The Journal of Chemical Physics

RESEARCH ARTICLE | FEBRUARY 23 2024

Accurate estimates of dynamical statistics using memory

Chatipat Lorpaiboon ; Spencer C. Guo ; John Strahan ; Jonathan Weare ; Aaron R. Dinner

J. Chem. Phys. 160, 084108 (2024)
https://doi.org/10.1063/5.0187145

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

Accurate estimates of dynamical statistics using memory

Cite as: J. Chem. Phys. 160, 084108 (2024); doi: 10.1063/5.0187145
Submitted: 11 November 2023 • Accepted: 29 January 2024 •
Published Online: 23 February 2024

Chatipat Lorpaiboon,[1] Spencer C. Guo,[1] John Strahan,[1] Jonathan Weare,[2] and Aaron R. Dinner[1,a)]

AFFILIATIONS

[1] Department of Chemistry and James Franck Institute, University of Chicago, Chicago, Illinois 60637, USA
[2] Courant Institute of Mathematical Sciences, New York University, New York, New York 10012, USA

[a)] Author to whom correspondence should be addressed: dinner@uchicago.edu

ABSTRACT

Many chemical reactions and molecular processes occur on time scales that are significantly longer than those accessible by direct simulations. One successful approach to estimating dynamical statistics for such processes is to use many short time series of observations of the system to construct a Markov state model, which approximates the dynamics of the system as memoryless transitions between a set of discrete states. The dynamical Galerkin approximation (DGA) is a closely related framework for estimating dynamical statistics, such as committors and mean first passage times, by approximating solutions to their equations with a projection onto a basis. Because the projected dynamics are generally not memoryless, the Markov approximation can result in significant systematic errors. Inspired by quasi-Markov state models, which employ the generalized master equation to encode memory resulting from the projection, we reformulate DGA to account for memory and analyze its performance on two systems: a two-dimensional triple well and the AIB_9 peptide. We demonstrate that our method is robust to the choice of basis and can decrease the time series length required to obtain accurate kinetics by an order of magnitude.

I. INTRODUCTION

Models that seek to treat complex systems with high fidelity typically have many variables (e.g., the positions and velocities of all the atoms in a molecular system). However, often the dynamics of interest of such models can be largely captured by a relatively small number of variables that are functions of the original ones [collective variables (CVs)]. Projecting the dynamics on to these CVs can facilitate analysis, by both improving the convergence of statistics and simplifying interpretation. However, the coupling of the CVs to orthogonal ones generally introduces a form of memory in which the projected dynamics depend not only on the states of the CVs but also on their histories.

When the dynamics have a separation of time scales, it may be possible to find CVs that minimize the memory so that it can be neglected. This is the idea behind Markov State Models (MSMs), in which the dynamics are treated as hops between discrete states, and a large number of states are often used in the hope of minimizing the time for equilibration within states. However, not all dynamics have a clear separation of time scales, and, even if they do, finding CVs that minimize the memory can be challenging.[1]

Thus, various strategies have been introduced to account for memory. One is to consider explicit histories of the CVs, as in delay embedding.[2–7] However, this increases the effective number of variables, mitigating the advantages of projecting described above. An alternative is to account for the memory through the Mori–Zwanzig formalism.[8,9] In this approach, the projected dynamics are represented through a generalized Langevin/master equation (GLE/GME) that involves a time integral over the CV histories and a kernel. This approach is physically well-motivated and can, in principle, yield highly accurate dynamics. In practice, it can be hard to determine the memory kernel.

Traditionally, researchers chose the memory kernel based on physical intuition,[10,11] and many studies still assume a specific functional form for the kernel (most often exponential) when estimating it from data. A few studies allow for more flexibility by introducing an expansion in basis functions[12–15] or alternative fitting forms[16,17] for the kernel or its Laplace/Fourier transforms,[18–20] and these show

that it does not have a simple universal functional form. However, solving for the coefficients of the expansion can be numerically ill-posed,[21] and data-driven parameterization of GLEs/GMEs remains an active area of research.[22–26]

Recently, approaching the problem from the perspective of improving the accuracy of MSMs with relatively small numbers of states,[27] Cao and co-workers obtained the memory kernel at successive discrete time points by directly inverting a GME for the state-to-state transition matrix.[27–29] This enabled them to significantly shorten the lag times needed to construct the transition matrix, which reduced the amount of simulation needed to converge statistics and increased the time resolution of the model; the small number of states facilitates interpretation.

Like many MSM studies, the studies described immediately above focus on computing implied time scales (via time correlation functions rather than directly from the eigenvalues of the transition matrix). However, MSMs can be used to solve for statistics that provide more direct insights into the mechanisms of specified events, such as the committor,[30] and we recently introduced a closely related framework that solves operator equations for such statistics, which we term dynamical Galerkin approximation (DGA).[6,7] The goal of this paper is to show how memory can improve the accuracy of such statistics.

To this end, we incorporate memory into DGA by interpreting it as an iterative algorithm and applying a discrete-time Mori–Zwanzig formalism. We present formulas for computing the aforementioned statistics and numerically demonstrate that memory, indeed, improves results for a two-dimensional triple well and the AIB_9 peptide.

II. THEORY

Here, we briefly review DGA as previously presented.[6,7] We then recast it as an iteration to show that its solutions can be viewed as projections; the iteration and resulting projection operators lead naturally to a discrete-time GME and, in turn, an improved estimator. This estimator can be used without iteration.

A. Form of the problem

Let X_t denote a time-homogeneous ergodic Markov process at time t. Our goal is to compute a statistic u of this process that satisfies an equation of the following form:

$$\dot{\mathcal{A}}^0 u = -\dot{b}^0, \tag{1}$$

where $\dot{\mathcal{A}}^0$ is a linear operator that describes the time evolution of expectations of functions of X_t (akin to a time derivative) and $\dot{b}^0$ is the rate of change in u at X_t. We can interpret (1) as finding a time-invariant solution to the equation specifying the dynamics of u,

$$\frac{\mathrm{d}u_t}{\mathrm{d}t} = \dot{\mathcal{A}}^0 u_t + \dot{b}^0. \tag{2}$$

Integrating (2) over a time interval $\tau > 0$ that we call the lag time yields the Richardson iteration,

$$u_t = u_{t-\tau} + (\mathcal{A}^\tau u_{t-\tau} + b^\tau), \tag{3}$$

where $\mathcal{A}^\tau = \int_0^\tau \mathcal{C}^{\tau'} \dot{\mathcal{A}}^0 \,\mathrm{d}\tau' = \mathcal{C}^\tau - \mathcal{I}$, $b^\tau = \int_0^\tau \mathcal{C}^{\tau'} \dot{b}^0 \,\mathrm{d}\tau'$, $\mathcal{C}^\tau = e^{\dot{\mathcal{A}}^0 \tau}$, and $\mathcal{I}$ is the identity operator. Equation (3), as we discuss further in Sec. II C, corresponds to an iteration in u that is consistent with the dynamics in (2). The fixed point of the iteration is the finite time analog of (1),

$$\mathcal{A}^\tau u = -b^\tau. \tag{4}$$

For example, the mean time to first enter the set B starting from state x, $u(x) := m(x) = \mathbf{E}[T_B \mid X_0 = x]$, where $T_B = \min\{t \geq 0 \mid X_t \in B\}$, satisfies[7]

$$(\mathcal{S}_B^\tau - \mathcal{I})m = -\int_0^\tau \mathcal{S}_B^t \mathbf{1}_{B^C} \,\mathrm{d}t, \tag{5}$$

subject to the boundary condition $m(x) = 0$ for $x \in B$. Here,

$$\mathcal{S}_B^\tau f(x) = \mathbf{E}[f(X_{\tau \wedge T_B}) \mid X_0 = x] \tag{6}$$

is the stopped transition operator,

$$\int_0^\tau \mathcal{S}_B^t \mathbf{1}_{B^C}(x) \,\mathrm{d}t = \mathbf{E}[\tau \wedge T_B \mid X_0 = x], \tag{7}$$

$\tau \wedge T_B = \min\{\tau, T_B\}$, and $\mathbf{1}_{B^C}(x)$ is an indicator function that equals one for $x \in B^C$ and zero for $x \notin B^C$. We define other statistics of interest in Sec. III.

B. Dynamical Galerkin approximation (DGA)

The DGA solves (4) by expanding u in a basis and estimating the resulting matrix elements by averages over random samples. We represent the basis by a vector ϕ of functions that satisfy homogeneous boundary conditions. Then, we approximate u by

$$\hat{u} = \hat{u}_0 + \phi^\mathsf{T} v, \tag{8}$$

where $\hat{u}_0$ is a "guess function" for u that satisfies the boundary conditions, v is a vector of coefficients, and we use the symbol $\hat{}$ to denote functions that can be represented by (8). To solve for the coefficients, we substitute (8) into (4), multiply from the left by ϕ, and integrate over an arbitrary distribution of samples of initial conditions μ. This yields

$$G^\tau v = -h^\tau, \tag{9}$$

where

$$G^\tau = \langle \phi, \mathcal{A}^\tau \phi^\mathsf{T} \rangle, \tag{10}$$

$$h^\tau = \langle \phi, \mathcal{A}^\tau \hat{u}_0 + b^\tau \rangle, \tag{11}$$

ALGORITHM 1. DGA.

$$\begin{aligned} G^\tau &= \langle \phi, \mathcal{A}^\tau \phi^\mathsf{T} \rangle \\ h^\tau &= \langle \phi, \mathcal{A}^\tau \hat{u}_0 + b^\tau \rangle \\ v &= -(G^\tau)^{-1} h^\tau \\ \hat{u}^\tau &= \hat{u}_0 + \phi^\mathsf{T} v \\ u^\tau &= \hat{u}^\tau + (\mathcal{A}^\tau \hat{u}^\tau + b^\tau) \end{aligned}$$

and $\langle f, g\rangle = \int f(x)g(x)\mu(x)\,\mathrm{d}x$. The matrix G^τ and vector h^τ are estimated from averages over trajectories of length τ; explicit expressions are given in Sec. III. Given these estimates, (9) can be solved directly for v. DGA is summarized in Algorithm 1. We select τ by increasing its value until the estimate of u appears to converge.

C. Iterative solution

The form in (3) and our recent use of it in the context of inexact numerical linear algebra[31] suggest that one can view DGA as the fixed point of the *inexact* iteration,

$$u_t^\tau = \hat{u}_{t-\tau}^\tau + (\mathcal{A}^\tau \hat{u}_{t-\tau}^\tau + b^\tau), \tag{12a}$$

$$\hat{u}_t^\tau = \hat{u}_{t-\tau}^\tau + \mathcal{P}(u_t^\tau - \hat{u}_{t-\tau}^\tau), \tag{12b}$$

where $\mathcal{P}$ is an operator that projects functions onto linear combinations of basis functions,

$$\mathcal{P}f = \phi^\mathsf{T}(K^0)^{-1}\langle \phi, f\rangle \tag{13}$$

and

$$K^t = \langle \phi, \mathcal{C}^t \phi^\mathsf{T}\rangle. \tag{14}$$

We use the superscript τ on u_t^τ to distinguish that it comes from the inexact iteration (12) involving the projected function $\hat{u}_t^\tau$ (and thus a layer of approximation) rather than the exact iteration (3). The fixed points are indicated without subscripts: $u^\tau = \lim_{t\to\infty} u_t^\tau$ and $\hat{u}^\tau = \lim_{t\to\infty} \hat{u}_t^\tau$.

