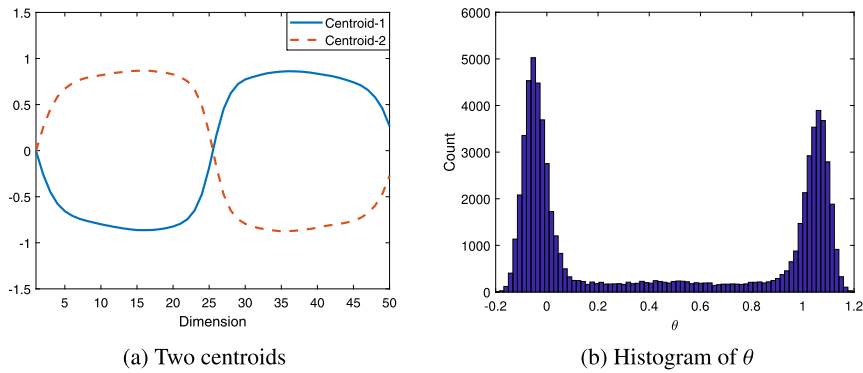


Fig. 4.7. Analysis of the $q = 0.5$ isosurface for $T = 8$. (a) Centroids of the two clusters in the $q = 0.5$ isosurface. (b) Histogram of the 1-dimensional coordinate θ of the isosurface samples along the line between the two clusters.

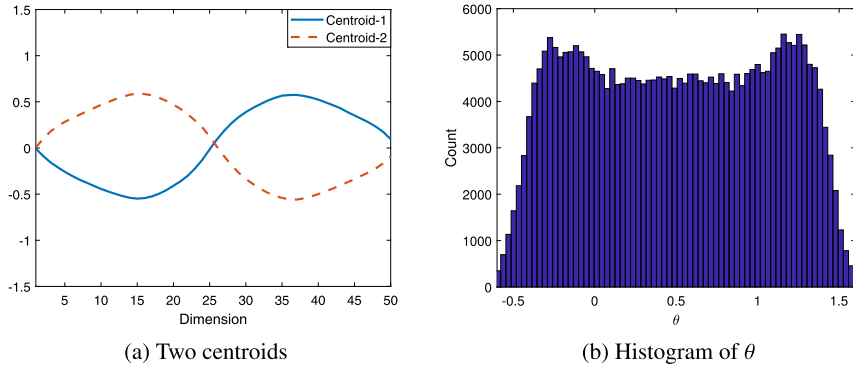


Fig. 4.8. Analysis of the $q = 0.5$ isosurface for $T = 16$. (a) Centroids of the two clusters in the $q = 0.5$ isosurface. (b) Histogram of the 1-dimensional coordinate θ of the isosurface samples along the line between the two clusters.

Based on Fig. 4.7, we expect that at low temperatures the transition paths between A and B are localized within in two reaction tubes. We visualize one of the transition paths at temperature $T = 8$ in Fig. 4.9. The leftmost curve corresponds to the initial state of (4.8), for which $q = 0.1$. Meanwhile $q = 0.9$ for the rightmost curve. The red arrow indicates the direction of time evolution.

4.3. Gaussian mixture equilibrium density

To evaluate the model performance on rugged energy landscape, we consider constructing a more complicated equilibrium distribution with several isolated local maxima using Gaussian mixture models. In this example we use a mixture of 7 Gaussian densities,

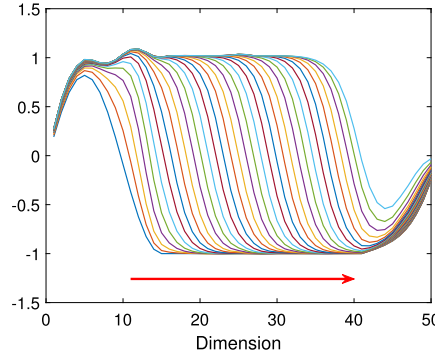


Fig. 4.9. Visualization of one transition path for $T = 8$.

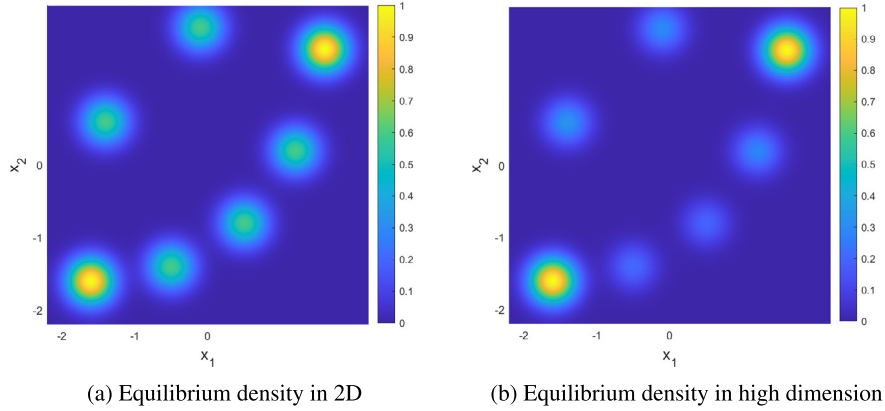


Fig. 4.10. The Gaussian mixture model equilibrium density visualized in the first two dimensions. (a) Equilibrium density in 2D. (b) Equilibrium density where the mean of Gaussians is in high dimension.

