

Computing generalized Langevin equations and generalized Fokker–Planck equations

Eric Darve^{a,b,1}, Jose Solomon^b, and Amirali Kia^b

^aInstitute for Computational and Mathematical Engineering; and ^bMechanical Engineering Department, Stanford University, Stanford CA 94305

Edited by Alexandre J. Chorin, University of California, Berkeley, CA, and approved April 10, 2009 (received for review March 10, 2009)

The Mori–Zwanzig formalism is an effective tool to derive differential equations describing the evolution of a small number of resolved variables. In this paper we present its application to the derivation of generalized Langevin equations and generalized non-Markovian Fokker–Planck equations. We show how long time scales rates and metastable basins can be extracted from these equations. Numerical algorithms are proposed to discretize these equations. An important aspect is the numerical solution of the orthogonal dynamics equation which is a partial differential equation in a high dimensional space. We propose efficient numerical methods to solve this orthogonal dynamics equation. In addition, we present a projection formalism of the Mori–Zwanzig type that is applicable to discrete maps. Numerical applications are presented from the field of Hamiltonian systems.

Mori–Zwanzig formalism | optimal prediction with memory | coarse-grained model | reduced order model | multiscale model

Many applications such as molecular dynamics lead to the solution of a system of ordinary differential equations,

$$\frac{d\mathbf{u}}{dt} = \mathbf{R}(\mathbf{u}), \quad \mathbf{u} \in \mathbb{R}^n, \quad [1]$$

involving a wide range of time scales. For example the time step in molecular dynamics simulations of proteins is 1 fs while typical events of interests are in the micro- or millisecond time scale. Carrying out these simulations using brute force techniques is impractical (e.g., by integrating the equations of motion); this is one of the main limiting factors towards greater predictability and applicability. This issue can be partially addressed by techniques that attempt to model this high-dimensional system by using a reduced set of resolved variables $\mathbf{A}(\mathbf{u})$, or observables. It is not possible to formulate exact equations for $d\mathbf{A}(t)/dt$ in closed form, that is in terms of \mathbf{A} only. Approximations are necessary to close the equations. An effective approach can be derived from the Mori–Zwanzig formalism (1–3), which assumes that there is a probability distribution $\mu(d\mathbf{u})$ conserved by the dynamics. This formalism leads to a decomposition of $d\mathbf{A}(t)/dt$ in three terms (4): a drift term that is a function of $\mathbf{A}(t)$, a memory term that depends on $\mathbf{A}(s)$ for $0 \leq s \leq t$, and a fluctuating term \mathbf{F}_t . One may then replace the fluctuating term by a stochastic process, for example white or colored noise (5, 6), to close the system of equations.

As examples of applications in biochemistry, one might be interested in modeling the position of an ion in a membrane channel, or the motion of the centers of mass of groups of atoms without resolving internal vibrations. It might also be desirable to model a large number of degrees of freedom which are computationally expensive to calculate; this is the case for example in implicit water models where water molecules are removed from the system and replaced by a stochastic model such as a Langevin model. Many other such examples can be found from the literature on multiscale modeling (5).

The same Mori–Zwanzig formalism can be used to derive a kind of generalized Fokker–Planck equation for the evolution of a probability density function $\phi^t(\mathbf{A})$. This equation, contrary to the Fokker–Planck equation for diffusive processes, contains a term function of ϕ^t and a non-Markovian term function of past values ϕ^s , $0 \leq s \leq t$. We will show how all the relevant time scales in the

system, e.g., reaction rates, and metastable basins can be extracted numerically from this equation.

One of the main numerical difficulties in these equations is that the fluctuating term \mathbf{F}_t and the memory kernels require in principle the solution of the so-called “orthogonal dynamics equation” (7) which is a partial differential equation with $n + 1$ variables (recall that $\mathbf{u} \in \mathbb{R}^n$). This is impractical in most real life applications where n can be in the range 10^4 – 10^6 . Many techniques have been developed to address this issue (see refs. 5–8 for example). We propose a new approach to solve this equation. This approach does not require a time scale separation, wherein the variable \mathbf{A} is assumed to be much slower than other time scales in the system, or an adiabatic or Markovian approximation. The method is numerically robust, e.g., it is not sensitive to small perturbations in the data (see for example ref. 9 which requires solving a Volterra integral equation of the first kind). The method has a low computational cost and can be carried out on desktop computers.

The paper is organized as follows. We first present the standard Mori–Zwanzig formalism. For any phase variable \mathbf{B} , this gives an equation for $e^{t\mathcal{L}}\mathbf{B}$ where $\mathcal{L} \stackrel{\text{def}}{=} \sum_{i=1}^n R_i \partial/\partial u_i$ is the Liouville operator. We also derive a new formulation applicable to a discrete map \mathcal{M} , in which we obtain equations for $\mathcal{M}^k\mathbf{B}$. This is followed by 2 important equations which can be derived from the Mori–Zwanzig formalism: the generalized Langevin equation (GLE) and generalized non-Markovian Fokker–Planck equation (GFPE). A numerical discretization of the GFPE based on a Galerkin scheme is then proposed along with an algorithm to calculate reaction rates and other time scales in the system. These equations rely on the solution of the orthogonal dynamic equations. An algorithm to carry out this calculation is presented. The paper ends with numerical results. The notation $\stackrel{\text{def}}{=}$ indicates a definition or an equality that cannot be derived from previous statements.

Mori–Zwanzig Projection

We consider the dynamical system given by Eq. 1 where $\mathbf{u} \in \Omega \subset \mathbb{R}^n$. In the context of molecular dynamics of proteins, the vector \mathbf{u} is the set (\mathbf{q}, \mathbf{p}) of atom coordinates and momenta. In many contexts it is desirable to model the dynamical system using only a subset of variables instead of the full set \mathbf{u} . This might be the case if one is trying to build a coarse grained model. These problems can be formulated abstractly in the following fashion. Let us denote $\mathbf{v}(\mathbf{u}_0, t)$ the solution of Eq. 1 at time t with initial conditions $\mathbf{u}(0) = \mathbf{u}_0$. A phase variable \mathbf{A} is an m -dimensional vector valued function defined on Ω , $\mathbf{A}(\mathbf{u})$. Associated with the Liouville operator \mathcal{L} , we define a time evolution operator $e^{t\mathcal{L}} : [e^{t\mathcal{L}}\mathbf{A}](\mathbf{u}) \stackrel{\text{def}}{=} \mathbf{A}(\mathbf{v}(\mathbf{u}, t))$.