We now identify the approximation associated with DGA by comparing the inexact iteration (12) with the *exact* iteration,

$$u_t = u_{t-\tau} + (\mathcal{A}^\tau u_{t-\tau} + b^\tau), \tag{15a}$$

$$\hat{u}_t = \hat{u}_{t-\tau} + \mathcal{P}(u_t - \hat{u}_{t-\tau}). \tag{15b}$$

Let us define the complementary projection

$$\mathcal{Q} = \mathcal{I} - \mathcal{P}. \tag{16}$$

We note that $\mathcal{Q}(\hat{u}_t - \hat{u}_{t'}) = 0$ for any t and t' since $\mathcal{Q}\phi = 0$ and $\hat{u}_t - \hat{u}_{t'} = \phi^\mathsf{T} v$ for some v (we work with differences because we do not assume $\mathcal{Q}\hat{u}_0 = 0$). By substituting $\mathcal{P} = \mathcal{I} - \mathcal{Q}$ into (15b) and replacing t with $t - \tau$, we find that $u_{t-\tau} = \hat{u}_{t-\tau} + \mathcal{Q}(u_{t-\tau} - \hat{u}_{t-2\tau})$; in turn, by substituting this expression into (15a), we can express (15) as

$$u_t = \hat{u}_{t-\tau} + (\mathcal{A}^\tau \hat{u}_{t-\tau} + b^\tau) + \mathcal{C}^\tau \mathcal{Q}(u_{t-\tau} - \hat{u}_{t-2\tau}), \tag{17a}$$

$$\hat{u}_t = \hat{u}_{t-\tau} + \mathcal{P}(u_t - \hat{u}_{t-\tau}). \tag{17b}$$

The fixed point $\hat{u} = \lim_{t\to\infty} \hat{u}_t$ satisfies $\mathcal{P}(\mathcal{A}^\tau \hat{u} + b^\tau) + \mathcal{P}\mathcal{C}^\tau \mathcal{Q}(u - \hat{u}) = 0$. Comparing (12a) and (15a) shows that they both use the previous estimate ($\hat{u}_{t-\tau}^\tau$ or $u_{t-\tau}^\tau$, respectively) to construct an improved estimate u_t^τ. Because $\hat{u}_{t-\tau}^\tau$ can be represented using the basis, we only need length τ trajectories to approximate u_t^τ. Hence, DGA makes the approximation $\mathcal{P}\mathcal{C}^\tau \mathcal{Q}(u - \hat{u}) \approx 0$, which avoids the need for trajectories longer than τ. In contrast, the exact iteration requires infinite-length trajectories since the last term of (17a) prevents $\hat{u}_t$ from depending on $\hat{u}_{t-\tau}$ alone.

To understand the need for memory, we observe that (15a) specifies the Markovian dynamics for u_t: the value of u_t at the current iteration t is fully determined by the value of $u_{t-\tau}$ at the last iteration $t - \tau$. DGA amounts to making the approximation that the dynamics of the *projected* function $\hat{u}_t$ observed at discrete time intervals τ is Markovian. This perspective suggests that we can derive an improved estimator by mitigating the Markov approximation. We do this by using projected statistics at multiple times (i.e., using memory), rather than a single time, as we now describe.

D. Iterative solution with memory

Our starting point is (17a) with τ replaced by σ, where τ/σ is a positive integer,

$$u_t = \hat{u}_{t-\sigma} + (\mathcal{A}^\sigma \hat{u}_{t-\sigma} + b^\sigma) + \mathcal{C}^\sigma \mathcal{Q}(\underline{u_{t-\sigma}} - \hat{u}_{t-2\sigma}). \tag{18}$$

The underlined term is not representable by the basis and prevents solving for $\hat{u}$ by a fixed point iteration with length τ trajectories. By repeatedly substituting (18), we obtain

$$\begin{aligned} u_t &= \hat{u}_{t-\sigma} + \sum_{n=1}^{2} (\mathcal{C}^\sigma \mathcal{Q})^{n-1}(\mathcal{A}^\sigma \hat{u}_{t-n\sigma} + b^\sigma) \\ &\quad + (\mathcal{C}^\sigma \mathcal{Q})^2(\underline{u_{t-2\sigma}} - \hat{u}_{t-3\sigma}) \\ &\quad \vdots \\ &= \hat{u}_{t-\sigma} + \sum_{n=1}^{\tau/\sigma} (\mathcal{C}^\sigma \mathcal{Q})^{n-1}(\mathcal{A}^\sigma \hat{u}_{t-n\sigma} + b^\sigma) \\ &\quad + (\mathcal{C}^\sigma \mathcal{Q})^{\tau/\sigma}(\underline{u_{t-\tau}} - \hat{u}_{t-\tau-\sigma}), \end{aligned} \tag{19}$$

where the underline indicates the location of the next substitution. We now have an alternative iteration to (17) that depends on the last τ/σ projections,

$$\begin{aligned} u_t &= \hat{u}_{t-\sigma} + \sum_{n=1}^{\tau/\sigma} (\mathcal{C}^\sigma \mathcal{Q})^{n-1}(\mathcal{A}^\sigma \hat{u}_{t-n\sigma} + b^\sigma) \\ &\quad + (\mathcal{C}^\sigma \mathcal{Q})^{\tau/\sigma}(u_{t-\tau} - \hat{u}_{t-\tau-\sigma}), \end{aligned} \tag{20a}$$

$$\hat{u}_t = \hat{u}_{t-\sigma} + \mathcal{P}(u_t - \hat{u}_{t-\sigma}). \tag{20b}$$

Equation (20a) is similar to a discrete-time GME (e.g., Ref. 27) or a discrete-time Mori–Zwanzig decomposition (e.g., Ref. 32 for eigendecomposition of the transition operator). It describes u_t at iteration t by its projection $\hat{u}_{t-\sigma}$ at iteration $t - \sigma$ (*Markov approximation*, first term and $n = 1$ of the second term), the projections $\hat{u}_{t-n\sigma}$ at previous iterations $t - n\sigma$ (*memory*, $n > 1$ of the second term), and the complementary projection $\mathcal{Q}u_{t-\tau}$ at time $t - \tau$ (*orthogonal dynamics*, last term). The approximation used in DGA is the first two terms of (20a) with $\sigma = \tau$ (i.e., the previous solution and the first term of the sum). The remaining terms correct for the Markov approximation by introducing memory.

Omitting the last term of (20a) yields a computable iteration,

ALGORITHM 2. DGA with memory.

for $n \in \{1, \ldots, \tau/\sigma\}$ **do**
$K^{n\sigma} = \langle \phi, \mathcal{C}^{n\sigma}\phi^{\mathsf{T}}\rangle$
$G^{n\sigma} = \langle \phi, \mathcal{A}^{n\sigma}\phi^{\mathsf{T}}\rangle$
$h^{n\sigma} = \langle \phi, \mathcal{A}^{n\sigma}\hat{u}_0 + b^{n\sigma}\rangle$
$G^{\sigma,n\sigma} = G^{n\sigma} - \sum_{n'=1}^{n-1} K^{(n-n')\sigma}(K^0)^{-1}G^{\sigma,n'\sigma}$
$h^{\sigma,n\sigma} = h^{n\sigma} - \sum_{n'=1}^{n-1} K^{(n-n')\sigma}(K^0)^{-1}h^{\sigma,n'\sigma}$
end for
$v = -(G^{\sigma,\tau})^{-1}h^{\sigma,\tau}$
$\hat{u}^{\sigma,\tau} = \hat{u}_0 + \phi^{\mathsf{T}}v$
$u^{\sigma,\tau} = \hat{u}^{\sigma,\tau} + (\mathcal{A}^{\tau}\hat{u}^{\sigma,\tau} + b^{\tau})$
$\quad - \sum_{n=1}^{\tau/\sigma} \mathcal{C}^{\tau-n\sigma}\phi^{\mathsf{T}}(K^0)^{-1}(G^{\sigma,n\sigma}v + h^{\sigma,n\sigma})$

$$u_t^{\sigma,\tau} = \hat{u}_{t-\sigma}^{\sigma,\tau} + \sum_{n=1}^{\tau/\sigma} (\mathcal{C}^{\sigma}\mathcal{Q})^{n-1}(\mathcal{A}^{\sigma}\hat{u}_{t-n\sigma}^{\sigma,\tau} + b^{\sigma}), \tag{21a}$$

$$\hat{u}_t^{\sigma,\tau} = \hat{u}_{t-\sigma}^{\sigma,\tau} + \mathcal{P}(u_t^{\sigma,\tau} - \hat{u}_{t-\sigma}^{\sigma,\tau}), \tag{21b}$$

with a fixed point that solves

$$\sum_{n=1}^{\tau/\sigma} \mathcal{P}(\mathcal{C}^{\sigma}\mathcal{Q})^{n-1}(\mathcal{A}^{\sigma}\hat{u}^{\sigma,\tau} + b^{\sigma}) = 0. \tag{22}$$

Comparing (20) and (22), we see that the latter corresponds to making the approximation $\mathcal{P}(\mathcal{C}^{\sigma}\mathcal{Q})^{\tau/\sigma}(u - \hat{u}) \approx 0$. When the basis captures much of the slower dynamics of the system, we expect the DGA with memory approximation $\mathcal{P}(\mathcal{C}^{\sigma}\mathcal{Q})^{\tau/\sigma}(u - \hat{u}) \approx 0$ to be better than the DGA approximation $\mathcal{P}\mathcal{C}^{\tau}\mathcal{Q}(u - \hat{u}) \approx 0$ because the additional applications of $\mathcal{Q}$ remove these slower dynamics. We show this numerically in Sec. IV.

E. DGA with memory

Here, we derive the DGA with memory algorithm, which we summarize in Algorithm 2. As in (8), we expand $\hat{u}^{\sigma,\tau}$ in a basis. Substituting $\hat{u}^{\sigma,\tau} = \hat{u}_0 + \phi^{\mathsf{T}}v$ into (22) and applying $\langle \phi_i, \mathcal{P}f\rangle = \langle \phi_i, f\rangle$ leads to the linear system,

$$G^{\sigma,\tau}v = -h^{\sigma,\tau}, \tag{23}$$

where

$$G^{\sigma,\tau} = \sum_{n=1}^{\tau/\sigma} \langle \phi, (\mathcal{C}^{\sigma}\mathcal{Q})^{n-1}\mathcal{A}^{\sigma}\phi^{\mathsf{T}}\rangle, \tag{24}$$

$$h^{\sigma,\tau} = \sum_{n=1}^{\tau/\sigma} \langle \phi, (\mathcal{C}^{\sigma}\mathcal{Q})^{n-1}(\mathcal{A}^{\sigma}\hat{u}_0 + b^{\sigma})\rangle. \tag{25}$$

We can calculate matrices $G^{\sigma,\tau}$ and $h^{\sigma,\tau}$ by expanding each $\mathcal{Q}$ in the expectations as $\langle f, \mathcal{Q}g\rangle = \langle f, g\rangle - \langle f, \mathcal{P}g\rangle = \langle f, g\rangle - \langle f, \phi^{\mathsf{T}}\rangle(K^0)^{-1}\langle \phi, g\rangle$ until none remain. For instance, to calculate the term $h^{\sigma,3\sigma}$, we expand

$$\begin{aligned}
h^{\sigma,3\sigma} &= \langle \phi, b^{\sigma} + \mathcal{C}^{\sigma}\mathcal{Q}b^{\sigma} + (\mathcal{C}^{\sigma}\mathcal{Q})^2 b^{\sigma}\rangle \\
&= \langle \phi, b^{\sigma} + \mathcal{C}^{\sigma}b^{\sigma} + \mathcal{C}^{2\sigma}b^{\sigma}\rangle \\
&\quad - \langle \phi, \mathcal{C}^{\sigma}\mathcal{P}(b^{\sigma} + \mathcal{C}^{\sigma}\mathcal{Q}b^{\sigma})\rangle - \langle \phi, \mathcal{C}^{2\sigma}\mathcal{P}b^{\sigma}\rangle \\
&= h^{3\sigma} - K^{\sigma}(K^0)^{-1}\underline{h^{\sigma,2\sigma}} - K^{2\sigma}(K^0)^{-1}h^{\sigma}, \\
h^{\sigma,2\sigma} &= \langle \phi, b^{\sigma} + \mathcal{C}^{\sigma}\mathcal{Q}b^{\sigma}\rangle \\
&= \langle \phi, b^{\sigma} + \mathcal{C}^{\sigma}b^{\sigma}\rangle - \langle \phi, \mathcal{C}^{\sigma}\mathcal{P}b^{\sigma}\rangle \\
&= h^{2\sigma} - K^{\sigma}(K^0)^{-1}h^{\sigma}.
\end{aligned}$$

The underlined expectations contain $\mathcal{Q}$ and must be expanded in terms of matrices that do not contain $\mathcal{Q}$, which can be directly evaluated as averages over random samples. We have rearranged and grouped terms for numerical stability because $G^{\sigma,\tau}$ and $h^{\sigma,\tau}$ can be very sensitive to how they are computed. Using this approach, we can derive the following iterative formulas:

$$G^{\sigma,\tau} = G^{\tau} - \sum_{n=1}^{(\tau/\sigma)-1} K^{\tau-n\sigma}(K^0)^{-1}G^{\sigma,n\sigma}, \tag{26a}$$

$$h^{\sigma,\tau} = h^{\tau} - \sum_{n=1}^{(\tau/\sigma)-1} K^{\tau-n\sigma}(K^0)^{-1}h^{\sigma,n\sigma}. \tag{26b}$$

A similar iteration was used to calculate the discretized memory kernel in Ref. 27 and the transfer tensors in Ref. 33.