$$p(\mathbf{x}) = \exp(-\beta \|\mathbf{x} - C_A\|) + \exp(-\beta \|\mathbf{x} - C_B\|) + 0.6 \sum_{i=1}^3 \exp(-\beta \|\mathbf{x} - C_i\|) + \eta \sum_{i=4}^5 \exp(-\beta \|\mathbf{x} - C_i\|), \quad (4.9)$$

where C_A , C_B and C_i for $i = 1, \dots, 5$ are the 7 Gaussian centers. η is some parameter. In this example let the problem dimension $d = 10$ and all the centers be roughly contained in the first two dimensions for visualization purpose. Specifically for the first two dimensions, $C_A = (-1.6, -1.6)$, $C_B = (1.6, 1.6)$, $C_1 = (-0.5, -1.4)$, $C_2 = (0.5, -0.8)$, $C_3 = (1.2, 0.2)$, $C_4 = (-1.4, 0.6)$, $C_5 = (-0.1, 1.9)$. If we pad all 8 other dimensions with 0, the centers strictly lie on a 2-dimensional subspace. In Fig. 4.10 (a) we show the density in 2D. To make the problem more difficult, we perturb the mean of these Gaussians with small independent Gaussian noise $0.1N(0, 1)$. The magnitude of the perturbation is chosen such that the Gaussian centers can still be visualized in the first two dimensions. In Fig. 4.10 (b) we show the “perturbed” density, which is used in the following numerical tests.

Note the five Gaussian centers in the middle are roughly aligned on two curves: a lower curve containing C_1, C_2, C_3 and an upper curve containing C_4, C_5 . Both curves share C_A and C_B as their end points. Similarly we study the transition paths between the two boundaries by simulating the reactive flow (4.8). Specifically we study how the transition path changes with the additional local maxima of the equilibrium density. To this end, we pick two η values 0.6 and 1.6 to alter the magnitude the top two Gaussian densities.

Let the boundaries A and B be the balls centered at two centers $\{\mathbf{x} : \|\mathbf{x} - C_A\| \leq R\}$ and $\{\mathbf{x} : \|\mathbf{x} - C_B\| \leq R\}$, respectively. In this example $R = 0.22$, chosen such that balls A and B roughly contained the regions of high equilibrium density around the two centers. The temperature is set to be $T = 0.1$. We define the solution domain to be the hypercube $\Omega = [-2.4, 2.4]^{10}$, which guarantees the equilibrium density has negligible mass outside the solution domain. We use the first 60 Fourier basis to parameterize the committor function. The ranks of the coefficient MPS/TT \mathcal{Q} are all taken to be 4.

In Fig. 4.11, we show the transition path between boundaries A, B for $\eta = 0.6, 1.6$. Both the transition paths share the same starting point $[-1.2, -1.6]$. We can observe that the transition path is shifting towards the top as the magnitude η increases.