The general model reduction problem or coarse graining problem can be formulated as follows: given a phase variable \mathbf{B} and $e^{s\mathcal{L}}\mathbf{A}$ ($0 \leq s \leq t$), is it possible to approximate $e^{t\mathcal{L}}\mathbf{B}$? We call

Author contributions: E.D. designed research; E.D., J.S., and A.K. performed research; E.D. analyzed data; and E.D., J.S., and A.K. wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

¹To whom correspondence should be addressed at: 496 Lomita Mall, Durand Building, Room 209, Stanford University, Stanford, CA 94305. E-mail: darve@stanford.edu.

this the closure problem. For example, \mathbf{B} could be $d\mathbf{A}/dt$. This problem is trivial if \mathcal{A} is a complete set of generalized coordinates, i.e., $m = n$. In most applications however, $m \ll n$, so that this (coarse-graining) procedure may provide significant computational speed-up.

The Mori–Zwanzig procedure is a very general and powerful formalism to help answer such problems. From now on, we assume that the initial conditions for \mathbf{u} are drawn from a probability distribution μ^0 . The probability distribution μ^t is defined by the condition

$$\forall \mathbf{B}, \int \mathbf{B}(\mathbf{u})\mu^t(d\mathbf{u}) \stackrel{\text{def}}{=} \int \mathbf{B}(\mathbf{v}(\mathbf{u}, t))\mu^0(d\mathbf{u}).$$

We will assume that μ^t is conserved by the dynamics, i.e., $\mu^t = \mu^0$. We now simply denote this measure by μ .

Standard Mori–Zwanzig Decomposition. The Mori–Zwanzig procedure uses a projector operator \mathcal{P} . We define \mathcal{P} as the following conditional expectation (4):

$$\mathcal{P} : \mathbf{B} \mapsto [\mathcal{P}\mathbf{B}](\mathbf{u}) \stackrel{\text{def}}{=} \int \mathbf{B}(\mathbf{u}^*)\delta(\mathcal{A}(\mathbf{u}^*) - \mathcal{A}(\mathbf{u}))\mu(d\mathbf{u}^*)$$

We say that the phase variable C is a function of phase variable D if $D(\mathbf{u}) = D(\mathbf{u}^*)$ implies $C(\mathbf{u}) = C(\mathbf{u}^*)$. As an example $\mathcal{P}\mathbf{B}$ is a function of \mathcal{A} .

Because the Mori–Zwanzig decomposition has been derived by many authors (1, 4), we skip the derivation and simply state the final formula. We define the phase variable F_t (fluctuating term) as the solution of the following partial differential equation with $n + 1$ variables, the orthogonal dynamics equation:

$$F_0 = \mathbf{B} - \mathcal{P}\mathbf{B}, \quad \frac{\partial F_t}{\partial t} = \mathcal{L}F_t - \mathcal{P}\mathcal{L}F_t. \quad [2]$$

Then:

$$e^{t\mathcal{L}}\mathbf{B} = e^{t\mathcal{L}}\mathcal{P}\mathbf{B} + \int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}F_s ds + F_t. \quad [3]$$

For later convenience, we denote S_{\perp}^t the evolution operator associated with the orthogonal dynamics equation:

$$S_{\perp}^t(\mathbf{B}) \stackrel{\text{def}}{=} e^{(t-\mathcal{P})\mathcal{L}}(I - \mathcal{P})\mathbf{B} (= F_t). \quad [4]$$

A key point to observe in Eq. 3 is that $e^{t\mathcal{L}}\mathcal{P}\mathbf{B}$ is a function of $e^{t\mathcal{L}}\mathcal{A}$, and $e^{t\mathcal{L}}\mathcal{P}\mathcal{L}F_s$ is a function of $e^{t\mathcal{L}}\mathcal{A}$. This means that given $e^{s\mathcal{L}}\mathcal{A}$, $0 \leq s \leq t$, we can calculate $e^{t\mathcal{L}}\mathcal{P}\mathbf{B}$ and $\int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}F_s ds$ without knowing the fully resolved trajectory $\mathbf{v}(\mathbf{u}, t)$. In this sense, the first 2 terms in the decomposition satisfy the closure problem. The function $e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}F_s$ is often called the memory kernel because it is a function of past values of \mathcal{A} . In addition, the last term satisfies $\mathcal{P}F_t = 0$ for all t , and may therefore be called the fluctuating term.

Discrete Mori–Zwanzig Decomposition. It is possible to derive a similar looking Mori–Zwanzig decomposition where the continuous integration over time is replaced by a discrete sum. This decomposition can be useful in different contexts, when the data itself is discrete, or when a discretization is applied in numerical computation. For example, the set Ω might be divided into N^{cell} cells and the data $\mathcal{A}(\mathbf{u})$ could be given as a vector of length N^{cell} such that $A_i = 1$ if \mathbf{u} is in cell i and 0 otherwise. The discrete Mori–Zwanzig decomposition can be formulated using an arbitrary map $\mathcal{M} : \mathbf{u} \mapsto \mathcal{M}\mathbf{u}$. For any phase variable \mathcal{A} , we define: $\mathcal{M}\mathcal{A} : \mathbf{u} \mapsto \mathcal{A}(\mathcal{M}\mathbf{u})$. As a typical example, \mathcal{M} can be defined as $\mathcal{M} \stackrel{\text{def}}{=} e^{\Delta t \mathcal{L}}$. Let us define F_k recursively by

$$F_0 \stackrel{\text{def}}{=} \mathbf{B} - \mathcal{P}\mathbf{B}, \quad F_{k+1} \stackrel{\text{def}}{=} \mathcal{M}F_k - \mathcal{P}\mathcal{M}F_k. \quad [5]$$

The following decomposition can be obtained for an arbitrary phase variable \mathbf{B} , with $k \geq 1$ an integer:

$$\mathcal{M}^k \mathbf{B} = \mathcal{M}^k \mathcal{P}\mathbf{B} + \sum_{l=1}^k \mathcal{M}^{k-l} \mathcal{P}\mathcal{M}F_{l-1} + F_k. \quad [6]$$

This decomposition is in the same spirit as the original Mori–Zwanzig decomposition because it satisfies the following properties: $\mathcal{P}\mathbf{B}$ and $\mathcal{P}\mathcal{M}F_l$ are functions of \mathcal{A} , and $\mathcal{P}F_k = 0$. It can be proved by induction.