After we solve (23), we can compute a stochastic approximation of u,

$$u^{\sigma,\tau} = \hat{u}^{\sigma,\tau} + \sum_{n=1}^{\tau/\sigma} (\mathcal{C}^{\sigma}\mathcal{Q})^{n-1}(\mathcal{A}^{\sigma}\hat{u}^{\sigma,\tau} + b^{\sigma}), \tag{27}$$

which is the fixed point of (21a). In this case, we expand each $\mathcal{Q}f = f - \phi^{\mathsf{T}}(K^0)^{-1}\langle \phi, f\rangle$ from left to right until only $\mathcal{Q}$s inside expectations remain (which can be grouped into matrices $G^{\sigma,n\sigma}$ and $h^{\sigma,n\sigma}$),

$$u^{\sigma,\tau} = \hat{u}^{\sigma,\tau} + (\mathcal{A}^{\tau}\hat{u}^{\sigma,\tau} + b^{\tau}) - \sum_{n=1}^{\tau/\sigma} \mathcal{C}^{\tau-n\sigma}\delta u_n^{\sigma,\tau}, \tag{28}$$

where $\delta u_n^{\sigma,\tau} = \phi^{\mathsf{T}}(K^0)^{-1}(G^{\sigma,n\sigma}v + h^{\sigma,n\sigma})$. As in (26), we have combined terms in (28) for numerical stability.

We select σ and τ by increasing τ with τ/σ fixed until a statistic appears to converge; we try this for a few small integer values of τ/σ. As we show numerically in Sec. IV, only a few correction terms (i.e., small τ/σ) are generally needed to obtain most of the benefit of including memory.

III. DYNAMICAL STATISTICS

Here, we introduce the statistics that we want to compute and the equations that they solve. We describe a variety of statistics beyond the MFPT that can be solved using DGA or DGA with memory. Provided that a statistic can be written in the form (4), we can solve for it with DGA or DGA with memory by evaluating (9) or (23), respectively, with appropriate K^t, G^t, and h^t matrices.

A. Stationary distribution

The stationary distribution π is invariant with respect to time translation, and therefore,

$$\int \pi(x) f(x)\,\mathrm{d}x = \int \pi(x)\,\mathcal{T}^t f(x)\,\mathrm{d}x \tag{29}$$

for all functions f, where the transition operator

$$\mathcal{T}^t f(x) = \mathbf{E}[f(X_t) \mid X_0 = x] \tag{30}$$

propagates the expectations of functions f forward-in-time. The sampling distribution μ can be reweighted to obtain the stationary distribution π via $w = \pi/\mu$,[7,34] which satisfies an operator equation in the form of (4),

$$((\mathcal{T}^t)^\dagger - \mathcal{I})w = 0, \tag{31}$$

where $(\mathcal{T}^t)^\dagger$ denotes the adjoint of $\mathcal{T}^t$ with respect to μ: $\langle f, (\mathcal{T}^t)^\dagger g\rangle = \langle g, \mathcal{T}^t f\rangle$ for all functions f and g. We expand $\hat{w} = \hat{w}_0 + \phi^\mathsf{T} v$, where $\int \hat{w}_0(x)\mu(x)\,\mathrm{d}x = 1$ and $\int \phi(x)\mu(x)\,\mathrm{d}x = 0$; these conditions ensure that the resulting estimates of $\hat{w}$ and w are normalized. The corresponding matrices are

$$\begin{aligned} K^t &= \langle \phi, (\mathcal{T}^t)^\dagger \phi^\mathsf{T}\rangle \\ &= \mathbf{E}[\phi(X_t)\phi^\mathsf{T}(X_0) \mid X_0 \sim \mu], \end{aligned} \tag{32a}$$

$$\begin{aligned} G^t &= \langle \phi, ((\mathcal{T}^t)^\dagger - \mathcal{I})\phi^\mathsf{T}\rangle \\ &= \mathbf{E}[(\phi(X_t) - \phi(X_0))\phi^\mathsf{T}(X_0) \mid X_0 \sim \mu], \end{aligned} \tag{32b}$$

$$\begin{aligned} h^t &= \langle \phi, ((\mathcal{T}^t)^\dagger - \mathcal{I})\hat{w}_0\rangle \\ &= \mathbf{E}[(\phi(X_t) - \phi(X_0))\hat{w}_0(X_0) \mid X_0 \sim \mu], \end{aligned} \tag{32c}$$

and we approximate the expectations of functions f with respect to π as

$$\langle f, w^{\sigma,\tau}\rangle = \mathbf{E}\left[f(X_\tau)\hat{w}^{\sigma,\tau}(X_0) - \sum_{n=1}^{\tau/\sigma} f(X_{\tau-n\sigma})\delta w_n^{\sigma,\tau}(X_0)\,\middle|\, X_0 \sim \mu\right]. \tag{33}$$

B. Mean first passage time

The mean first passage time (MFPT)

$$m(x) = \mathbf{E}[T_B \mid X_0 = x] \tag{34}$$

is the expected time until a trajectory starting at x first enters B. As noted above, it satisfies an equation of the form (4),

$$(\mathcal{S}_B^t - \mathcal{I})m = -\int_0^t \mathcal{S}_B^{t'} \mathbf{1}_{B^\mathsf{C}}\,\mathrm{d}t', \tag{35}$$

subject to the boundary condition $m(x) = 0$ for $x \in B$ [cf. (5)–(7)]. Physically, this equation states that applying $\mathcal{S}_B^t$ to $m(x)$ reduces the mean remaining time to enter by $\mathbf{E}[t \wedge T_B \mid X_0 = x]$. The corresponding matrices are

$$\begin{aligned} K^t &= \langle \phi, \mathcal{S}_B^t \phi^\mathsf{T}\rangle \\ &= \mathbf{E}[\phi(X_0)\phi^\mathsf{T}(X_{t\wedge T_B}) \mid X_0 \sim \mu], \end{aligned} \tag{36a}$$

$$\begin{aligned} G^t &= \langle \phi, (\mathcal{S}_B^t - \mathcal{I})\phi^\mathsf{T}\rangle \\ &= \mathbf{E}[\phi(X_0)(\phi^\mathsf{T}(X_{t\wedge T_B}) - \phi^\mathsf{T}(X_0)) \mid X_0 \sim \mu], \end{aligned} \tag{36b}$$

$$\begin{aligned} h^t &= \left\langle \phi, (\mathcal{S}_B^t - \mathcal{I})\hat{m}_0 + \int_0^t \mathcal{S}_B^{t'} \mathbf{1}_{B^\mathsf{C}}\,\mathrm{d}t' \right\rangle \\ &= \mathbf{E}[\phi(X_0)(\hat{m}_0(X_{t\wedge T_B}) - \hat{m}_0(X_0) + (t \wedge T_B)) \mid X_0 \sim \mu], \end{aligned} \tag{36c}$$

and m can be approximated from its projection using

$$m^{\sigma,\tau}(x) = \mathbf{E}\left[\hat{m}^{\sigma,\tau}(X_{\tau\wedge T_B}) + (\tau \wedge T_B) - \sum_{n=1}^{\tau/\sigma} \delta m_n^{\sigma,\tau}(X_{(\tau-n\sigma)\wedge T_B})\,\middle|\, X_0 = x\right]. \tag{37}$$

C. Forward committor

The forward committor

$$q(x) = \mathbf{E}[\mathbf{1}_B(X_{T_{A\cup B}}) \mid X_0 = x] \tag{38}$$

is the probability that a trajectory starting at x will enter B before A. The forward committor satisfies

$$(\mathcal{S}_{A\cup B}^t - \mathcal{I})q = 0 \tag{39}$$

with boundary condition $q(x) = \mathbf{1}_B(x)$ for $x \in A \cup B$. The corresponding matrices are

$$\begin{aligned} K^t &= \langle \phi, \mathcal{S}_{A\cup B}^t \phi^\mathsf{T}\rangle \\ &= \mathbf{E}[\phi(X_0)\phi(X_{t\wedge T_{A\cup B}}) \mid X_0 \sim \mu], \end{aligned} \tag{40a}$$

$$\begin{aligned} G^t &= \langle \phi, (\mathcal{S}_{A\cup B}^t - \mathcal{I})\phi^\mathsf{T}\rangle \\ &= \mathbf{E}[\phi(X_0)(\phi(X_{t\wedge T_{A\cup B}}) - \phi(X_0)) \mid X_0 \sim \mu], \end{aligned} \tag{40b}$$

$$\begin{aligned} h^t &= \langle \phi, (\mathcal{S}_{A\cup B}^t - \mathcal{I})\hat{q}_0\rangle \\ &= \mathbf{E}[\phi(X_0)(\hat{q}_0(X_{t\wedge T_{A\cup B}}) - \hat{q}_0(X_0)) \mid X_0 \sim \mu], \end{aligned} \tag{40c}$$

and q can be approximated as

$$q^{\sigma,\tau}(x) = \mathbf{E}\left[\hat{q}^{\sigma,\tau}(X_{\tau\wedge T_{A\cup B}}) - \sum_{n=1}^{\tau/\sigma} \delta q_n^{\sigma,\tau}(X_{(\tau-n\sigma)\wedge T_{A\cup B}})\,\middle|\, X_0 = x\right]. \tag{41}$$

D. Backward committor

The backward committor is the probability that a trajectory ending at x exited A after B,

$$\tilde{q}(x) = \mathbf{E}[\mathbf{1}_A(X_{-\tilde{T}_{A\cup B}}) \mid X_0 = x], \tag{42}$$

where $\tilde{T}_{A\cup B} = \min\{t \geq 0 \mid X_{-t} \in A \cup B\}$. The backward committor satisfies

$$(\mathcal{S}^{-t}_{A\cup B} - \mathcal{I})\tilde{q} = 0 \tag{43}$$

with boundary condition $\tilde{q}(x) = \mathbf{1}_A(x)$ for $x \in A \cup B$.