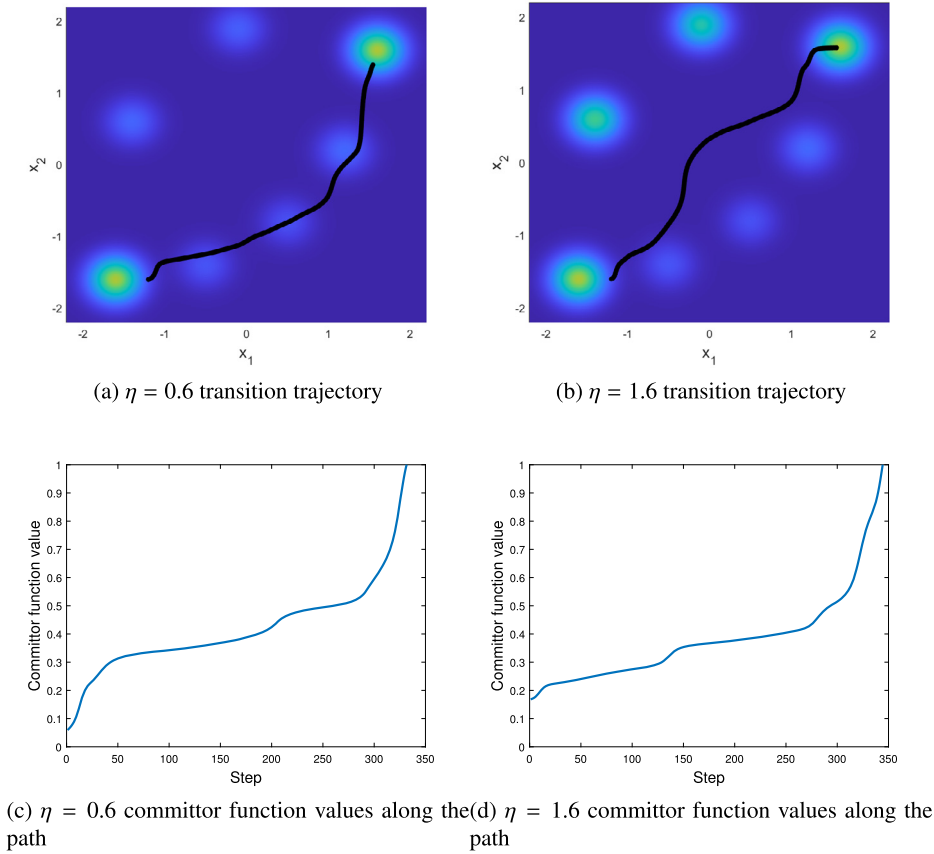


Fig. 4.11. Transition paths (black curves) overlaid on the Gaussian mixture model equilibrium density visualized in the first two dimensions for (a) $\eta = 0.6$ and (b) $\eta = 1.6$. Committor function values along the transition path for (c) $\eta = 0.6$ and (d) $\eta = 1.6$.

5. Conclusion

In this paper, we propose a novel approach for computing high-dimensional committor functions using MPS/TT. In particular, we start from the variational formulation (2.7) for the soft committor function, which can be viewed as an approximation of the committor function but which also enjoys a probabilistic interpretation in its own right. To compute high-dimensional integrals, we approximate the equilibrium density and soft boundary measures in MPS/TT format. Meanwhile, the unknown committor function q is also parametrized in MPS/TT format. The variational problem can then be reformulated using standard MPO and MPS/TT operations, and the optimization of q can be performed with $\mathcal{O}(d)$ complexity. Extensive numerical experiments demonstrate the computational efficiency and accuracy of the proposed method.

Notice that our proposed method relies on the assumption of certain structures of the variables in order to approximate the equilibrium density efficiently in MPS/TT format. One expects the MPS/TT-based approximation to be successful in a very high-dimensional limit when there is a 1D or quasi-1D graphical model structure underlying the equilibrium measure. Already we believe that the capacity to treat such systems with high accuracy is a meaningful contribution, since many previous works in the literature have considered problems with such structure, such as the double-well and rugged-Muller potentials (embedded in high dimensions) and the Ginzburg-Landau potential [19,24,26,42]. Even in the case without obvious 1D ordering, MPS/TT have demonstrated excellent empirical performance in high-dimensional PDE and chemistry [40,2,43,3].

However in principle our method can also be extended to more complicated systems and more general network with significant interactions between any particles, provided that suitable tensor network contractions can be performed. It has become an active research field of extending the tensor network algorithm to more general systems, see for example [18]. Our complexity analysis relies on the assumption that the rank of q remains constant as the number of dimension increases, which may not hold for complicated networks. As a future work, one can monitor the rank of the committor function q via a two-site alternating minimization scheme. If we see any signs of rapidly growing rank, it may indicate the method is more computationally expensive than expected and MPS/TT is not the most efficient format to represent the true committor function. As a future work, one can monitor the rank of the committor function q via a two-site alternating minimization scheme [10].

CRediT authorship contribution statement

Yian Chen: Conceptualization, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Jeremy Hoskins:** Conceptualization, Formal analysis, Methodology, Resources, Software, Writing – review & editing. **Yuehaw Khoo:** Conceptualization, Formal analysis, Methodology, Resources, Software, Supervision, Validation, Writing – review & editing. **Michael Lindsey:** Conceptualization, Formal analysis, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The work of Y.C. and Y.K. is supported in part by the National Science Foundation under Award DMS-2111563. The work of M.L. is supported by the National Science Foundation under Award No.1903031.

Appendix A. Soft committor functions

Consider the optimization problem (2.7). We offer a probabilistic interpretation of the optimizer q , which we call a ‘soft committor function.’ This probabilistic interpretation will justify its use qualitatively and quantitatively as a proxy for the committor function.

Define

$$f := \frac{\rho \cdot p_A}{\beta \cdot p}, \quad g := \frac{\rho \cdot p_B}{\beta \cdot p},$$

and note that the Euler-Lagrange equation associated to (2.7) then reads as

$$-\beta^{-1} \Delta q(\mathbf{x}) + \nabla V(\mathbf{x}) \cdot \nabla q(\mathbf{x}) + (f(\mathbf{x}) + g(\mathbf{x})) q(\mathbf{x}) = g(\mathbf{x}). \quad (\text{A.1})$$

We shall now rederive this PDE (A.1) via a probabilistic construction. In particular this construction will imply that the solution q satisfies $0 \leq q(\mathbf{x}) \leq 1$ for all $\mathbf{x} \in \Omega$.