Generalized Langevin Equations

The Mori–Zwanzig decomposition can be further transformed to reach a form more suitable to construct stochastic models of \mathcal{A} . In particular, this leads to a generalized Langevin equation (10). If we assume that the dynamics is volume preserving ($\nabla \cdot \mathbf{R} \stackrel{\text{def}}{=} 0$), then, using integration by parts and the chain rule, the memory kernel $\mathcal{P}\mathcal{L}F_t$ can be shown to be equal to

$$\begin{aligned} \mathcal{P}\mathcal{L}F_t &= [(\nabla_{\mathcal{A}} - \nabla_{\mathcal{A}}\mathcal{H}) \cdot \mathcal{P}[\mathcal{L}\mathcal{A} \otimes F_t]]^T \\ \mathcal{H} &\stackrel{\text{def}}{=} -\ln \int \delta(\mathcal{A}(\mathbf{u}^*) - \mathcal{A})\mu(d\mathbf{u}^*), \end{aligned}$$

where T is the transpose operator and \otimes is the outer product of 2 vectors. Using this result with $\mathbf{B} \stackrel{\text{def}}{=} \mathcal{L}\mathcal{A}$ along with the Mori–Zwanzig decomposition (Eq. 3), we get a form of the fluctuation dissipation theorem (10, 11):

$$\begin{aligned} e^{t\mathcal{L}}\mathcal{L}\mathcal{A} &= e^{t\mathcal{L}}\mathcal{P}\mathcal{L}\mathcal{A} \\ &+ \int_0^t e^{(t-s)\mathcal{L}} [(\nabla_{\mathcal{A}} - \nabla_{\mathcal{A}}\mathcal{H}) \cdot \mathcal{P}[F_0 \otimes F_s]]^T ds + F_t, \end{aligned} \quad [7]$$

where the memory kernel is related to the autocorrelation of the fluctuations F_t .

This equation can be further simplified to some of the usual forms. We briefly discuss an example. Consider the case of a separable Hamiltonian system $H \stackrel{\text{def}}{=} K(\mathbf{p}) + U(\mathbf{q})$, in the canonical ensemble ($\beta \stackrel{\text{def}}{=} (k_B T)^{-1}$), with atomic positions q_l , momenta p_l , and masses m_l . We may choose \mathcal{A} to be a coordinate and its momentum (ξ, p_{ξ}) . For simplicity we further assume that the mass $m_{\xi}^{-1} \stackrel{\text{def}}{=} \sum_l \frac{1}{m_l} (\frac{\partial \xi}{\partial q_l})^2$ is constant. From Eq. 7, we can prove that the equations of motion are then given by $d\xi/dt = p_{\xi}/m_{\xi}$, and

$$\begin{aligned} \frac{dp_{\xi}}{dt} &= -\frac{dU}{d\xi} + \int_0^t e^{(t-s)\mathcal{L}} \left[\left(\frac{d}{dp_{\xi}} - \frac{\beta p_{\xi}}{m_{\xi}} \right) \mathcal{P}[F_0 F_s] \right] ds + F_t \\ \text{with: } U(\xi) &\stackrel{\text{def}}{=} -\beta^{-1} \ln \int \delta(\xi(\mathbf{q}) - \xi) e^{-\beta U(\mathbf{q})} d\mathbf{q}. \end{aligned} \quad [8]$$

Fokker–Planck Equation

We now derive a Fokker–Planck equation for the resolved variable \mathcal{A} . We have to distinguish between variable \mathcal{A} seen as a function of \mathbf{u} , which is denoted by $\mathcal{A}(\mathbf{u})$, and seen as an independent variable, then denoted a . We apply the Mori–Zwanzig projection (Eq. 3) to the scalar phase variable $B_a \stackrel{\text{def}}{=} \mathcal{L}\delta(\mathcal{A}(\mathbf{u}) - a)$:

$$e^{t\mathcal{L}}B_a = e^{t\mathcal{L}}\mathcal{P}B_a + \int_0^t e^{(t-s)\mathcal{L}}\mathcal{P}\mathcal{L}F_s ds + F_t. \quad [9]$$

We denote $\phi^{\infty}(\mathbf{a})$ the equilibrium probability density function of $\mathbf{a} = \mathcal{A}(\mathbf{u})$. The Fokker–Planck equation assumes a simple form if we choose an initial probability distribution of the form

$$v^0(d\mathbf{u}) \stackrel{\text{def}}{=} \frac{\phi^0(\mathcal{A}(\mathbf{u}))}{\phi^{\infty}(\mathcal{A}(\mathbf{u}))}\mu(d\mathbf{u}),$$

where ϕ^0 is some given initial condition. This corresponds to a constrained equilibrium where variables orthogonal to \mathbf{a} are sampled from the equilibrium distribution while \mathbf{a} is sampled according to ϕ^0 . We denote $\phi^t(\mathbf{a})$ the probability density function corresponding to the initial probability distribution $\nu^0(\mathbf{du})$.

To obtain a Fokker–Planck equation, we need to multiply Eq. 9 by $\nu^0(\mathbf{du})$ and integrate over \mathbf{u} . Using the chain rule, integration by parts and properties of the Dirac δ functions (a long derivation), we can show that this leads to a GFPE:

$$\frac{\partial \psi^t(\mathbf{a})}{\partial t} = -\mathcal{P}_a(\mathcal{L}\mathcal{A})\nabla\psi^t(\mathbf{a}) + \frac{1}{\phi^\infty(\mathbf{a})}\nabla_{\mathbf{a}'}\cdot\int_{\mathbf{a}'\in\Omega}\int_{s=0}^{s=t}\phi^\infty(\mathbf{a}')\mathbf{K}(\mathbf{a},\mathbf{a}',s)^T\nabla\psi^{t-s}(\mathbf{a}')\mathbf{d}\mathbf{a}'\mathbf{d}s, \quad [10]$$

where $\psi^t \stackrel{\text{def}}{=} \phi^t/\phi^\infty$, and \mathbf{K} is a tensor phase-variable:

$$\mathbf{K}(\mathbf{a},\mathbf{a}',s) \stackrel{\text{def}}{=} \mathcal{P}_{\mathbf{a}'}(\mathcal{L}\mathcal{A}(\mathbf{u}) \otimes [\mathcal{S}_\perp^s \mathcal{L}\mathcal{A}(\mathbf{u})\delta(\mathcal{A}(\mathbf{u}) - \mathbf{a})]). \quad [11]$$

The notation $\mathcal{P}_{\mathbf{a}'}$ indicates explicitly the value of $\mathcal{A}(\mathbf{u}) = \mathbf{a}'$ used in the projection.