$$\mathcal{S}^{-t}_{A\cup B} f(x) = \mathbf{E}[f(X_{-(t\wedge\tilde{T}_{A\cup B})}) \mid X_0 = x] \tag{44}$$

is the stopped transition operator for the time-reversed process. We take conditional expectations backward-in-time with respect to the *time-reversed process*: given an infinite, statistically stationary trajectory X, we look backward in time from each time t, where $X_t = x$. Mathematically,

$$\begin{aligned}&\mathbf{E}[f(X_{-(t\wedge\tilde{T}_{A\cup B})}) \mid X_0 = x] \\ &\quad = \int \mathbf{E}[f(X_{-(t\wedge\tilde{T}_{A\cup B})}) \mid X_0 = x, X_{-\tau} = x']\pi(x')\,\mathrm{d}x',\end{aligned} \tag{45}$$

which follows from the definition of conditional probability and $\pi(x) = \mathbf{P}[X_{-\tau} = x]$. Because $\pi(x) = w(x)\mu(x) = \mathbf{P}[X_0 = x] = \mathbf{P}[X_{-\tau} = x]$,

$$\begin{aligned}\langle g, \mathcal{S}^{-t}_{A\cup B} f\rangle &= \int \mathbf{E}[g(X_0) f(X_{-(t\wedge\tilde{T}_{A\cup B})}) \mid X_0 = x]\mu(x)\,\mathrm{d}x \\ &= \mathbf{E}\left[g(X_0) f(X_{-(t\wedge\tilde{T}_{A\cup B})}) \frac{w(X_{-\tau})}{w(X_0)} \middle| X_{-\tau} \sim \mu\right],\end{aligned} \tag{46}$$

as we show explicitly in Ref. 7 (with the zero of time shifted). The corresponding matrices are

$$\begin{aligned}K^t &= \langle \phi, \mathcal{S}^{-t}_{A\cup B}\phi^{\mathsf{T}}\rangle \\ &= \mathbf{E}\left[\phi(X_0)\phi(X_{-(t\wedge\tilde{T}_{A\cup B})}) \frac{w(X_{-\tau})}{w(X_0)} \middle| X_{-\tau} \sim \mu\right],\end{aligned} \tag{47a}$$

$$\begin{aligned}G^t &= \langle \phi, (\mathcal{S}^{-t}_{A\cup B} - \mathcal{I})\phi^{\mathsf{T}}\rangle \\ &= \mathbf{E}\left[\phi(X_0)(\phi(X_{-(t\wedge\tilde{T}_{A\cup B})}) - \phi(X_0)) \frac{w(X_{-\tau})}{w(X_0)} \middle| X_{-\tau} \sim \mu\right],\end{aligned} \tag{47b}$$

$$\begin{aligned}h^t &= \langle \phi, (\mathcal{S}^{-t}_{A\cup B} - \mathcal{I})\hat{\tilde{q}}_0\rangle \\ &= \mathbf{E}\left[\phi(X_0)(\hat{\tilde{q}}_0(X_{-(t\wedge\tilde{T}_{A\cup B})}) - \hat{\tilde{q}}_0(X_0)) \frac{w(X_{-\tau})}{w(X_0)} \middle| X_{-\tau} \sim \mu\right].\end{aligned} \tag{47c}$$

We estimate $\tilde{q}$ as

$$\tilde{q}^{\sigma,\tau}(x) = \mathbf{E}\left[\hat{\tilde{q}}^{\sigma,\tau}(X_{-(\tau\wedge\tilde{T}_{A\cup B})}) - \sum_{n=1}^{\tau/\sigma} \delta\tilde{q}_n^{\sigma,\tau}(X_{-((\tau-n\sigma)\wedge\tilde{T}_{A\cup B})}) \middle| X_0 = x\right]. \tag{48}$$

IV. NUMERICAL EXAMPLES

Here, we demonstrate our method on two systems. First, we illustrate our method with a two-dimensional triple-well potential, for which an exact reference can be calculated using finite differences. Next, we test our method on a more complex system, AIB_9, for which we have long, unbiased trajectories as a reference.

A. Two-dimensional triple-well

Here, we illustrate DGA with memory on a simple model system. The system is a drift–diffusion process obeying the Fokker–Planck equation,

$$\partial_t p_t(x) = \nabla \cdot (p_t(x)\nabla V(x)) + \beta^{-1}\nabla^2 p_t(x), \tag{49}$$

where β is the inverse temperature, which we set to $\beta = 2$, $x = (\chi_1, \chi_2)$, and V is the potential[35]

$$\begin{aligned}V(\chi_1,\chi_2) = {}& 3e^{-\chi_1^2-(\chi_2-1/3)^2} - 3e^{-\chi_1^2-(\chi_2-5/3)^2} \\ & - 5e^{-(\chi_1-1)^2-\chi_2^2} - 5e^{-(\chi_1+1)^2-\chi_2^2} \\ & + 0.2\chi_1^4 + 0.2(\chi_2 - 1/3)^4.\end{aligned} \tag{50}$$

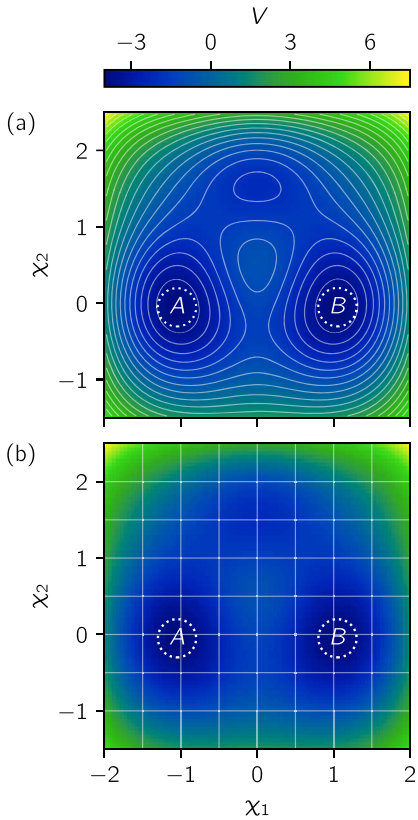

FIG. 1. Potential energy surface of the two-dimensional triple well. The boundaries of sets A and B are indicated using dotted lines. (a) Contours are drawn every $0.5\beta^{-1}$. (b) 8×8 grid basis used for DGA and DGA with memory.

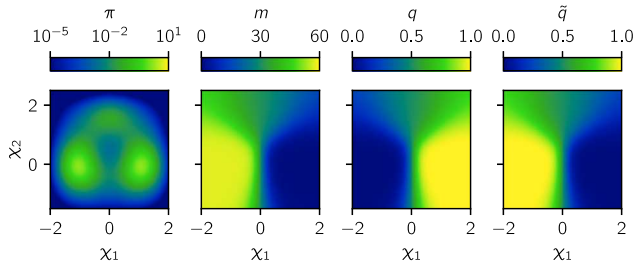

FIG. 2. Reference statistics for the two-dimensional triple well.

We define the reactant state A and product state B as circles with radii 0.25 centered at $(\pm 1.05, -0.05)$, respectively [Fig. 1(a)].

We discretize the system on an 80×80 grid of equally spaced points (with spacing $h = 0.05$) in the range $[-2, 2] \times [-1.5, 2.5]$. The dynamics of this discretized system follow the master equation,[6,31]

$$\partial_t p_t(x) = \sum_{x'} p_t(x')\,\mathcal{L}(x', x), \tag{51}$$

where the elements of the matrix $\mathcal{L}$ are

$$\mathcal{L}(x, x') = \frac{2\beta^{-1}}{h^2} \frac{1}{1 + e^{-\beta(V(x)-V(x'))}} \tag{52}$$

for $x - x' \in \{(0, \pm h), (\pm h, 0)\}$, $\mathcal{L}(x, x) = -\sum_{x' \neq x} \mathcal{L}(x, x')$, and $\mathcal{L}(x, x') = 0$ otherwise. We use $\mathcal{L}$ to calculate finite-time operators, such as $(\mathcal{T}^t)^\dagger$ and $\mathcal{S}_B^t$. These expressions are given in Appendix B. We then directly compute the DGA matrices by taking inner products. For example, we compute (32a) as

$$\langle \phi, (\mathcal{T}^t)^\dagger \phi^\mathsf{T} \rangle = \sum_{x,x'} \mu(x)\phi(x)(\mathcal{T}^t)^\dagger(x, x')\phi^\mathsf{T}(x'),$$

where $(\mathcal{T}^t)^\dagger f(x) = \sum_{x'} (\mathcal{T}^t)^\dagger(x, x') f(x')$.

We calculate reference statistics by directly solving (31), (35), (39), and (43). For the DGA calculations, we use a basis of 64 indicator functions on an equally spaced 8×8 grid in the range $[-2, 2] \times [-1.5, 2.5]$, as shown in Fig. 1(b). Operationally, we first calculate the stationary distribution π using (33) with an arbitrary distribution of initial conditions and $\hat{w}_0 = 1$; we subtract the mean from each basis function to normalize π. For other statistics, we use the resulting stationary distribution (estimated with the same σ and τ) for μ, and we set the value of each basis function to zero for points in B (for m) or $A \cup B$ (for q and $\tilde{q}$). We use guess functions $\hat{m}_0 = 0$, $\hat{q}_0 = \mathbf{1}_B$, and $\hat{\tilde{q}}_0 = \mathbf{1}_A$.

In Fig. 2, we show the stationary distribution (π), the MFPT to B (m), and the forward and backward committors for the reaction A to B (q and $\tilde{q}$). The improvement due to memory is most noticeable in the MFPT, which we plot in Fig. 3. For DGA ($\tau/\sigma = 1$), a lag time of $\tau = 1$ is required for $m^{\sigma,\tau}$ to qualitatively match the reference in Fig. 2. Meanwhile, for DGA with memory ($\tau/\sigma = 5$), $\tau = 0.05$ is sufficient to obtain a reasonable approximation of the reference. When trajectory length is the limiting factor, this shorter lag time directly translates to an order of magnitude savings in simulation time.

We look at the effect of varying parameters σ and τ in Fig. 4. We focus on the rate constant k_{AB} for the transition from A to B because it is the most sensitive to errors; we calculate it as the inverse of the MFPT from configurations in the A state,[36]

$$k_{AB}^{-1} = \int_A m(x)\pi(x)\,\mathrm{d}x. \tag{53}$$

For this system, the true inverse rate constant is $k_{AB}^{-1} \approx 57$. In Fig. 4(a), keeping the number of memory terms constant (by fixing τ/σ) and increasing τ makes the rate constant converge to the correct value. The best improvement that DGA with memory can give over DGA for a particular τ (which is limited by the dataset) can be seen by comparing the $\tau/\sigma = 1$ (DGA) curve with the $\tau/\sigma = 10$ curve (which is approximately the same as the $\tau/\sigma = \infty$ limit). The inverse rate converges at $\tau \approx 10$ without memory and at $\tau \approx 0.05$ with memory. In Fig. 4(b), keeping σ constant and increasing the number of memory terms also leads to convergence to the correct value. We note that DGA with memory overestimates the inverse rate around $\tau = 0.05$: memory can overcorrect for the error, so one must be careful when assessing convergence since the curve may appear flat at local extrema. In Fig. 4(c), we consider the effect of increasing the number of memory terms with fixed τ. In this case, the result does not converge to the true value but, instead, retains a fixed error when

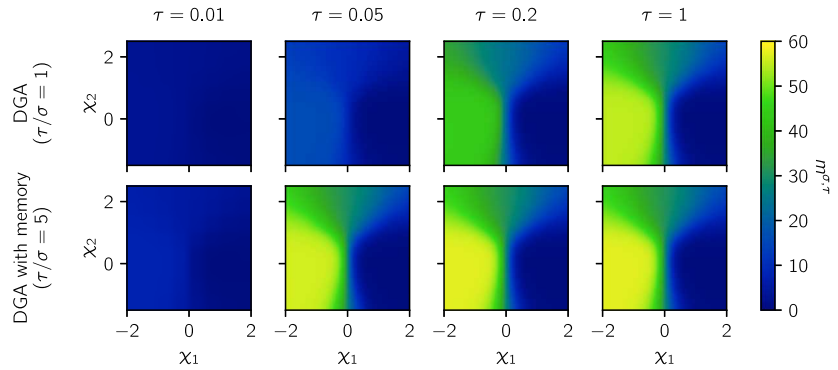

FIG. 3. Mean first passage time for the two-dimensional triple well, estimated using DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 5$). DGA with memory requires shorter lag times to achieve comparable results to DGA.

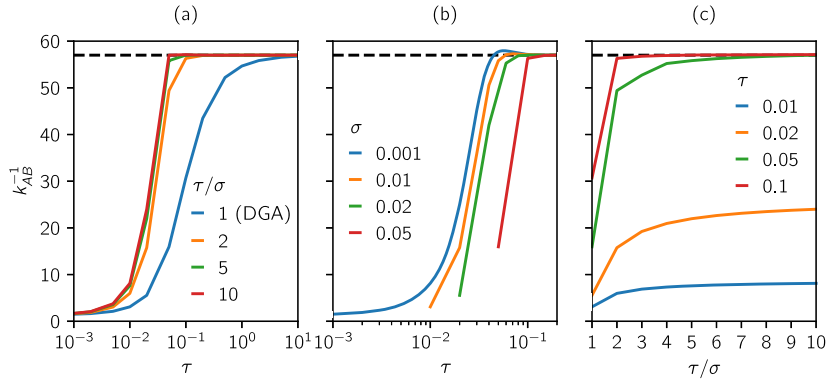

FIG. 4. Dependence of the estimated inverse rate constant k_{AB}^{-1} for the two-dimensional triple well on (a) τ with a fixed number of memory terms, (b) τ with fixed σ, and (c) the number of memory terms with fixed τ. The true value of the inverse rate constant is indicated by the dashed line.