Consider a stochastic process \mathbf{X}_t that modifies the standard overdamped Langevin diffusion

$$d\mathbf{X}_t = -\nabla V(\mathbf{X}_t) dt + \sqrt{2\beta^{-1}} d\mathbf{W}_t,$$

where \mathbf{W}_t is a Wiener process, by adding jumps to one of two possible ‘cemetery states’ c_A, c_B with state-dependent rates specified by f, g , respectively.

In other words, we view $\mathbf{X}_t \in \Omega \cup \{c_A\} \cup \{c_B\}$ as the continuous-time limit of the discrete-time Markov chain (also denoted \mathbf{X}_t , abusing notation slightly) defined by the update

$$\mathbf{X}_{t+\Delta t} = \begin{cases} c_A, & \text{if } U_t^A < f(\mathbf{X}_t) \Delta t, \\ c_B, & \text{otherwise if } U_t^B < g(\mathbf{X}_t) \Delta t \\ \mathbf{X}_t - \nabla V(\mathbf{X}_t) \Delta t + \sqrt{2\beta^{-1} \Delta t} \mathbf{Z}_t & \text{otherwise,} \end{cases}$$

whenever $\mathbf{X}_t \in \Omega$. Here the U_t^A, U_t^B are i.i.d. uniformly distributed random variables on $[0, 1]$, and the \mathbf{Z}_t are i.i.d. standard Gaussian random variables. In other words, at each time step the stochastic process is sent to the cemetery state c_A with probability $f(\mathbf{X}_t) \Delta t$ and the cemetery state c_B with probability $g(\mathbf{X}_t) \Delta t$, else it is advanced by the usual overdamped Langevin dynamics. Note that in the limit of Δt small, it is unlikely for both $U_t^A < f(\mathbf{X}_t) \Delta t$ and $U_t^B < g(\mathbf{X}_t) \Delta t$ to hold. Specifically, the probability of this is only $O(\Delta t^2)$ and does not affect the continuous-time limit.

Moreover, if $\mathbf{X}_t = c_A$ (resp., c_B), then we define $\mathbf{X}_{t+\Delta t} = c_A$ (resp., c_B) deterministically. Then define the stopping time τ as

$$\tau = \inf\{t : \mathbf{X}_t \in \{c_A, c_B\}\},$$

and define $q : \Omega \rightarrow [0, 1]$ by

$$q(\mathbf{x}) = \mathbb{P}(\mathbf{X}_\tau = c_B \mid \mathbf{X}_0 = \mathbf{x}).$$

We claim that q so defined (in the continuous-time limit $\Delta t \rightarrow 0$) satisfies the PDE (A.1). Note that this construction of q coincides with the usual probabilistic construction for the committor function, modulo a change in the underlying stochastic process. We justify the claim only formally. A rigorous argument can be made by analogy to arguments made for the ordinary committor function [5].

A.1. Formal PDE derivation

The PDE is derived by conditioning on $\mathbf{X}_0 = \mathbf{x}$, and writing

$$\begin{aligned} q(\mathbf{x}) &= \mathbb{P}(\mathbf{X}_\tau = c_B) \\ &= \mathbb{P}(\mathbf{X}_{\Delta t} = c_B) + \mathbb{P}(\mathbf{X}_{\Delta t} \notin \{c_A, c_B\}) \mathbb{E} \left[\mathbb{P}(\mathbf{X}'_\tau = c_B \mid \mathbf{X}'_0 = \mathbf{X}_{\Delta t}) \right] + O(\Delta t^2), \end{aligned}$$

where \mathbf{X}'_t is an independent dummy stochastic process with the same law as \mathbf{X}_t . But then

$$q(\mathbf{x}) = \mathbb{P}(\mathbf{X}_{\Delta t} = c_B) + \mathbb{P}(\mathbf{X}_{\Delta t} \notin \{c_A, c_B\}) \mathbb{E} [q(\mathbf{X}_{\Delta t})] + O(\Delta t^2).$$

Expanding further we obtain

$$q(\mathbf{x}) = g(\mathbf{x}) \Delta t + (1 - f(\mathbf{x}) \Delta t - g(\mathbf{x}) \Delta t) \mathbb{E} \left[q \left(\mathbf{x} - \nabla V(\mathbf{x}) \Delta t + \sqrt{2\beta^{-1} \Delta t} \mathbf{Z}_0 \right) \right] + O(\Delta t^2). \quad (\text{A.2})$$