Numerical Solution Using a Galerkin Discretization

We now discuss how the GFPE may be solved numerically. The direct numerical calculation of the memory kernel $\mathbf{K}(\mathbf{a},\mathbf{a}',s)$ is difficult because of the Dirac δ function in its definition (Eq. 11). However this becomes relatively straightforward if one uses a Galerkin discretization of ψ^t . Suppose we have some basis functions $N_j(\mathbf{a})$ and:

$$\psi^t(\mathbf{a}) \approx \sum_{j=1}^{N^G} \psi_j(t)N_j(\mathbf{a})$$

Galerkin Discretization. The Galerkin formulation is derived from Eq. 10 by multiplying by the test function $N_i(\mathbf{a})$ and integrating over \mathbf{a} . The coefficients $\psi_i(t)$ are then solutions of the following set of integro-differential equations.

$$\begin{aligned} \boldsymbol{\Psi}(t) &\stackrel{\text{def}}{=} [\psi_1(t), \dots, \psi_{N^G}(t)]^T \\ \mathbf{M} \frac{d\boldsymbol{\Psi}}{dt} &= \mathbf{L}_d \boldsymbol{\Psi}(t) + \int_0^t \mathbf{L}_m(s)\boldsymbol{\Psi}(t-s)\mathbf{d}s \end{aligned} \quad [12]$$

with the following matrices:

$$\begin{aligned} \mathbf{M}_{ij} &\stackrel{\text{def}}{=} \int N_i(\mathbf{a})N_j(\mathbf{a})\mathbf{d}\mathbf{a} \\ [\mathbf{L}_d]_{ij} &\stackrel{\text{def}}{=} \int N_j \frac{d}{dt} \left(\frac{N_i}{\phi^\infty} \right) \mu(\mathbf{du}) \\ [\mathbf{L}_m(s)]_{ij} &\stackrel{\text{def}}{=} - \int \frac{dN_j}{dt} \mathcal{S}_\perp^s \left(\frac{d}{dt} \frac{N_i}{\phi^\infty} \right) \mu(\mathbf{du}). \end{aligned}$$

For irreversible processes, $[\mathbf{L}_m(s)]_{ij}$ decays when the support of N_i and N_j are far from one another (diagonally dominant matrix) or when s is large.

Discrete Mori–Zwanzig Decomposition. A second formulation can be obtained by taking advantage of the discrete Mori–Zwanzig formalism (Eq. 6). We choose as \mathcal{M} the time evolution operator over Δt : $\mathcal{M} \stackrel{\text{def}}{=} e^{\Delta t \mathcal{L}}$ and as B_a the scalar phase-variable $(\mathcal{M} - \mathcal{I})\delta(\mathcal{A}(\mathbf{u}) - \mathbf{a})$. Then, we obtain the following scheme ($t_k = k\Delta t$):

$$\mathbf{M}(\boldsymbol{\Psi}(t_{k+1}) - \boldsymbol{\Psi}(t_k)) = \mathbf{L}_d^{(1)} \boldsymbol{\Psi}(t_k) + \sum_{l=1}^k \mathbf{L}_m^{(1)}(l-1)\boldsymbol{\Psi}(t_{k-l}), \quad [13]$$

where

$$\begin{aligned} [\mathbf{L}_d^{(1)}]_{ij} &\stackrel{\text{def}}{=} \int N_j(\mathcal{M} - \mathcal{I}) \frac{N_i}{\phi^\infty} \mu(\mathbf{du}) \\ [\mathbf{L}_m^{(1)}(l)]_{ij} &\stackrel{\text{def}}{=} \int N_j \mathcal{M} \mathcal{S}_\perp^l \left((\mathcal{M} - \mathcal{I}) \frac{N_i}{\phi^\infty} \right) \mu(\mathbf{du}). \end{aligned}$$

The notation \mathcal{S}_\perp^l refers to the discrete orthogonal dynamics defined by Eq. 5. The integer superscript and the context are hopefully sufficient to remove the ambiguity with \mathcal{S}_\perp^s from Eq. 4.

This decomposition is not a finite-difference approximation in time. In particular it gives exactly the same solution as Eq. 12 at times $k\Delta t$. There is no time discretization error. This is because the discrete decomposition (Eq. 6) is an exact equation.

This formulation does not require any derivative of N_i and therefore is also applicable for discontinuous basis functions such as the hat function. ($N_i(x) = 1$ if $0 \leq x \leq \Delta x$ and 0 otherwise.) A simple piecewise constant approximation of $\boldsymbol{\Psi}$ is therefore possible. This makes the numerical implementation relatively simple. This numerical scheme was chosen for the discussion in *Numerical Results*.

Reaction Rate and Metastable Basins. In many chemical systems, metastable basins are separated by energy barriers making transition a rare event. Calculating the rate of transition between these basins is often of great importance. Let us consider Eq. 13, and look for solutions of the form $\boldsymbol{\Psi}(t_n) \stackrel{\text{def}}{=} \mu_l^n \boldsymbol{\Psi}_l$. From these solutions, we will derive the general solution of our problem. Plug in the form of our solution in Eq. 13 and assume that the memory kernel $\mathbf{L}_m^{(1)}(l)$ becomes negligible for $l \geq l_0$. Then, we obtain a polynomial eigenvalue problem

$$[-\mu_l^{l_0+1} \mathbf{M} + \mu_l^{l_0} (\mathbf{L}_d^{(1)} + \mathbf{M}) + \mu_l^{l_0-1} \mathbf{L}_m^{(1)}(0) + \dots + \mathbf{L}_m^{(1)}(l_0-1)] \boldsymbol{\Psi}_l = 0.$$

All solutions to this equation can be associated with solutions of $\mathbf{A}\mathbf{z} = \mu\mathbf{B}\mathbf{z}$ with

$$\mathbf{A} \stackrel{\text{def}}{=} \begin{pmatrix} 0 & \mathbf{I} & 0 & \dots & 0 \\ 0 & 0 & \mathbf{I} & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \mathbf{I} \\ \mathbf{L}_m^{(1)}(l_0-1) & \mathbf{L}_m^{(1)}(l_0-2) & \dots & \mathbf{L}_m^{(1)}(0) & \mathbf{L}_d^{(1)} + \mathbf{M} \end{pmatrix}$$

$$\mathbf{B} \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{I} & & & & \\ & \mathbf{I} & & & \\ & & \ddots & & \\ & & & \mathbf{I} & \\ & & & & \mathbf{M} \end{pmatrix}$$

and $\mathbf{z}^T = (\boldsymbol{\Psi}^T, \mu \boldsymbol{\Psi}^T, \dots, \mu^{l_0} \boldsymbol{\Psi}^T)$. If we assume that the eigenvalue problem $\mathbf{A}\mathbf{z} = \mu\mathbf{B}\mathbf{z}$ admits $N^G(l_0 + 1)$ distinct eigenvalues μ_l and eigenvectors \mathbf{z}_l , we can form the general solution of our problem. Consider the vector