τ is too short for the orthogonal variables to relax (see Appendix A for a graphical explanation).

These results inform our procedure for selecting τ and σ. Namely, we increase τ with fixed τ/σ until statistics of interest converge, as in Fig. 4(a). Our examples suggest that we could, instead, hold σ fixed, as in Fig. 4(b), but we expect this procedure to be more sensitive to statistical errors in more demanding applications because it increases the number of terms in the memory sum.

To demonstrate that these improvements are not specific to the MFPT and the rate constant, we plot the errors in other dynamical statistics in Figs. 5 and 6. We calculate the relative errors for π and m as $\Delta u^{\sigma,\tau}/u$ and for q and $\tilde{q}$ as $\Delta u^{\sigma,\tau}/(u(1-u))$, where $\Delta u^{\sigma,\tau} = u^{\sigma,\tau} - u$. In Fig. 5, we look at the relative error in each dynamical statistic at $\tau = 0.05$. In all cases, we observe that DGA with memory $(\tau/\sigma = 5)$ has a significantly lower error than DGA $(\tau/\sigma = 1)$. We note that DGA has large regions of errors with the same sign, while DGA with memory has smaller regions of oscillating errors from memory corrections with opposite signs. In Fig. 6, we look at the mean of the absolute value of the relative error in regions with $V \leq 0$ (where the system spends the most time; see Fig. 1) for each of the dynamical statistics. In all cases, DGA with memory outperforms DGA at the same lag time, especially at shorter lag times. We observe that, for $\tau/\sigma > 2$, DGA with memory estimates for all statistics converge at $\tau \approx 0.1$, even though each statistic has a different memory kernel. In contrast, for DGA $(\tau/\sigma = 1)$, m requires a much larger τ to converge than other statistics.

B. Left-to-right helix transition in AIB_9

To demonstrate DGA with memory for a molecular system, we analyze the dynamics of AIB_9, a peptide consisting of nine α-aminoisobutyric acid (AIB) residues. AIB is an unnatural achiral amino acid that forms both left-handed and right-handed 3_{10} helices with equal probability. The left-to-right helix transition was previously studied using MSMs and long, unbiased molecular dynamics

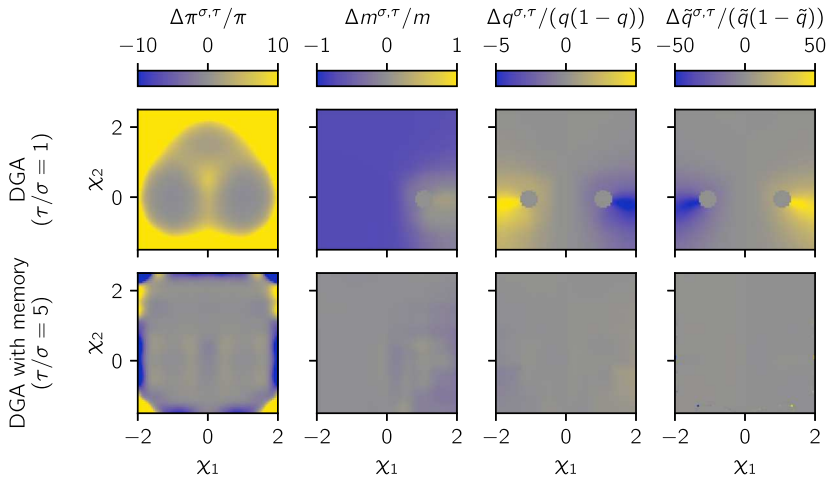

FIG. 5. Relative error in the DGA $(\tau/\sigma = 1)$ and DGA with memory $(\tau/\sigma = 5)$ estimates for each dynamical statistic at $\tau = 0.05$, for the two-dimensional triple well.

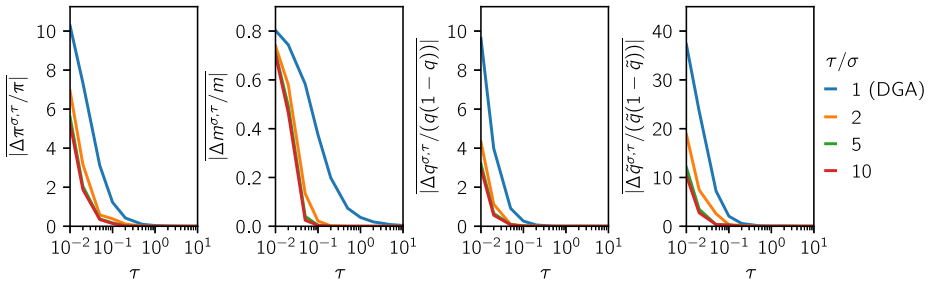

FIG. 6. Effect of the lag time and number of memory terms on the mean of the absolute value of the relative error for each dynamical statistic for the two-dimensional triple well.

simulations.[37,38] We recently performed both short and long simulations and used the resulting data to estimate statistics with MSMs and neural networks.[31]

Each amino acid can isomerize between left-handed (l) and right-handed (r) states, which are defined by the ϕ and ψ dihedral angles. Similarly to Ref. 31, we take an amino acid to be in the l state if its dihedral angle values are within a circle of radius 25° centered at $(41°, 47°)$, that is, $(\phi - 41°)^2 + (\psi - 47°)^2 \leq (25°)^2$. Amino acids are classified as being in the r state using the same radius but centered, instead, at $(-41°, -47°)$. States A and B are defined by the amino acids at sequence positions 3–7 being all l or all r, respectively. We show such configurations in Fig. 7. We do not use the two residues on each end of AIB_9 in defining the states as these are typically more flexible.[38]

The datasets that we use are from Ref. 31. For that work, we ran simulations in OpenMM 7.7.0[39] using the Langevin integrator at 300 K with a friction coefficient of 1 ps^{-1} and a time step of 4 fs. We used AIB parameters from Forcefield NCAA[40] with a hydrogen mass repartitioning scheme[41] and the GBNeck2 implicit-solvent model.[42] The short-trajectory dataset that we analyze consists of 10 trajectories of duration 20 ns from each of the 691 starting configurations in Ref. 38. The dataset thus contains 6910 trajectories, corresponding to a total sampling time of 138.2 μs. As a reference for comparison, we use 20 simulations of 15 μs with the same simulation parameters, corresponding to a total sampling time of 300 μs.

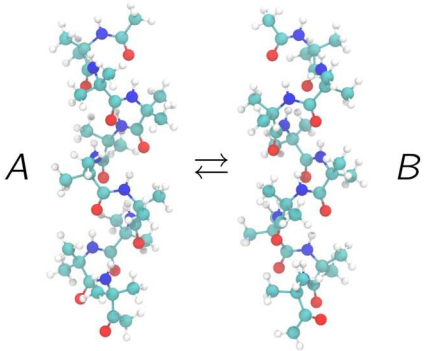

FIG. 7. A and B configurations of AIB_9. Carbon, nitrogen, oxygen, and hydrogen atoms are colored cyan, blue, red, and white, respectively.

In addition, we augment the reference dataset with its reflection (since AIB_9 is achiral).

We visualize the results on the CV space,

$$\chi_1 = \gamma_3 + \gamma_4 + \gamma_5 + \gamma_6 + \gamma_7, \tag{54a}$$

$$\chi_2 = \gamma_3 + \gamma_4 - \gamma_6 - \gamma_7, \tag{54b}$$

where $\gamma_i = -0.8(\sin\phi_i + \sin\psi_i)$. The l state has $\gamma \approx -1$ and the r state has $\gamma \approx 1$, although these values also contain states other than l and r.

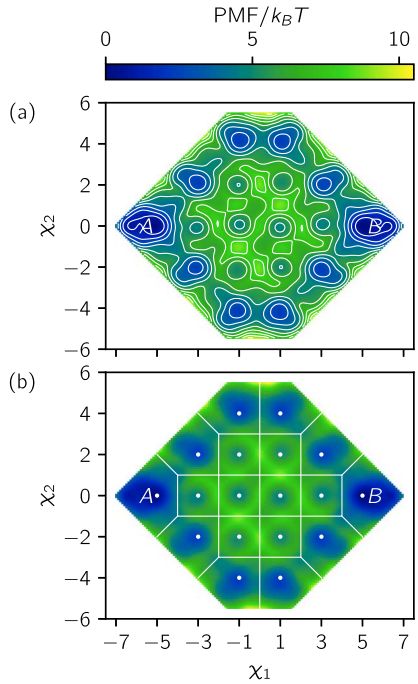

FIG. 8. (a) Potential of the mean force of AIB_9 in the (χ_1, χ_2) CV space with two metastable states A and B labeled. The contours are drawn every k_BT. (b) Location and coordinates of the centers (dots) of the 18 states in the CV space. Note that multiple intermediates may overlap in the same location in the CV space (30 different intermediates appear as 16 states). The edges of the basis functions are indicated by lines.

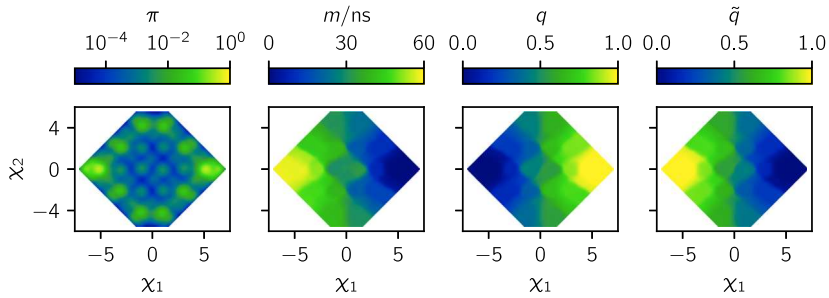

FIG. 9. Reference statistics for AlB_9 calculated from the long trajectories. From left to right, the stationary distribution (π), the MFPT to the all-right state B (m), the forward committor for the transition from the all-left state A to the all-right state B (q), and the backward committor for the same transition ($\tilde{q}$).

We choose these CVs because they can distinguish the ten most populated states and have physical meaning: χ_1 measures the chirality of a configuration, and χ_2 measures the difference in chirality between the two halves of the molecule. These CVs are similar to the first two principal components of the dihedral angle dynamics, which were used in Refs. 31, 38, and 43. We project statistics onto the CV space using a kernel density estimate with a Gaussian kernel with a scale parameter of 0.2.

We compute the potential of mean force (PMF) on the (χ_1, χ_2) space from a histogram of CV pairs sampled in the reference dataset [Fig. 8(a)]. This shows the metastable A and B states on the left and right sides, respectively, as well as the intermediate states connecting them. In Fig. 9, we show the reference statistics for the stationary distribution, the MFPT to the B state, and the forward and backward committors from the A state to the B state.

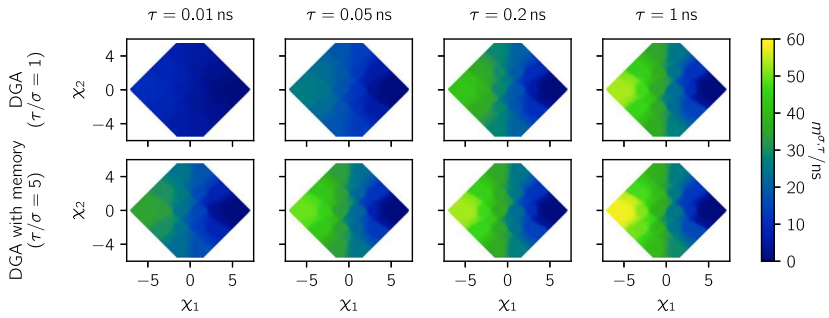

FIG. 10. Mean first passage time to the all-right B state for AlB_9 estimated using DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 5$), at different lag times τ. DGA with memory requires shorter lag times to achieve comparable results to DGA.