Then we can expand q via Taylor expansion:

$$\begin{aligned} q \left(\mathbf{x} - \nabla V(\mathbf{x}) \Delta t + \sqrt{2\beta^{-1} \Delta t} \mathbf{Z}_0 \right) \\ = q(\mathbf{x}) - \nabla V(\mathbf{x}) \cdot \nabla q(\mathbf{x}) \Delta t + \sqrt{2\beta^{-1} \Delta t} \mathbf{Z}_0 \cdot \nabla V(\mathbf{x}) + \beta^{-1} \Delta t \mathbf{Z}_0^\top \nabla^2 q(\mathbf{x}) \mathbf{Z}_0 + o(\Delta t), \end{aligned}$$

from which we obtain

$$\mathbb{E} \left[q \left(\mathbf{x} - \nabla V(\mathbf{x}) \Delta t + \sqrt{2\beta^{-1} \Delta t} \mathbf{Z}_0 \right) \right] = q(\mathbf{x}) + (-\nabla V(\mathbf{x}) \cdot \nabla q(\mathbf{x}) + \beta^{-1} \Delta q(\mathbf{x})) \Delta t + o(\Delta t).$$

It follows from plugging into (A.2) that

$$q(\mathbf{x}) = q(\mathbf{x}) + [g(\mathbf{x}) - \nabla V(\mathbf{x}) \cdot \nabla q(\mathbf{x}) + \beta^{-1} \Delta q(\mathbf{x}) - (f(\mathbf{x}) + g(\mathbf{x}))q(\mathbf{x})] \Delta t + o(\Delta t).$$

Canceling $q(\mathbf{x})$ from both sides, dividing by Δt , and taking the limit as $\Delta t \rightarrow 0$, we obtain precisely (A.1), as desired.

Appendix B. Discretization of the Ginzburg-Landau density

In this section, we show that one can approximate the equilibrium distribution of the Ginzburg-Landau potential as (3.7) via the eigenfunctions of a certain kernel.

The computation of the committor function for the Ginzburg-Landau potential requires the numerical approximation of the operator $\mathcal{H} : L^2([-R, R])^d \rightarrow \mathbb{R}$ defined by

$$\begin{aligned} \mathcal{H}[\phi_1, \dots, \phi_d] &= c_\lambda \int_{-R}^R \dots \int_{-R}^R K(0, x_1) K(x_1, x_2) K(x_2, x_3) \dots K(x_{d-1}, x_d) \\ &\quad \times K(x_d, 0) \phi_1(x_1) \dots \phi_d(x_d) dx_1 \dots dx_d, \end{aligned}$$

where

$$K(x, y) = e^{-\frac{1}{8\lambda}(1-x^2)^2} e^{\frac{\lambda}{2h^2}(x-y)^2} e^{-\frac{1}{8\lambda}(1-y^2)^2},$$

$c_\lambda = e^{-\frac{1}{4\lambda}}$, and R is some fixed positive constant.

Considering the operator $\mathcal{H}_0 : L^2([-R, R]) \rightarrow L^2([-R, R])$ defined by

$$\mathcal{H}_0[\phi](x) = \int_{\Omega} K(x, y) \phi(y) dy,$$

we observe that it is compact, symmetric, and positive semi-definite. In particular, it has an eigendecomposition consisting of a countable basis of orthonormal eigenfunctions $u_1, u_2, \dots \in L^2([-R, R])$, together with corresponding non-negative eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq \dots \geq 0$, such that

$$\int_{-R}^R K(x, y) \phi(y) dy = \sum_{j=1}^{\infty} \lambda_j u_j(x) \int_{-R}^R u_j(y) \phi(y) dy$$

for all $\phi \in L^2([-R, R])$. Moreover, for the kernel K defined above it is easily shown that $\lambda_j \in o(e^{-\alpha j})$ for any $\alpha > 0$ as $j \rightarrow \infty$ (see [29] for example). We note that the implicit constant in the previous estimate will depend on α but not on j , and increases rapidly as $\alpha \rightarrow \infty$. For convenience, let us define the re-scaled eigenfunctions $v_i = \sqrt{\lambda_i} u_i$. Upon substitution of the eigendecomposition of \mathcal{H}_0 into the definition of \mathcal{H} , we obtain

$$\mathcal{H}[\phi_1, \dots, \phi_d] = c_\lambda \sum_{j_0, \dots, j_d=1}^{\infty} v_{j_0}(0) A_{j_0, j_1}[\phi_1] \cdots A_{j_{d-1}, j_d}[\phi_d] v_{j_d}(0), \quad (\text{B.1})$$

where $A_{j, \ell}[\phi] = \int_{\Omega} v_j(x) v_\ell(x) \phi(x) dx$. We observe that

$$|A_{j, \ell}[\phi]| \leq \sqrt{\lambda_j \lambda_\ell} \|\phi\|_{L^\infty},$$

and thus, if the sums in (B.1) are truncated at term J , then

$$\begin{aligned} & \left| \mathcal{H}[\phi_1, \dots, \phi_d] - \mathcal{H}^{(J)}[\phi_1, \dots, \phi_d] \right| \\ & \leq 2 c_\lambda \|\phi_1\|_{L^\infty} \cdots \|\phi_d\|_{L^\infty} \sqrt{K(0, 0) K^{(J)}(0, 0) \text{Tr}(\mathcal{H}_0)^{d-1/2}} \sqrt{\sum_{j=J+1}^{\infty} \lambda_j} \\ & \quad + (d-1) c_\lambda \|\phi_1\|_{L^\infty} \cdots \|\phi_d\|_{L^\infty} K(0, 0) \text{Tr}(\mathcal{H}_0)^{d-1} \sum_{j=J+1}^{\infty} \lambda_j, \end{aligned}$$