$$\mathbf{z}(t_0)^T \stackrel{\text{def}}{=} (\boldsymbol{\Psi}(t_0)^T, \boldsymbol{\Psi}(t_1)^T, \dots, \boldsymbol{\Psi}(t_{l_0})^T)$$

for a given initial condition $\boldsymbol{\Psi}(t_0)$. It can be expanded in the basis \mathbf{z}_l : $\mathbf{z}(t_0) = \sum_{l=1}^{N^G(l_0+1)} c_l \mathbf{z}_l$. By induction and using Eq. 13, we can prove that the general solution is of the form $\mathbf{z}(t_n) = \sum_{l=1}^{N^G(l_0+1)} c_l (\mu_l)^n \mathbf{z}_l$, and in terms of $\boldsymbol{\Psi}$:

$$\boldsymbol{\Psi}(t_n) = \sum_{l=1}^{N^G(l_0+1)} c_l (\mu_l)^n \boldsymbol{\Psi}_l, \quad [14]$$

where $\boldsymbol{\Psi}_l$ is the vector formed by taking the first N^G components of \mathbf{z}_l . Note that the $\boldsymbol{\Psi}_l$ are not linearly independent because they

are in a space of dimension N^G but we do need $N^G(l_0 + 1)$ such vectors to expand the general solution.

In many chemical reactions, there is a (or a few) time scale that is very slow compared to the other time scales in the system. This is the case for example when an energy barrier separates two metastable basins. That time scale can be obtained from μ_l . One of the eigenvalues must be equal to 1 and the corresponding eigenvector Ψ_0 is the equilibrium distribution. All others eigenvalues in general have a real part in the interval $] -1, 1[$. In many instances, there is a single (or a few) eigenvalue μ_1 close to 1; this is the slowest time scale in the system. The corresponding reaction rate is given by $-\ln(\mu_1)/\Delta t$. This is the rate of transition across the energy barrier separating the 2 most stable basins. The eigenvector Ψ_1 is approximately constant in each metastable basin but changes sign between basins. The change of sign can be used to identify precisely the boundary of metastable sets (for additional details see ref. 12 for example). This is illustrated in *Numerical Results*. More generally, all time scales in the system can be extracted from the eigenvector decomposition, for example the first passage times. If needed the conditional probability $p(a, t|a_0, 0)$ can be obtained.

Orthogonal Dynamics Equation

Both the GLE and GFPE require computing the solution of the orthogonal dynamics equation, which is a partial differential equation in dimension $n + 1$. A direct solution is impractical in most cases. Various strategies with low computational cost have been proposed most notably by Lange et al. (9) and Chorin et al. (7). In ref. 9, the authors reconstruct the memory kernel in the GLE from the velocity autocorrelation; their equation is derived from Eq. 8 assuming that $\mathcal{P}[F_0 F_s]$ is independent of p_ξ . This leads to a Volterra integral equation of the first kind which is difficult to solve numerically. In ref. 7, a computationally efficient scheme is proposed to calculate $\mathcal{P}LF_s$ using a Galerkin approach.

We now assume that we have a numerical algorithm, e.g., Molecular Dynamics or Monte-Carlo, that allows generating samples \mathbf{u} with a distribution equal to (close to) the equilibrium distribution μ . We define the following notations. N^{sam} : number of sample points \mathbf{u} ; \mathbf{u}_0^k : sample k ; N^G : number of basis functions $N_i(\mathbf{a})$; N^{mem} : number of discrete times s at which the memory term is computed. Assume we integrate in time using some numerical procedure; we denote \mathbf{u}_m^k the sample at step m , using \mathbf{u}_0^k as initial condition.

Given a function G , there are several ways to numerically approximate $\mathcal{P}G$ from $G(\mathbf{u}_0^k)$. For our application, we consider a Galerkin expansion of the form $\mathcal{P}^G G(\mathbf{a}) \stackrel{\text{def}}{=} \sum_j P_j N_j(\mathbf{a})$ with

$$\sum_j M_{ij} P_j = \frac{1}{N^{\text{sam}}} \sum_k \frac{N_i(\mathbf{A}(\mathbf{u}_0^k)) G(\mathbf{u}_0^k)}{\phi^\infty(\mathbf{A}(\mathbf{u}_0^k))}.$$

The pseudocode to numerically compute F_s is then given by

For $l = 0$ **to** $N^{\text{mem}} - 2$, **do**

 Calculate $\mathcal{P}^G G$ with $G(\mathbf{u}_0^k) = F_l(\mathbf{u}_1^k)$

 Calculate $F_{l+1}(\mathbf{u}_m^k) = F_l(\mathbf{u}_{m+1}^k) - (\mathcal{P}^G G)(\mathbf{A}(\mathbf{u}_m^k))$,
 for $0 \leq m \leq N^{\text{mem}} - 2 - l$

end

This is a numerical implementation of Eq. 5 where \mathcal{M} is the evolution operator $e^{\Delta t \mathcal{L}}$. In the limit of taking $\Delta t \rightarrow 0$, the sequence F_k in Eq. 5 ($0 \leq k \leq t_1/\Delta t$) converges to $F_{k\Delta t}$ in Eq. 2; the single step error is $O(\Delta t^2)$. It is possible to derive higher order integrators, however, in practice, statistical errors incurred when approximating \mathcal{P} are larger than the time discretization errors. The cost of this calculation is $O(N^{\text{sam}}(N^{\text{mem}})^2)$. If we apply this to calculate the matrices in Eq. 12 or 13, the total cost is $O(N^{\text{sam}}(N^{\text{mem}})^2 N^G)$. This assumes that the basis N_i has local support. If the basis has global support, e.g., Legendre polynomials, the total cost is $O(N^{\text{sam}}(N^{\text{mem}})^2 N^G + N^{\text{sam}} N^{\text{mem}} (N^G)^2)$.

In the presence of energy barriers, it is possible to generate samples \mathbf{u}_0^k from the constrained ensemble, that is for various \mathbf{a}_0 we generate samples lying on the surface $\mathbf{A}(\mathbf{u}) = \mathbf{a}_0$. This is sufficient to calculate \mathcal{P}^G and the efficiency of the method becomes independent of $\phi^\infty(\mathbf{a})$ and in particular of energy barriers along $\mathbf{A}(\mathbf{u})$. If the system does not exhibit large energy barriers, it is possible to carry out this calculation using a single (a few) very long trajectory. In that case the total cost using our approach is reduced to $O(N^{\text{sam}} N^{\text{mem}} N^G)$ for a basis with local support.