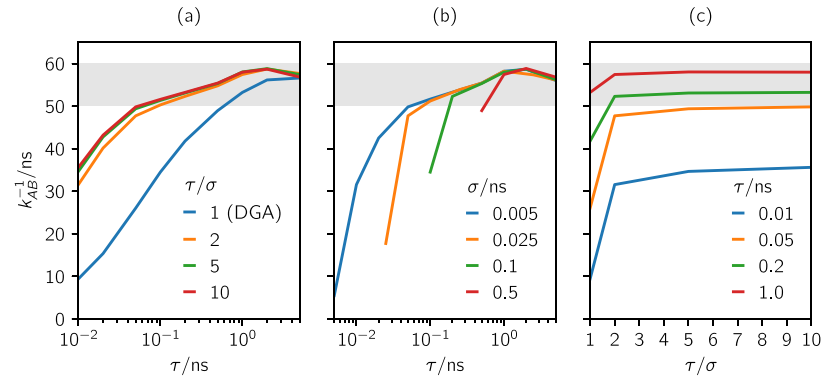

FIG. 11. Dependence of the estimated AlB_9 inverse rate constant on (a) τ with fixed τ/σ, (b) τ with fixed σ, and (c) τ/σ with fixed τ. The true inverse rate constant is between 50 and 60 ns, which is indicated by the gray region.

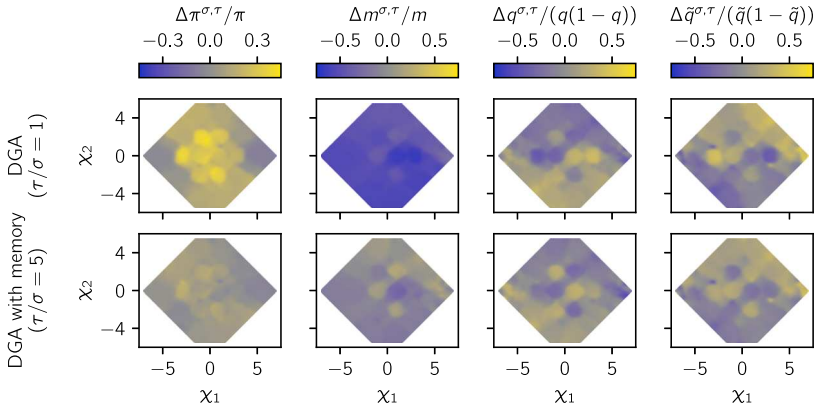

FIG. 12. Relative errors in the DGA $(\tau/\sigma = 1)$ and DGA with memory $(\tau/\sigma = 5)$ estimates of the AIB_9 dynamical statistics at lag time $\tau = 0.05$ ns.

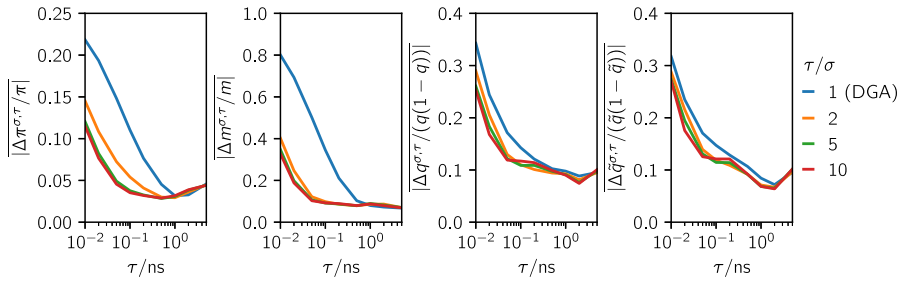

FIG. 13. Effect of the lag time and number of memory terms on the mean of the absolute value of the relative error for each dynamical statistic for AIB_9.

For the DGA calculations, we use a basis of 18 indicator functions on the (χ_1,χ_2) space as shown in Fig. 8(b). These indicator functions are Voronoi cells with centers at points (χ_1,χ_2) with each $\gamma_i = \pm 1$. We compute statistics using the same procedure as in Sec. IV A, where we first estimate the stationary distribution and then use it for other statistics. However, here, we use short-trajectory data. We use a rolling window, so all choices of σ and τ use the same amount of data.

In Fig. 10, we look at the MFPT, which, as for the two-dimensional triple well, most clearly shows the improvement due to memory. At $\tau = 0.05$ ns, DGA with memory $(\tau/\sigma = 5)$ is quite similar to the reference (Fig. 9). Meanwhile, DGA $(\tau/\sigma = 1)$ drastically underestimates the MFPT in all of the CV space. The DGA-with-memory results at $\tau = 0.2$ ns are comparable to the DGA results at $\tau = 1$ ns. Looking at the inverse rate constant from A to B (Fig. 11), which we estimate to be between 50 and 60 ns from the reference trajectories, DGA with memory enters the correct range at $\tau \approx 0.05$ ns, whereas DGA requires an order of magnitude more, $\tau \approx 0.5$ ns.

We calculate errors by first projecting onto the (χ_1,χ_2) space and then using the relative error formulas as described in Sec. IV A. We restrict the points used to calculate the mean absolute value of the relative absolute error to those with $|\chi_1| + |\chi_2| \le 7$ and $|\chi_2| \le 5.5$ due to a lack of samples outside of this region. The committors show limited improvement with memory (as shown in Figs. 12 and 13), so we focus on the stationary distribution and the MFPT. In Fig. 12, we show the results of DGA $(\tau/\sigma = 1)$ and DGA with memory $(\tau/\sigma = 5)$ at $\tau = 0.05$ ns. Both algorithms accurately estimate the stationary probabilities of states A and B even at this short lag time, but DGA overestimates the weights of intermediates, especially the minor intermediates in the center. DGA drastically underestimates the MFPT to the B state in all regions. Figure 13 shows that, for the stationary distribution and MFPT, DGA with memory requires an of order magnitude shorter lag time to achieve a comparable error to DGA. Note that the error fails to reach zero and even increases at longer lag times for π, q, and $\tilde{q}$; this is due to an increase in the statistical error at longer lag times in both the DGA estimates and the reference statistics. Overall, the relative performances of DGA and DGA with memory for AIB_9 are qualitatively similar to those for the two-dimensional triple well, despite the complications that we compute the statistics for AIB_9 from limited simulation data and quantify the error following projection to the (χ_1,χ_2) space.

V. CONCLUSIONS

In this paper, we incorporate memory to improve the accuracy of statistics estimated by the dynamical Galerkin approximation. Our work builds on numerical approaches for solving the orthogonal dynamics equation of the Mori–Zwanzig formalism[32] and quasi-MSMs[27–29] and shows that the basic concept behind these approaches can be generalized and used to compute much more

than relaxation time scales. A key feature of quasi-MSMs and the present approach is that the memory corrections are obtained by directly inverting a GME so that it is not necessary to assume a particular functional form for the memory. Quasi-MSMs and DGA with memory also contrast with traditional applications of the Mori–Zwanzig formalism in that the models are high-dimensional. As a result, the memory decays rapidly, and we find that even a few correction terms are sufficient to achieve high accuracy (with most of the correction coming from the first term beyond the Markov model). In our numerical experiments, the memory correction decreases the data requirement by an order of magnitude.

Our approach has numerical advantages over quasi-MSMs and other methods that predict long-time statistics by integrating the GME. Integrating the GME requires estimating the full coarse-grained propagator and then using it to extrapolate in time. Solving statistic-specific equations, as we do here, allows better control of errors. In particular, because boundary conditions and other constraints are satisfied regardless of the choice of τ and σ, we can increase τ to decrease the error while keeping τ/σ small. A separate strategy that can be applied to improve the numerical performance further is to integrate over lag times.[28,44]

While DGA with memory can decrease the sensitivity of estimates to the choice of basis, a choice must still be made. To address this issue, we recently introduced an inexact subspace iteration to learn a basis represented by neural networks.[31] In Ref. 31, we apply the memory correction after we learn the basis, but one could apply it after each subspace iteration to accelerate convergence. Even when the representation of a dynamical statistic does not have the form in (8) (e.g., Ref. 45, or the Richardson iterate in Ref. 31), one could define a nonlinear projection operator and, in turn, compute a memory correction to compensate for deficiencies of the input features.

It would be interesting to compare the Mori–Zwanzig approach to memory taken here to alternative means of treating memory, such as delay embedding,[2,4–7] with regard to data requirements and robustness to hyperparameter choices for high-dimensional models like those considered here.

ACKNOWLEDGMENTS

We are grateful to Xuhui Huang for making us aware of Refs. 27–29. This work was supported by the National Institutes of Health, Award No. R35 GM136381, and the National Science Foundation, Award No. DMS-2054306. S.C.G. acknowledges the support provided by the National Science Foundation Graduate Research Fellowship under Grant No. 2140001. This work was completed in part with resources provided by the University of Chicago Research Computing Center, and we are grateful for their assistance with the calculations. "Beagle-3: A Shared GPU Cluster for Biomolecular Sciences" is supported by the National Institutes of Health (NIH) under the High-End Instrumentation (HEI) grant program, Award No. 1S10OD028655-0.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Chatipat Lorpaiboon: Conceptualization (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Software (lead); Writing – original draft (equal); Writing – review & editing (equal). **Spencer C. Guo**: Investigation (supporting); Methodology (supporting); Writing – original draft (supporting); Writing – review & editing (supporting). **John Strahan**: Methodology (supporting). **Jonathan Weare**: Conceptualization (supporting); Supervision (equal); Writing – review & editing (supporting). **Aaron R. Dinner**: Conceptualization (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The code for our methods and examples is available at https://github.com/dinner-group/dgamem.

APPENDIX A: GRAPHICAL DEPICTION OF THE EFFECT OF MEMORY

We compare DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 2$) graphically in Fig. 14. The projected dynamics move the system horizontally, while the orthogonal dynamics move it vertically. We show early iterations in Fig. 14(a) and the fixed point in Fig. 14(b). The iterations alternate between using length-τ trajectories to propagate the projected function $\hat{u}_t^\tau$ to $u_{t+\tau}^\tau$ (colored curves) and projecting back onto the subspace spanned by the basis (dashed lines).

In the case of DGA [Fig. 14(a), $\tau/\sigma = 1$], the iteration (12) rapidly diverges from the exact iteration in (15), which is plotted in gray. The fixed point (12) occurs when propagating $\hat{u}^\tau$ for lag time τ to u^τ such that projecting returns the same $\hat{u}^\tau$ [Fig. 14(b)]. As drawn, fast variables propagate $\hat{u}^\tau$ up and to the right and slow variables ultimately return the system so as to be directly above $\hat{u}^\tau$, yielding the blue curve. In the absence of error due to sampling trajectories, $\hat{u}^\tau$ deviates from $\hat{u}$ because τ is insufficiently long for u^τ to reach u.

Comparing the iteration without memory ($\tau/\sigma = 1$) to the iteration with memory ($\tau/\sigma = 2$), we see that the latter approximates the exact iteration with significantly greater accuracy. Each iteration with memory shown in the figure consists of four steps: (1) propagating $\hat{u}_t^{\sigma,\tau}$ by time σ (which initially moves the system up and to the right as drawn), (2) adjusting the projected variable with the orthogonal projection so that the projection is $\hat{u}_\sigma^{\sigma,\tau}$ (which moves the system to the left horizontally), (3) propagating by another time σ to $u_{t+2\sigma}^{\sigma,\tau}$ (which again moves the system up and to the right), and (4) projecting to $\hat{u}_{t+2\sigma}^{\sigma,\tau}$ (which moves the system vertically). More generally, whereas each iteration without memory propagates the dynamics for time τ, each iteration with memory propagates the dynamics for time σ, in which the projected variable is adjusted $(\tau/\sigma) - 1$ times.