where $K^{(J)}(0, 0) := \sum_{j=J+1}^{\infty} v_j(0)^2 \leq K(0, 0)$ and $\mathcal{H}^{(J)}$ denotes the truncation of \mathcal{H} . The analyticity of K guarantees that $\sum_{j=J+1}^{\infty} \lambda_j$ go to zero exponentially quickly [29]) and hence for any $\alpha > 0$ there exists a constant $M_{\alpha, \lambda, d, R}$ depending on α , λ , d , and R such that

$$\left| \mathcal{H}[\phi_1, \dots, \phi_d] - \mathcal{H}^{(J)}[\phi_1, \dots, \phi_d] \right| \leq M_{\alpha, \lambda, d, R} e^{-\alpha J/2} \|\phi_1\|_{L^\infty} \cdots \|\phi_d\|_{L^\infty},$$

for all $J \geq 1$.

Next, we express the input functions ϕ_i in a basis of (suitably-scaled) Chebyshev polynomials,

$$\phi_i(x) = \sum_{n=0}^{\infty} \phi_{i,n} T_n\left(\frac{x}{R}\right),$$

where T_n is the n th standard Chebyshev polynomial. Let $\phi_i^{(N)}$ denote the N th order truncation of ϕ_i defined by

$$\phi_i^{(N)}(x) := \sum_{n=0}^N \phi_{i,n} T_n\left(\frac{x}{R}\right).$$

Then, substituting these Chebyshev expansions into our expression for \mathcal{H} , we find

$$\mathcal{H}[\phi_1, \dots, \phi_d] = c_\lambda \sum_{j_0, \dots, j_d=1}^{\infty} \sum_{n_1, \dots, n_d=1}^{\infty} A_{j_0, j_1}^{n_1} \cdots A_{j_{d-1}, j_d}^{n_d} v_{j_0}(0) v_{j_d}(0) \phi_{1, n_1} \cdots \phi_{d, n_d}, \quad (\text{B.2})$$

where

$$A_{j, \ell}^n = A_{j, \ell}[T_n(x/R)].$$

In the following, we assume that ϕ_1, \dots, ϕ_d are in $C^{p+1}([-R, R])$ for some fixed integer $p \geq 1$ and set

$$V_p = \max_i \left\| \frac{d^{p+1}}{dx^{p+1}} \phi_i(x/R) \right\|_{L^1([-1, 1])}.$$

A standard estimate from approximation theory [46] gives the following bound on the rate of decay of the coefficients of ϕ_1, \dots, ϕ_d :

$$|\phi_{i,n}| \leq \frac{2V_p}{\pi n(n-1) \cdots (n-p)}.$$

In particular,

$$\sum_{n=N+1}^{\infty} |\phi_{i,n}| \leq \frac{2V_p}{\pi p(N-p)^p}.$$

If the sums over the Chebyshev coefficients (the n indices) in (B.2) are truncated at a fixed integer N and the sums over the eigenvalues (the j indices) are truncated at J , then the error is bounded by

$$\begin{aligned} & \left| \mathcal{H}[\phi_1, \dots, \phi_d] - \mathcal{H}^{(J,N)}[\phi_1, \dots, \phi_d] \right| \\ & \leq M_{\lambda,d,R} e^{-\alpha J/2} \|\phi_1\|_{L^\infty} \cdots \|\phi_d\|_{L^\infty} + \frac{2dc_\lambda V_p}{\pi p(N-p)^p} \left(1 + \frac{2V_p}{\pi p(N-p)^p} \right)^{d-1} K(0,0) \text{Tr}(\mathcal{H}_0)^d \max_i \|\phi_i\|_{L^\infty}^{d-1}. \end{aligned}$$

Here $\mathcal{H}^{(J,N)}$ denotes the operator obtained by truncating the sums over eigenvalues in \mathcal{H} at J and projecting onto the first $N+1$ terms in the Chebyshev expansions of ϕ_1, \dots, ϕ_d . This latter projection, along with the eigendecomposition of \mathcal{H}_0 , can be performed easily on the computer using standard numerical integration. This yields a discrete, finite-dimensional tensor which is the object we use in our approach when approximating \mathcal{H} .