Accuracy and Limitations

The method works irrespective of energy barriers along $\mathbf{A}(\mathbf{u})$. However it relies on techniques to numerically estimate \mathcal{P} . This requires being able to efficiently sample the surface $\mathbf{A}(\mathbf{u}) = \mathbf{a}$. Roughly speaking, if the coordinates orthogonal to \mathbf{A} contain metastable basins, the number of steps in a molecular dynamics simulation required to generate N^{sam} uncorrelated points is very large. A precise statement is beyond the scope of this paper and depends in general on the rate of decay of the auto-correlation of \mathbf{B} when moving on the surface $\mathbf{A}(\mathbf{u}) = \mathbf{a}$. If we consider a simulation in the hypersurface $\mathbf{A}(\mathbf{u}) = \mathbf{a}$, the previous analysis (see *Reaction Rate and Metastable Basins*) shows that the number of steps required should be proportional to $-1/\ln(\max_a \mu_1^+(\mathbf{a}))$, where $\mu_1^+(\mathbf{a})$ is the largest eigenvalue different from 1 for the constrained dynamics with $\mathbf{A}(\mathbf{u}) = \mathbf{a}$. In this case, specific acceleration techniques must be applied. We mention the technique of Zheng et al. ((13), biasing force along ∇A and ∇F_A), normal mode approximations (14), biasing techniques, etc. These techniques allow lowering $\max_a \mu_1^+(\mathbf{a})$ away from 1.

We also note that the method will be computationally expensive to apply to problems where the dimensionality of \mathbf{A} is large. This is because all the functions of \mathbf{A} become difficult to discretize in an efficient manner. Techniques like sparse grid of Smolyak (15) may become useful in those cases.

Numerical Results

Oscillators. It is possible for some systems to calculate analytically the fluctuating term F_t . We will use such a system to check the accuracy of Eq. 5 and the pseudo-code above. Consider the case of a particle attached to some masses with springs:

$$\frac{d^2 \xi}{dt^2} \stackrel{\text{def}}{=} -\frac{dU}{d\xi} - \sum_{i=1}^{N_p} k_i (\xi - x_i), \quad \frac{d^2 x_i}{dt^2} \stackrel{\text{def}}{=} -\omega_i^2 (x_i - \xi)$$

We can derive the following equation for ξ'' (5):

$$\frac{d^2 \xi}{dt^2} = -\frac{dU}{d\xi} - \int_0^t \left(\sum_{i=1}^{N_p} k_i \cos(\omega_i s) \right) \xi'(t-s) ds + F_t, \quad [15]$$

$$\text{with : } F_t \stackrel{\text{def}}{=} \sum_{i=1}^{N_p} k_i (C_i^1 \cos(\omega_i t) + C_i^2 \sin(\omega_i t)).$$

If we assume that the initial positions and momenta of the masses x_i are generated randomly according to the canonical distribution at temperature T ($= (k_B \beta)^{-1}$), then $C_i^1 = 1/\sqrt{\beta k_i} \eta_i$, $C_i^2 = 1/\sqrt{\beta k_i} \zeta_i$, with η_i and ζ_i normally distributed variables with variance 1 and mean 0. Consequently,

$$\mathcal{P}[F_t F_{t+s}] = \frac{1}{\beta} \sum_{i=1}^{N_p} k_i \cos(\omega_i s). \quad [16]$$

This is consistent with Eqs. 8 and 15.

To demonstrate the accuracy of Eq. 5 and its correspondence with Eq. 2, we use Eq. 5 to calculate F_t (using the pseudo-code above) and then calculate $\mathcal{P}[F_t F_{t+s}]$ as a function of s . The result is compared with the analytical expression given by Eq. 16. Choose

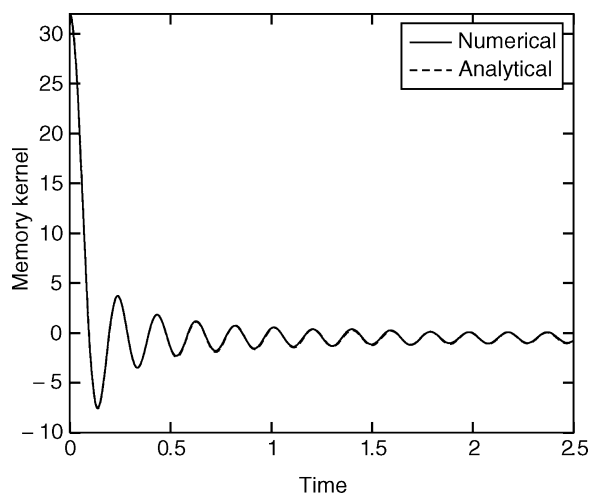


Fig. 1. Oscillators with $\omega_i = i$ and $k_i = 1$. The numerical solution is compared with an analytical expression (Eq. 16). Eq. 5 and the pseudocode in *Orthogonal Dynamics Equations* were used.

for example $N_p = 32$ particles with $\omega_i = i$, $k_i = 1$. As the number of particles goes to infinity, $\mathcal{P}[F_t F_{t+s}]$ approximates a Dirac δ function at 0. Fig. 1 shows a comparison of Eqs. 5 and 16. We used a trajectory with $4 \cdot 10^7$ steps and a step size of 5×10^{-3} . The

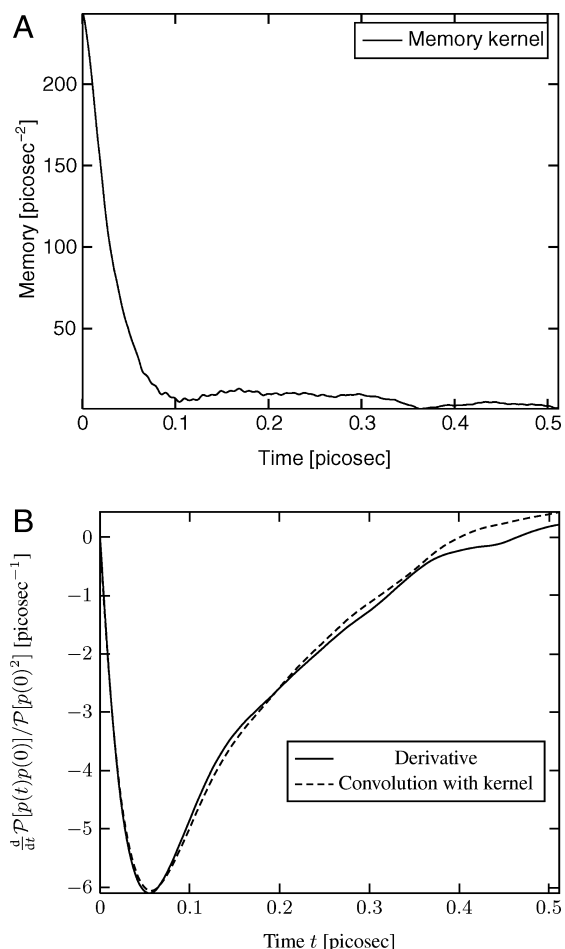


Fig. 2. (A) Memory kernel $\beta/m\mathcal{P}[F_0 F_s]$ (see Eq. 8) for fluctuating atomic forces exerted on a protein by water molecules. (B) Derivative of the velocity auto-correlation function. We plot the left and right hand sides of Eq. 17 as an indirect way to verify our computation of $\mathcal{P}[F_0 F_s]$.