As for the case of DGA, the fixed point is reached when the final projection in each step of the iteration returns the system to $\hat{u}^{\sigma,\tau}$ [Fig. 14(b), bottom]. In DGA, the fast modes propagate $\hat{u}^\tau$ toward the upper right; τ must be long enough for the slow mode to correct this additional rightward shift. The impact of the memory correction is to eliminate this shift so that if τ is sufficiently large to relax the fast modes that affect the projected variable, the deviation from u

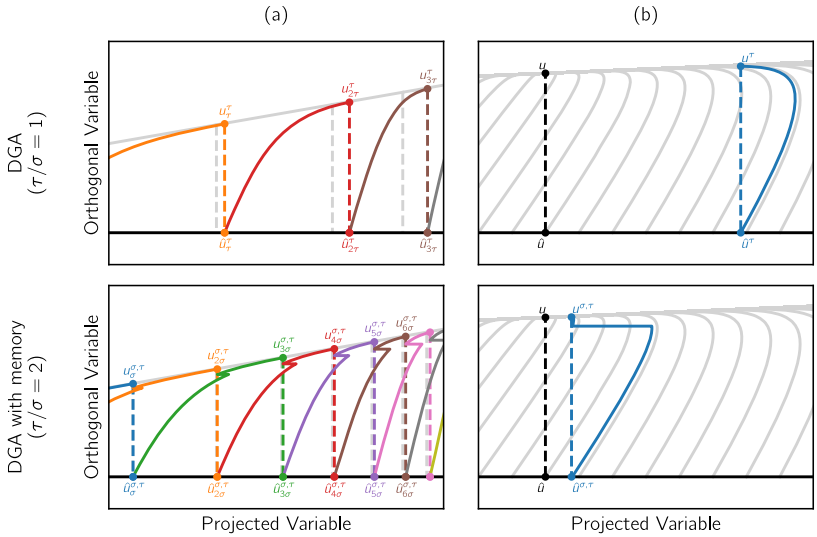

FIG. 14. Schematic of the iterations and their solutions. (a) Inexact Richardson iteration corresponding to DGA [$\tau/\sigma = 1$, (12)] and DGA with memory [$\tau/\sigma = 2$, (21)]. The exact Richardson iteration in (15) is plotted is gray. (b) Solutions to DGA ($\tau/\sigma = 1$) and DGA with memory ($\tau/\sigma = 2$). To guide the eye, exact Richardson iterations starting at different projections are plotted in gray.

depends only on the interaction between the slow modes and the orthogonal variables.

APPENDIX B: COMPUTING FINITE-TIME OPERATORS USING THE INFINITESIMAL GENERATOR

Here, we express the finite-time operators (e.g., $\mathcal{T}^t$ and $\mathcal{S}_B^t$) in terms of the infinitesimal generator,

$$\mathcal{L} = \lim_{\epsilon \to 0^+} \frac{\mathcal{T}^\epsilon - \mathcal{I}}{\epsilon}. \tag{B1}$$

For a continuous-time Markov jump process on a finite discrete state space, $\mathcal{L}$ is a matrix [e.g., (52)] and functions on the grid are vectors, so

$$\mathcal{L}f(x) = \sum_{x'} \mathcal{L}(x,x')f(x'), \tag{B2}$$

and we can evaluate expressions in terms of $\mathcal{L}$ directly.

1. Transition operator

The transition operator can be expressed as

$$\mathcal{T}^t = (\mathcal{T}^\epsilon)^{t/\epsilon} = \lim_{\epsilon \to 0^+} (\mathcal{I} + \epsilon\mathcal{L})^{t/\epsilon} = e^{t\mathcal{L}}. \tag{B3}$$

Similarly,

$$(\mathcal{T}^t)^\dagger(x,x') = e^{t\mathcal{L}^\dagger}(x,x') \tag{B4}$$

and

$$\mathcal{L}^\dagger(x,x') = \mu(x')\mathcal{L}(x',x)/\mu(x) \tag{B5}$$

from the definition of the adjoint.

2. Stopped transition operator

Here, we express the stopped transition operator in terms of the infinitesimal generator. To this end, we observe that

$$f(X_{t\wedge T_B}) = \lim_{\epsilon \to 0^+} \left(f(X_0) + \sum_{n=0}^{(t/\epsilon)-1} \left[\prod_{n'=0}^{n} \mathbf{1}_{B^C}(X_{n'\epsilon}) \right] \times (f(X_{(n+1)\epsilon}) - f(X_{n\epsilon})) \right). \tag{B6}$$

That is, at time $n\epsilon$, we propagate the process forward-in-time only if $n\epsilon < T_B$ (the product above is 1 only if $X_0, \ldots, X_{n\epsilon} \in B^C$ and 0 otherwise). Therefore, we can express (6) as

$$\mathcal{S}_B^t f(x) = \mathbf{E}[f(X_{t\wedge T_B}) \mid X_0 = x] \tag{B7}$$

$$= \lim_{\epsilon \to 0^+} \left(f(x) + \sum_{n=0}^{(t/\epsilon)-1} (\mathcal{D}(\mathbf{1}_{B^C})\mathcal{T}^\epsilon)^n \mathcal{D}(\mathbf{1}_{B^C}) \times (\mathcal{T}^\epsilon - \mathcal{I})f(x) \right) \tag{B8}$$

$$= f(x) + \int_0^t e^{t'\mathcal{D}(\mathbf{1}_{B^C})\mathcal{L}} \mathcal{D}(\mathbf{1}_{B^C})\mathcal{L}f(x)\,dt' \tag{B9}$$

$$= e^{t\mathcal{D}(\mathbf{1}_{B^C})\mathcal{L}} f(x). \tag{B10}$$

We substitute (B6) into (B7) and express the resulting conditional expectation in terms of $\mathcal{T}^\epsilon(x,x') = \mathbf{P}[X_{t+\epsilon} = x' \mid X_t = x]$ to obtain (B8). Taking the limit $\epsilon \to 0^+$ yields (B9), and evaluating the integral yields (B10). We define $\mathcal{D}(g)$ to be a diagonal matrix with entries $\mathcal{D}(g)(x,x) = g(x)$; the matrix $\mathcal{D}(\mathbf{1}_{B^C})\mathcal{L}$ has entries

$$(\mathcal{D}(\mathbf{1}_{B^C})\mathcal{L})(x,x') = \mathbf{1}_{B^C}(x)\mathcal{L}(x,x'). \tag{B11}$$

We evaluate the integral $\int_0^t \mathcal{S}_B^{t'} \mathbf{1}_{B^C}\, dt' = \int_0^t e^{t' \mathcal{D}(\mathbf{1}_{B^C})\mathcal{L}} \mathbf{1}_{B^C}\, dt'$ in (36c) using the identity

$$\begin{aligned}
&\exp\left(t\begin{bmatrix} \mathcal{D}(\mathbf{1}_{B^C})\mathcal{L} & \mathbf{1}_{B^C} \\ 0 & 0 \end{bmatrix}\right) \\
&= \lim_{\epsilon\to 0^+}\left(\begin{bmatrix} \mathcal{I} & 0 \\ 0 & \mathcal{I} \end{bmatrix} + \epsilon\begin{bmatrix} \mathcal{D}(\mathbf{1}_{B^C})\mathcal{L} & \mathbf{1}_{B^C} \\ 0 & 0 \end{bmatrix}\right)^{t/\epsilon} \\
&= \lim_{\epsilon\to 0^+}\begin{bmatrix} (\mathcal{I}+\epsilon\mathcal{D}(\mathbf{1}_{B^C})\mathcal{L})^{t/\epsilon} & \epsilon\sum_{n=0}^{(t/\epsilon)-1}(\mathcal{I}+\epsilon\mathcal{D}(\mathbf{1}_{B^C})\mathcal{L})^n \mathbf{1}_{B^C} \\ 0 & \mathcal{I} \end{bmatrix} \\
&= \begin{bmatrix} e^{t\mathcal{D}(\mathbf{1}_{B^C})\mathcal{L}} & \int_0^t e^{t'\mathcal{D}(\mathbf{1}_{B^C})\mathcal{L}} \mathbf{1}_{B^C}\, dt' \\ 0 & 1 \end{bmatrix}. \qquad \text{(B12)}
\end{aligned}$$

That is, we compute the exponential and then take only the upper right element of the result.

3. Stopped transition operator for the time-reversed process

Similarly to (B6)–(B10), we use the identity

$$f(X_{-(t\wedge\tilde{T}_{A\cup B})}) = \lim_{\epsilon\to 0^+}\left(f(X_0) + \sum_{n=0}^{(t/\epsilon)-1}\left[\prod_{n'=0}^{n}\mathbf{1}_{(A\cup B)^C}(X_{-n'\epsilon})\right] \times (f(X_{-(n+1)\epsilon}) - f(X_{-n\epsilon}))\right) \qquad \text{(B13)}$$

to express (46) as

$$\langle g, \mathcal{S}_{A\cup B}^{-t} f\rangle = \mathbf{E}\left[g(X_0) f(X_{-(t\wedge\tilde{T}_{A\cup B})}) \frac{w(X_{-\tau})}{w(X_0)}\middle| X_{-\tau}\sim\mu\right] \qquad \text{(B14)}$$

$$\begin{aligned}
&= \lim_{\epsilon\to 0^+}\sum_x \frac{g(x)\mu(x)}{w(x)}\Bigg(f(x)(\mathcal{T}^\tau)^\dagger w(x) + \sum_{n=0}^{(t/\epsilon)-1} \\
&\quad\times\Big[(\mathcal{D}(\mathbf{1}_{(A\cup B)^C})(\mathcal{T}^\epsilon)^\dagger)^n \mathcal{D}(\mathbf{1}_{(A\cup B)^C}) \\
&\quad\times((\mathcal{T}^\epsilon)^\dagger\mathcal{D}(f) - \mathcal{D}(f)(\mathcal{T}^\epsilon)^\dagger)(\mathcal{T}^{\tau-(n+1)\epsilon})^\dagger w(x)\Big]\Bigg) \qquad \text{(B15)}
\end{aligned}$$

$$\begin{aligned}
&= \sum_x \frac{g(x)\mu(x)}{w(x)}\Bigg(f(x)e^{\tau\mathcal{L}^\dagger} w(x) + \int_0^t e^{t'\mathcal{D}(\mathbf{1}_{(A\cup B)^C})\mathcal{L}^\dagger} \\
&\quad\times \mathcal{D}(\mathbf{1}_{(A\cup B)^C})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger)e^{(\tau-t')\mathcal{L}^\dagger} w(x)\, dt'\Bigg). \qquad \text{(B16)}
\end{aligned}$$

We substitute (B13) into (B14) and expand the resulting expectation in terms of $(\mathcal{T}^\epsilon)^\dagger(x,x') = \mathbf{P}[X_{t+\epsilon} = x \mid X_t = x']\mu(x')/\mu(x)$ to obtain (B15). Taking the limit $\epsilon \to 0^+$ yields (B16), where the entries of matrices $\mathcal{D}(\mathbf{1}_{(A\cup B)^C})\mathcal{L}^\dagger$ and $\mathcal{D}(\mathbf{1}_{(A\cup B)^C})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger)$ are

$$(\mathcal{D}(\mathbf{1}_{(A\cup B)^C})\mathcal{L}^\dagger)(x,x') = \mathbf{1}_{(A\cup B)^C}(x)\mathcal{L}^\dagger(x,x'), \qquad \text{(B17)}$$

$$\begin{aligned}
&(\mathcal{D}(\mathbf{1}_{(A\cup B)^C})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger))(x,x') \\
&\quad= \mathbf{1}_{(A\cup B)^C}(x)\mathcal{L}^\dagger(x,x')(f(x') - f(x)). \qquad \text{(B18)}
\end{aligned}$$

Likewise,

$$\langle g, (\mathcal{S}_{A\cup B}^{-t} - \mathcal{I})f\rangle = \mathbf{E}\left[g(X_0)(f(X_{-(t\wedge\tilde{T}_{A\cup B})}) - f(X_0))\frac{w(X_{-\tau})}{w(X_0)}\middle| X_{-\tau}\sim\mu\right] \qquad \text{(B19)}$$

$$= \sum_x \frac{g(x)\mu(x)}{w(x)}\int_0^t e^{t'\mathcal{D}(\mathbf{1}_{(A\cup B)^C})\mathcal{L}^\dagger}\mathcal{D}(\mathbf{1}_{(A\cup B)^C})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger)e^{(\tau-t')\mathcal{L}^\dagger} w(x)\, dt'. \qquad \text{(B20)}$$