References

- [1] I. Affleck, T. Kennedy, E.H. Lieb, H. Tasaki, Valence bond ground states in isotropic quantum antiferromagnets, in: *Condensed Matter Physics and Exactly Soluble Models*, Springer, 1988, pp. 253–304.
- [2] M. Bachmayr, R. Schneider, A. Uschmajew, Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations, *Found. Comput. Math.* 16 (2016) 1423–1472.
- [3] A. Baiardi, M. Reiher, The density matrix renormalization group in chemistry and molecular physics: recent developments and new challenges, *J. Chem. Phys.* 152 (2020) 040903.
- [4] A. Berteotti, A. Cavalli, D. Branduardi, F.L. Gervasio, M. Recanatini, M. Parrinello, Protein conformational transitions: the closure mechanism of a kinase explored by atomistic simulations, *J. Am. Chem. Soc.* 131 (2009) 244–250.
- [5] R.N. Bhattacharya, E.C. Waymire, *Stochastic Processes with Applications*, SIAM, 2009.
- [6] A. Blum, J. Hopcroft, R. Kannan, *Foundations of data science*, in: *Vorabversion eines Lehrbuchs*, 2016, p. 5.
- [7] R.R. Coifman, I.G. Kevrekidis, S. Lafon, M. Maggioni, B. Nadler, Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems, *Multiscale Model. Simul.* 7 (2008) 842–864.
- [8] D.E. Deutsch, Quantum computational networks, *Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci.* 425 (1989) 73–90.
- [9] G. Evenbly, G. Vidal, Tensor network states and geometry, *J. Stat. Phys.* 145 (2011) 891–918.
- [10] M. Fishman, S. White, E. Stoudenmire, The ITensor software library for tensor network calculations, *SciPost Physics Codebases* (2022) 004.
- [11] B.J. Grant, A.A. Gorfe, J.A. McCammon, Large conformational changes in proteins: signaling and other functions, *Curr. Opin. Struct. Biol.* 20 (2010) 142–147.
- [12] L. Grasedyck, Hierarchical low rank approximation of tensors and multivariate functions, in: *Lecture Notes of the Zürich Summer School on Sparse Tensor Discretizations of High-Dimensional Problems*, 2010.
- [13] L. Grasedyck, D. Kressner, C. Tobler, A literature survey of low-rank tensor approximation techniques, *GAMM-Mitt.* 36 (2013) 53–78.
- [14] W. Hackbusch, B.N. Khoromskij, E.E. Tyrtshnikov, *Hierarchical Kronecker tensor-product approximations*, Walter de Gruyter, Berlin, Germany, 2005.
- [15] J. Han, A. Jentzen, E. Weinan, Solving high-dimensional partial differential equations using deep learning, *Proc. Natl. Acad. Sci.* 115 (2018) 8505–8510.
- [16] F.L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, *J. Math. Phys.* 6 (1927) 164–189.
- [17] K.-H. Hoffmann, Q. Tang, *Ginzburg-Landau Phase Transition Theory and Superconductivity*, vol. 134, Birkhäuser, 2012.
- [18] K. Hyatt, E.M. Stoudenmire, DMRG approach to optimizing two-dimensional tensor networks, *arXiv preprint*, arXiv:1908.08833, 2019.
- [19] Y. Khoo, J. Lu, L. Ying, Solving for high-dimensional committor functions using artificial neural networks, *Res. Math. Sci.* 6 (2019) 1–13.
- [20] Y. Khoo, J. Lu, L. Ying, Solving parametric PDE problems with artificial neural networks, *Eur. J. Appl. Math.* 32 (2021) 421–435.
- [21] B.N. Khoromskij, Tensor numerical methods for multidimensional PDEs: theoretical analysis and initial applications, *ESAIM Proc. Surv.* 48 (2015) 1–28.
- [22] B.N. Khoromskij, C. Schwab, Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs, *SIAM J. Sci. Comput.* 33 (2011) 364–385.
- [23] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (2009) 455–500.
- [24] R. Lai, J. Lu, Point cloud discretization of Fokker–Planck operators for committor functions, *Multiscale Model. Simul.* 16 (2018) 710–726.
- [25] A.C. Lasaga, 2. Transition state theory, in: *Kinetic Theory in the Earth Sciences*, Princeton University Press, 2014, pp. 152–219.
- [26] H. Li, Y. Khoo, Y. Ren, L. Ying, A semigroup method for high dimensional committor functions based on neural network, *arXiv preprint*, arXiv:2012.06727, 2020.
- [27] H. Li, Y. Khoo, Y. Ren, L. Ying, Solving for high dimensional committor functions using neural network with online approximation to derivatives, *arXiv preprint*, arXiv:2012.06727, 2020.
- [28] Q. Li, B. Lin, W. Ren, Computing committor functions for the study of rare events using deep learning, *J. Chem. Phys.* 151 (2019) 054112.
- [29] G. Little, J. Reade, Eigenvalues of analytic kernels, *SIAM J. Math. Anal.* 15 (1984) 133–136.
- [30] J. Lu, J. Nolen, Reactive trajectories and the transition path process, *Probab. Theory Relat. Fields* 161 (2015) 195–244.
- [31] J. Lu, E. Vanden-Eijnden, Exact dynamical coarse-graining without time-scale separation, *J. Chem. Phys.* 141 (2014), 07B619_1.
- [32] L. Maragliano, A. Fischer, E. Vanden-Eijnden, G. Ciccotti, String method in collective variables: minimum free energy paths and isocommittor surfaces, *J. Chem. Phys.* 125 (2006) 024106.
- [33] N. Okuyama-Yoshida, M. Nagaoka, T. Yamabe, Transition-state optimization on free energy surface: toward solution chemical reaction ergodography, *Int. J. Quant. Chem.* 70 (1998) 95–103.
- [34] R. Orús, Advances on tensor network theory: symmetries, fermions, entanglement, and holography, *Eur. Phys. J. B* 87 (2014) 1–18.
- [35] R. Orús, A practical introduction to tensor networks: matrix product states and projected entangled pair states, *Ann. Phys.* 349 (2014) 117–158.
- [36] I.V. Oseledets, Tensor-train decomposition, *SIAM J. Sci. Comput.* 33 (2011) 2295–2317.
- [37] D. Perez-Garcia, F. Verstraete, M.M. Wolf, J.I. Cirac, Matrix product state representations, *arXiv preprint*, arXiv:quant-ph/0608197, 2006.
- [38] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.