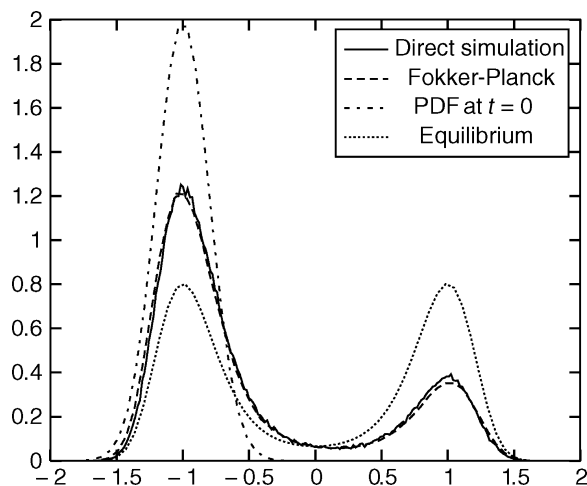


Fig. 3. Probability density functions (PDF) at time $16\Delta t = 4.1$. The label “PDF at $t = 0$ ” corresponds to the initial distribution of x . The label “Direct simulation” corresponds to an estimate of the PDF based on running many trajectories using Langevin dynamics. The label “Fokker-Planck” corresponds to the solution computed using Eq. 13. The label “Equilibrium” is the reference equilibrium distribution of x . The variable x is on the horizontal axis.

trajectory was generated using Langevin dynamics with a friction coefficient of 0.05 and a temperature of $k_B T = 1$. We note that the decay of the memory kernel happens on a time scale comparable to the time scale of ξ , and therefore the adiabatic approximation does not apply.

Implicit Water Model. It is common in molecular dynamics simulations of solvated molecules (e.g., protein) to model water using an implicit model. In that case, the water molecules are removed from the system and replaced by a model; the mean force may be estimated using various techniques such as the Poisson–Boltzmann equation or the Born and Onsager models (16). The fluctuating part is typically approximated by a Langevin term with friction and white noise. We revisit this problem using our approach.

We chose a small poly-peptide (alanine dipeptide) in water. This is a 22-atom molecule. We used 450 water molecules. The box size was $25.1 \times 24.5 \times 23 \text{ \AA}$. We considered the total atomic force that water is exerting on the protein and A is the location of the center of mass of the protein. The memory kernel $\beta/m\mathcal{P}[F_0 F_s]$ (see Eq. 8) is shown on Fig. 2.

In that case, we could not compare with a reference solution. However the following indirect verification was conducted. In Eq. 8, if we multiply by $p(0)$, average over all initial conditions, and neglect the derivative with respect to p (see ref. 9), we get

$$\frac{d}{dt} \mathcal{P}[p(t)p(0)] = - \int_0^t \mathcal{P}[p(t-s)p(0)] \frac{\beta}{m} \mathcal{P}[F_0 F_s] ds. \quad [17]$$

As a way to verify our calculation of $\mathcal{P}[F_0 F_s]$, we plot the left and right hand sides of Eq. 17 in Fig. 2. The agreement is very good.

Non-Markovian Fokker-Planck Equation. We tested our numerical scheme to calculate the GFPE. A comparison with a direct brute force calculation is made. We used the discrete Mori–Zwanzig scheme described in *Discrete Mori–Zwanzig Decomposition*.

Consider a particle x in a double well potential given by $x^2/4(x^2 - 2)$. This function has 2 minima at -1 and 1 , and a local maximum at 0 . We attached to the particle 16 other particles using springs, with stiffnesses chosen such that the kernel $\mathcal{P}[F_t F_{t+s}]$ decays approximately like e^{-20s} (see Eq. 16 and ref. 5). The temperature was chosen such that $k_B T = 0.1$. This corresponds to a barrier of $2.5 k_B T$. We generated trajectories using a Langevin equation with a friction of 1. The time step for the

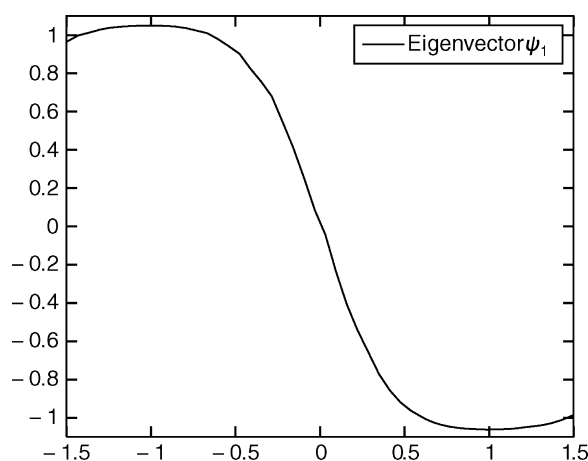


Fig. 4. Eigenvector ψ_1 vs. x_j . The eigenvalue for ψ_1 is 0.99462. The 0 of the function separates the 2 metastable basins.

integration was 0.001. The time interval Δt (see page 3) is equal to 0.256. In our implementation of Eq. 13, we did not generate a single long trajectory as this would have resulted in poor statistics near -2 and 2 . Instead we created bins of size 0.0625 and in each bin we generated a fixed number of initial conditions drawn from the constrained canonical ensemble distribution. For each initial condition, we ran a trajectory of length $16\Delta t = 4096$ steps. This algorithm generated accurate data with small statistical errors.

After computing the drift matrix $L_d^{(1)}$ and the memory matrices $L_m^{(1)}(l)$, we solved the GFPE numerically and compared with a brute force calculation. The initial conditions are taken from a Gaussian distribution centered at -1 with standard deviation 0.2 . Trajectories were run with these initial conditions and the final value of x was recorded after $32,768$ steps. This corresponds to $128\Delta t$. The resulting probability density functions are shown on Fig. 3.