The integrals in (B16) and (B20) can be evaluated similarly to (B12),

$$\exp\left(t\begin{bmatrix} \mathcal{D}(\mathbf{1}_{(A\cup B)^C})\mathcal{L}^\dagger & \mathcal{D}(\mathbf{1}_{(A\cup B)^C})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger) \\ 0 & \mathcal{L}^\dagger \end{bmatrix}\right) \qquad \text{(B21)}$$

$$= \begin{bmatrix} e^{t\mathcal{D}(\mathbf{1}_{(A\cup B)^C})\mathcal{L}^\dagger} & \int_0^t e^{t'\mathcal{D}(\mathbf{1}_{(A\cup B)^C})\mathcal{L}^\dagger}\mathcal{D}(\mathbf{1}_{(A\cup B)^C})(\mathcal{L}^\dagger\mathcal{D}(f) - \mathcal{D}(f)\mathcal{L}^\dagger)e^{(t-t')\mathcal{L}^\dagger}\, dt' \\ 0 & e^{\mathcal{L}^\dagger t} \end{bmatrix}. \qquad \text{(B22)}$$

The expressions for $\mathcal{S}_{A\cup B}^{-t}$ are more complicated than those for $\mathcal{S}_{A\cup B}^{t}$ because we want (B16) and (B20) to be equivalent to (B14) and (B19), respectively, when w is approximated. If we evaluate them without approximating w, $\mathcal{L}^\dagger w = 0$ and $e^{\mathcal{L}^\dagger t} w = w$; therefore, $\langle g, \mathcal{S}_{A\cup B}^{-t} f\rangle = \sum_{x,x'}(g(x)\mu(x)/w(x))e^{t\mathcal{D}(\mathbf{1}_{(A\cup B)^C})\mathcal{L}^\dagger}(x,x')f(x')w(x')$ and $\langle g, (\mathcal{S}_{A\cup B}^{-t} - \mathcal{I})f\rangle = \sum_{x,x'}(g(x)\mu(x)/w(x))(e^{t\mathcal{D}(\mathbf{1}_{(A\cup B)^C})\mathcal{L}^\dagger} - \mathcal{I})(x,x')f(x')w(x')$.

REFERENCES

[1] N. Guttenberg, J. F. Dama, M. G. Saunders, G. A. Voth, J. Weare, and A. R. Dinner, "Minimizing memory as an objective for coarse-graining," J. Chem. Phys. **138**, 094111 (2013).

[2] W. Liebert and H. G. Schuster, "Proper choice of the time delay for the analysis of chaotic time series," Phys. Lett. A **142**, 107–111 (1989).

[3] H. Arbabi and I. Mezić, "Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator," SIAM J. Appl. Dyn. Syst. **16**, 2096–2126 (2017).

[4] S. Das and D. Giannakis, "Delay-coordinate maps and the spectra of Koopman operators," J. Stat. Phys. **175**, 1107–1145 (2019).

[5] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz, "Time-delay observables for Koopman: Theory and applications," SIAM J. Appl. Dyn. Syst. **19**, 886–917 (2020).

[6] E. H. Thiede, D. Giannakis, A. R. Dinner, and J. Weare, "Galerkin approximation of dynamical quantities using trajectory data," J. Chem. Phys. **150**, 244111 (2019).

[7] J. Strahan, A. Antoszewski, C. Lorpaiboon, B. P. Vani, J. Weare, and A. R. Dinner, "Long-time-scale predictions from short-trajectory data: A benchmark analysis of the trp-cage miniprotein," J. Chem. Theory Comput. **17**, 2948–2963 (2021).

[8] H. Mori, "Transport, collective motion, and Brownian motion," Prog. Theor. Phys. **33**, 423–455 (1965).

[9] R. Zwanzig, "From classical dynamics to continuous time random walks," J. Stat. Phys. **30**, 255–262 (1983).

[10] M. Berkowitz, J. D. Morgan, D. J. Kouri, and J. A. McCammon, "Memory kernels from molecular dynamics," J. Chem. Phys. **75**, 2462–2463 (1981).

[11] M. Berkowitz, J. D. Morgan, and J. A. McCammon, "Generalized Langevin dynamics simulations with arbitrary time-dependent memory kernels," J. Chem. Phys. **78**, 3256–3261 (1983).

[12] A. Perico, R. Pratolongo, K. F. Freed, R. W. Pastor, and A. Szabo, "Positional time correlation function for one-dimensional systems with barrier crossing: Memory function corrections to the optimized Rouse–Zimm approximation," J. Chem. Phys. **98**, 564–573 (1993).

[13] K. S. Kostov and K. F. Freed, "Mode coupling theory for calculating the memory functions of flexible chain molecules: Influence on the long time dynamics of oligoglycines," J. Chem. Phys. **106**, 771–783 (1997).

[14] K. S. Kostov and K. F. Freed, "Long-time dynamics of met-enkephalin: Comparison of theory with Brownian dynamics simulations," Biophys. J. **76**, 149–163 (1999).

[15] A. J. Chorin, O. H. Hald, and R. Kupferman, "Optimal prediction with memory," Physica D **166**, 239–257 (2002).

[16] H. Lei, N. A. Baker, and X. Li, "Data-driven parameterization of the generalized Langevin equation," Proc. Natl. Acad. Sci. U. S. A. **113**, 14183–14188 (2016).

[17] F. Grogan, H. Lei, X. Li, and N. A. Baker, "Data-driven molecular modeling with the generalized Langevin equation," J. Comput. Phys. **418**, 109633 (2020).

[18] B. J. Berne, J. P. Boon, and S. A. Rice, "On the calculation of autocorrelation functions of dynamical variables," J. Chem. Phys. **45**, 1086–1096 (1966).

[19] J.-P. Boon and S. A. Rice, "Memory effects and the autocorrelation function of a dynamical variable," J. Chem. Phys. **47**, 2480–2490 (1967).

[20] C. Ayaz, L. Tepper, F. N. Brünig, J. Kappler, J. O. Daldrop, and R. R. Netz, "Non-Markovian modeling of protein folding," Proc. Natl. Acad. Sci. U. S. A. **118**, e2023856118 (2021).

[21] O. F. Lange and H. Grubmüller, "Collective Langevin dynamics of conformational motions in proteins," J. Chem. Phys. **124**, 214903 (2006).

[22] H. S. Lee, S.-H. Ahn, and E. F. Darve, "The multi-dimensional generalized Langevin equation for conformational motion of proteins," J. Chem. Phys. **150**, 174113 (2019).

[23] Y. T. Lin, Y. Tian, D. Livescu, and M. Anghel, "Data-driven learning for the Mori–Zwanzig formalism: A Generalization of the Koopman learning framework," SIAM J. Appl. Dyn. Syst. **20**, 2558–2601 (2021).

[24] H. Vroylandt, L. Goudenège, P. Monmarché, F. Pietrucci, and B. Rotenberg, "Likelihood-based non-Markovian models from molecular dynamics," Proc. Natl. Acad. Sci. U. S. A. **119**, e2117586119 (2022).

[25] D. Aristoff, M. Johnson, and D. Perez, "Arbitrarily accurate, nonparametric coarse graining with Markov renewal processes and the Mori–Zwanzig formulation," AIP Adv. **13**, 095131 (2023).

[26] Y. T. Lin, Y. Tian, D. Perez, and D. Livescu, "Regression-based projection for learning Mori–Zwanzig operators," SIAM J. Appl. Dyn. Syst. **22**, 2890–2926 (2023).

[27] S. Cao, A. Montoya-Castillo, W. Wang, T. E. Markland, and X. Huang, "On the advantages of exploiting memory in Markov state models for biomolecular dynamics," J. Chem. Phys. **153**, 014105 (2020).

[28] S. Cao, Y. Qiu, M. L. Kalin, and X. Huang, "Integrative generalized master equation: A method to study long-timescale biomolecular dynamics via the integrals of memory kernels," J. Chem. Phys. **159**, 134106 (2023).

[29] A. J. Dominic III, T. Sayer, S. Cao, T. E. Markland, X. Huang, and A. Montoya-Castillo, "Building insightful, memory-enriched models to capture long-time biochemical processes from short-time simulations," Proc. Natl. Acad. Sci. U. S. A. **120**, e2221048120 (2023).

[30] F. Noé, C. Schütte, E. Vanden-Eijnden, L. Reich, and T. R. Weikl, "Constructing the equilibrium ensemble of folding pathways from short off-equilibrium simulations," Proc. Natl. Acad. Sci. U. S. A. **106**, 19011–19016 (2009).

[31] J. Strahan, S. C. Guo, C. Lorpaiboon, A. R. Dinner, and J. Weare, "Inexact iterative numerical linear algebra for neural network-based spectral estimation and rare-event prediction," J. Chem. Phys. **159**, 014110 (2023).

[32] E. Darve, J. Solomon, and A. Kia, "Computing generalized Langevin equations and generalized Fokker–Planck equations," Proc. Natl. Acad. Sci. U. S. A. **106**, 10884–10889 (2009).

[33] J. Cerrillo and J. Cao, "Non-Markovian dynamical maps: Numerical processing of open quantum trajectories," Phys. Rev. Lett. **112**, 110401 (2014).

[34] H. Wu, F. Nüske, F. Paul, S. Klus, P. Koltai, and F. Noé, "Variational Koopman models: Slow collective variables and molecular kinetics from short off-equilibrium simulations," J. Chem. Phys. **146**, 154104 (2017).

[35] P. Metzner, C. Schütte, and E. Vanden-Eijnden, "Illustration of transition path theory on a collection of simple examples," J. Chem. Phys. **125**, 084110 (2006).

[36] P. Reimann, G. J. Schmid, and P. Hänggi, "Universal equivalence of mean first-passage time and Kramers rate," Phys. Rev. E **60**, R1–R4 (1999).

[37] S. Buchenberg, N. Schaudinnus, and G. Stock, "Hierarchical biomolecular dynamics: Picosecond hydrogen bonding regulates microsecond conformational transitions," J. Chem. Theory Comput. **11**, 1330–1336 (2015).

[38] A. Perez, F. Sittel, G. Stock, and K. Dill, "MELD-path efficiently computes conformational transitions, including multiple and diverse paths," J. Chem. Theory Comput. **14**, 2109–2116 (2018).

[39] P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, and V. S. Pande, "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics," PLOS Comput. Biol. **13**, e1005659 (2017).

[40] G. A. Khoury, J. Smadbeck, P. Tamamis, A. C. Vandris, C. A. Kieslich, and C. A. Floudas, "Forcefield_NCAA: *Ab initio* charge parameters to aid in the discovery and design of therapeutic proteins and peptides with unnatural amino acids and their application to complement inhibitors of the compstatin family," ACS Synth. Biol. **3**, 855–869 (2014).

[41] C. W. Hopkins, S. Le Grand, R. C. Walker, and A. E. Roitberg, "Long-time-step molecular dynamics through hydrogen mass repartitioning," J. Chem. Theory Comput. **11**, 1864–1874 (2015).

[42] H. Nguyen, D. R. Roe, and C. Simmerling, "Improved generalized Born solvent model parameters for protein simulations," J. Chem. Theory Comput. **9**, 2020–2034 (2013).

[43] F. Sittel, T. Filk, and G. Stock, "Principal component analysis on a torus: Theory and application to protein dynamics," J. Chem. Phys. **147**, 244101 (2017).

[44] C. Lorpaiboon, E. H. Thiede, R. J. Webber, J. Weare, and A. R. Dinner, "Integrated variational approach to conformational dynamics: A robust strategy for identifying eigenfunctions of dynamical operators," J. Phys. Chem. B **124**, 9354–9364 (2020).

[45] J. Strahan, J. Finkel, A. R. Dinner, and J. Weare, "Predicting rare events using neural networks and short-trajectory data," J. Comput. Phys. **488**, 112152 (2023).