- [39] W. Ren, E. Vanden-Eijnden, P. Maragakis, W. E. Transition pathways in complex systems: application of the finite-temperature string method to the alanine dipeptide, *J. Chem. Phys.* 123 (2005) 134109.
- [40] L. Richter, L. Sallandt, N. Nüsken, Solving high-dimensional parabolic PDEs using the tensor train format, in: *International Conference on Machine Learning*, PMLR, 2021, pp. 8998–9009.
- [41] G.M. Rotskoff, A.R. Mitchell, E. Vanden-Eijnden, Active importance sampling for variational objectives dominated by rare events: consequences for optimization and generalization, *arXiv preprint*, arXiv:2008.06334, 2020.
- [42] G.M. Rotskoff, E. Vanden-Eijnden, Learning with rare data: using active importance sampling to optimize objectives dominated by rare events, *arXiv preprint*, arXiv:2008.06334, 2020.
- [43] D.V. Savostyanov, S. Dolgov, J. Werner, I. Kuprov, Exact NMR simulation of protein-size spin systems using tensor train formalism, *Phys. Rev. B* 90 (2014) 085139.
- [44] J. Sirignano, K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations, *J. Comput. Phys.* 375 (2018) 1339–1364.
- [45] E.H. Thiede, D. Giannakis, A.R. Dinner, J. Weare, Galerkin approximation of dynamical quantities using trajectory data, *J. Chem. Phys.* 150 (2019) 244111.
- [46] L.N. Trefethen, *Approximation Theory and Approximation Practice*, SIAM, 2013.
- [47] L.R. Tucker, Some mathematical notes on three-mode factor analysis, *Psychometrika* 31 (1966) 279–311.
- [48] E. Vanden-Eijnden, et al., Transition-path theory and path-finding algorithms for the study of rare events, *Annu. Rev. Phys. Chem.* 61 (2010) 391–420.
- [49] E. Vanden-Eijnden, M. Venturoli, Revisiting the finite temperature string method for the calculation of reaction tubes and free energies, *J. Chem. Phys.* 130 (2009), 05B605.
- [50] R. Vershynin, *High-dimensional Probability: An Introduction with Applications in Data Science*, vol. 47, Cambridge University Press, 2018.
- [51] F. Verstraete, V. Murg, J.I. Cirac, Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems, *Adv. Phys.* 57 (2008) 143–224.
- [52] E. Weinan, W. Ren, E. Vanden-Eijnden, Transition pathways in complex systems: reaction coordinates, isocommittor surfaces, and transition tubes, *Chem. Phys. Lett.* 413 (2005) 242–247.
- [53] E. Weinan, W. Ren, E. Vanden-Eijnden, et al., Finite temperature string method for the study of rare events, *J. Phys. Chem. B* 109 (2005) 6688–6693.
- [54] E. Weinan, E. Vanden-Eijnden, Towards a theory of transition paths, *J. Stat. Phys.* 123 (2006) 503–523.
- [55] E. Weinan, B. Yu, The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems, *Commun. Math. Stat.* 6 (2018) 1–12.
- [56] S.R. White, Density matrix formulation for quantum renormalization groups, *Phys. Rev. Lett.* 69 (1992) 2863.
- [57] D. Zahn, S. Leoni, Nucleation and growth in pressure-induced phase transitions from molecular dynamics simulations: mechanism of the reconstructive transformation of NaCl to the CsCl-type structure, *Phys. Rev. Lett.* 92 (2004) 250201.
- [58] B. Zhao, L. Li, F. Lu, Q. Zhai, B. Yang, C. Schick, Y. Gao, Phase transitions and nucleation mechanisms in metals studied by nanocalorimetry: a review, *Thermochim. Acta* 603 (2015) 2–23.