We computed the eigenvectors and eigenvalues of the polynomial eigenvalue problem as described in the section “Reaction rate and metastable basins.” One of the eigenvalues was found to be almost equal to 1. The corresponding eigenvector matched the equilibrium distribution. The second eigenvalue is 0.9946. The third is 0.84 and the other eigenvalues have a smaller real part. In Fig. 4, we plot the vector ψ_1 (see Eq. 14). As described in *Reaction Rate and Metastable Basins*, we expect this vector to change sign at the transition region between the 2 metastable basins, $x < 0$ and $x > 0$. This is the case; we computed that the 0 of the function is near $x = 0.001$.

In addition the associated eigenvalue gives a rate equal to $-\ln(\mu_1)/\Delta t = 0.021$ [time unit] $^{-1}$. We compared this rate with an estimate based on the brute force calculation shown in Fig. 3 (solid curve): This gave 0.021 [time unit] $^{-1}$. The transition state theory from ref. 17 predicts a rate of 0.033 [time unit] $^{-1}$, which is consistent with the fact that a rate from transition state theory overestimates the actual rate since re-crossing of the transition region at $x = 0$ is not accounted for.

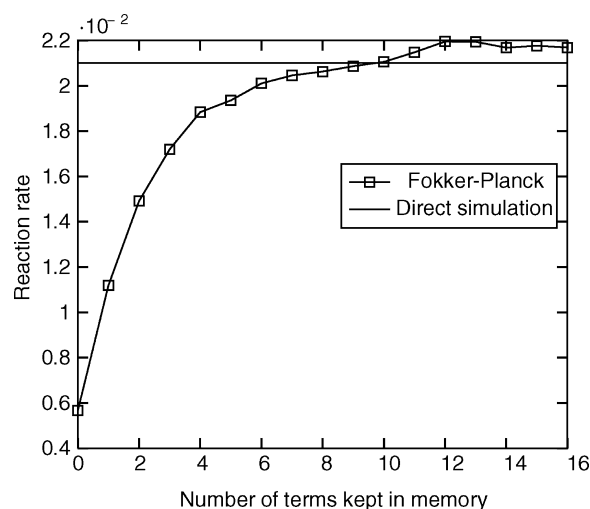


Fig. 5. Reaction rate, or rate of transition, vs. number of terms kept in the memory sequence $L_m^{(1)}(l)$. The direct simulation estimate is the horizontal line and is equal to 0.021.

In Fig. 5, we calculated the rate using the Fokker–Planck equation while varying the number of terms we keep in the memory kernel, that is for a given integer l on the x axis we only keep the terms $L_m^{(1)}(0), \dots, L_m^{(1)}(l-1)$. Fig. 5 shows the effect of the memory kernel on the rate, which is essentially multiplied by 3.5 when we keep 10 terms in the memory kernel. For $l > 9$, the memory kernel $L_m^{(1)}(l)$ is small and dominated by statistical noise. This plot shows the importance of memory and the non-Markovian effects in the evolution of ϕ^l .

Conclusion

We presented a theoretical framework and numerical techniques to calculate generalized Langevin equations and non-Markovian Fokker–Planck equations by sampling trajectories. A discrete form of the Mori–Zwanzig formalism has been derived (Eq. 6). A generalized non-Markovian Fokker–Planck equation was presented in a general setting (Eq. 10), along with its numerical discretization (Eq. 13). An algorithm to calculate the various terms in these equations is given. An important element is the procedure used to solve the orthogonal dynamics equation numerically. The accuracy of the method was shown with different examples, including one with analytical solutions, one with atomic forces exerted by water molecules on a polypeptide, and a problem with 2 metastable basins for the Fokker–Planck equation. We haven’t addressed the question of estimating statistical errors, and cases with large energy barriers.

ACKNOWLEDGMENTS. We thank Jesús Izaguirre and Chris Sweet from Notre-Dame University for providing the data for alanine dipeptide and Alexandre Chorin for advice and support. This work was funded in part by the NASA Research Center, the United States Army, and the BioX program at Stanford University.

- Mori H (1965) Transport, collective motion, and Brownian motion. *Prog Theo Phys* 33:423–455.
- Zwanzig R (2001) *Nonequilibrium Statistical Mechanics* (Oxford Univ Press, New York).
- Evans D, Morriss G (1990) *Statistical Mechanics of Nonequilibrium Liquids* (Academic, London).
- Chorin AJ, Hald OH, Kupferman R (2000) Optimal prediction and the Mori–Zwanzig representation of irreversible processes. *Proc Natl Acad Sci USA* 97:2968–2973.
- Givon D, Kupferman R, Stuart A (2004) Extracting macroscopic dynamics: model problems and algorithms. *Nonlinearity* 17:R55–R127.
- Berkowitz M, Morgan JD, McCammon JA (1983) Generalized Langevin dynamics simulations with arbitrary time-dependent memory kernels. *J Chem Phys* 78:3256–3261.
- Chorin AJ, Hald AH, Kupferman R (2002) Optimal prediction with memory. *Physica D* 166:239–257.
- Akkermans RLC, Briels WJ (2000) Coarse-grained dynamics of one chain in a polymer melt. *J Chem Phys* 113:6409–6422.
- Lange O, Grubmüller H (2006) Collective Langevin dynamics of conformational motions in proteins. *J Chem Phys* 124:214903.
- Akkermans RLC (2006) *New Algorithms for Macromolecular Simulation*. Lecture Notes in Computational Science and Engineering (Springer, Berlin), Vol 49, pp 155–165.
- Ciccotti G, Ryckaert J-P (1981) On the derivation of the generalized Langevin equation for interacting Brownian particles. *J Stat Phys* 26:73–82.
- Schütte C, Fischer A, Huisinga W, Deuffhard P (1999) A direct approach to conformational dynamics based on hybrid Monte Carlo. *J Comp Phys* 151:146–168.
- Zheng L, Chen M, Yang W (2008) Random walk in orthogonal space to achieve efficient free-energy simulation of complex systems. *Proc Natl Acad Sci USA* 105:20227–20232.
- Sweet C, Petrone P, Pande V, Izaguirre JA (2008) Normal mode partitioning of Langevin dynamics for biomolecules. *J Chem Phys* 128:145101.
- Smolyak SA (1963) Quadrature and interpolation formulas for tensor products of certain classes of functions. *Dokl Akad Nauk SSSR* 4:240–243.
- Leach A (2001) *Molecular Modelling: Principles and Applications* (Prentice Hall, New York), 2nd Ed.
- Chandler D (1987) *Introduction to Modern Statistical Mechanics* (Oxford Univ Press, Oxford), 1st Ed.