SIAM J. SCI. COMPUT.
Vol. 25, No. 6, pp. 1964–1981

ROBUST VARIANCE REDUCTION FOR RANDOM WALK METHODS*

GANG ZOU† AND ROBERT D. SKEEL‡

Abstract. Random walk methods are effective for solving linear partial differential equations in many dimensions, especially those involving complex geometries. They are based on an equivalence given by a Feynman–Kac formula between an expectation of a functional of a stochastic process and the solution at a point of a partial differential equation. The drawback is that the error is proportional only to the square root of the reciprocal of the number of trials. Efficiency depends critically on variance reduction. A general strategy for doing this in the case of stochastic differential equations is proposed by Milstein. The idea is to introduce a bias in the drift term and to exactly compensate for this by unequal weighting of the trials. There is an optimal bias defined in terms of the solution of the partial differential equation which reduces the variance to zero. In practice, an approximation is used. This idea has been tested under the name "biased Brownian dynamics" on the problem of calculating rate constants for diffusion-limited reactions. The approach is successful in some cases but is less successful in more difficult cases due to the occasional occurrence of a well-above-average weight. Proposed and tested here is a weight control algorithm, which greatly enhances the effectiveness of biased Brownian dynamics.

Key words. variance reduction, importance sampling, random walk methods, path integrals, stochastic differential equations

AMS subject classifications. 65C30, 65C05, 65N99, 92C45

DOI. 10.1137/S1064827503424025

1. Introduction.

Random walk methods are effective for certain calculations expressed in terms of linear partial differential equations in many dimensions, especially for complex geometries. Such methods necessarily have an order of accuracy of only 1/2 in terms of the number of trials N_{trials}, which can be ameliorated only by reducing the constant of proportionality by some type of variance reduction. A general approach based on importance sampling is proposed and analyzed by Milstein [13]. The idea is to introduce a systematic bias into the walk and perfectly adjust for this with a suitable unequal weighting of the results of different random walks. Milstein proves that there is an ideal bias that reduces the variance to zero. In practice, the bias "force" has to be guessed or estimated. For example, the article [24] on "biased Brownian dynamics" (BD) uses a heuristic choice of bias force. In this way it obtains a sevenfold speedup for the calculation of the rate constant for eucaryotic superoxide dismutase (SOD). However, when the same method is applied to bacterial SOD, which has a rate constant of one-tenth less, the biased method is no better than the unbiased method. The cause of the inaccuracy, it turns out, is the occurrence of occasional much-larger-than-average weights, which significantly increases the variance. The main contribution of this article is to introduce a *weight control* mechanism for the biased BD algorithm, which makes the combined algorithm

*Received by the editors March 5, 2003; accepted for publication (in revised form) September 30, 2003; published electronically May 25, 2004. This material is based upon work supported by the National Science Foundation under grants 9974555 and 020442.

http://www.siam.org/journals/sisc/25-6/42402.html

†Renaissance Technologies, 600 Route 25A, East Setauket, NY 11733 (gangzou@rentec.com). The work of this author was supported in part by a Computational Science and Engineering Fellowship (University of Illinois).

‡Department of Computer Science and Beckman Institute, University of Illinois, 201 North Goodwin Avenue, Urbana, IL 61801-2302 (skeel@cs.uiuc.edu).

robust and more efficient, and to determine mathematically the *ideal weight*, which is a guide to finding a good *target weight* for the weight control mechanism.

The basis for the random walk method is the relationship between a *stochastic differential equation* (SDE) and the corresponding Fokker–Planck equation, which is a linear *partial differential equation* (PDE) that governs the evolution of the probability density of the SDE. With the use of the Feynman–Kac formula and its variant for elliptic PDEs, many calculations have two equivalent formulations: one in terms of the expectation of a functional of the paths of an SDE (a path integral) and the other in terms of a functional of the solution of a PDE. An example, used to demonstrate the ideas of this article, is the calculation of the rate constant for a diffusion-limited reaction between an enzyme and a substrate whose motion is modeled by a system of SDEs. The problem can be reduced to a reaction probability problem for an SDE or an equivalent elliptic PDE, known as the Smoluchowski equation. An overview of SDEs, random walk methods for PDEs, and the enzyme–substrate reaction problem is given in section 2.

The reaction probability obtained from a BD simulation is the expectation of a random number ζ. The statistical error is proportional to $\sqrt{\mathrm{Var}\zeta/N_{\mathrm{trials}}}$, where N_{trials} is the number of simulated trajectories. For a given statistical error, the cost N_{trials} is proportional to $\mathrm{Var}\zeta$. The reaction probability is generally very small, especially for high-dimensional problems because of the entropy barrier. Standard BD rarely encounters reactions, thus resulting in a large variance relative to the expectation, and a very large N_{trials} is required to obtain acceptable results.

Variance reduction methods for stochastic simulations based on the work of Milstein [13] and Wagner [20, 21] are discussed in detail in the book by Kloeden and Platen [9]. Milstein introduces an additional drift term in the SDE, which is the same as doing a Girsanov transformation to the underlying probability measure. Wagner derives unbiased estimators for functional integrals of stochastic process based on the general principles of Monte Carlo integration. These methods are used by researchers in other areas, such as Melchior and Öttinger in [10, 11], to determine hydrodynamic properties for polymers, and Zuckerman and Woolf in [25], to calculate conformational transition rates. Another, more recent technique for variance reduction is the method of control variates. A recent and readable account comparing importance sampling to control variates is found in [12].

In the context of rate constant calculations, the *weighted ensemble Brownian dynamics* (WEBD) method of Huber and Kim [7] achieves variance reduction by maintaining an ensemble of paths, or trajectories. Our method—biased BD [24]—is motivated by the desire to find a streamlined alternative to the WEBD method, which avoids any coupling between trajectories; hence, biased BD is as embarrassingly parallel as standard BD. It is an adaptation to elliptic PDEs of the importance sampling formalism of Milstein [13] for parabolic PDEs. (It would be interesting to know how this might compare to the method of control variates in the case of stationary problems.)

Section 3 introduces and motivates the biased BD algorithm. It proves that with a biasing force present, the expectation of the weighted exit value gives the unbiased estimate to the reaction probability. Also, it introduces the optimal bias force and proves that the variance of the weighted exit value is zero under the optimal bias force.

Section 4 introduces the weight control algorithm for biased BD, which is needed to make biased BD robust and more efficient. It is proved that under the optimal bias force, the weight of a "particle" in configuration space is a deterministic function

of coordinates and does not depend on the path it traverses. This ideal weight under the optimal bias force is not practical to calculate, but a rough approximation can be calculated and used as a target weight for the weight control algorithm. If the weight of the particle exceeds a prescribed multiple of the target weight, then the particle is split into two and the sum of the weighted exit values is used. If the particle weight becomes less than a prescribed fraction of the target weight, with equal probability either the weight is doubled or the particle is destroyed. This method of control is from [5]. The WEBD method has a more elaborate mechanism that effects weight control.

Finding the optimal bias force or ideal weight is as hard as solving the reaction probability problem. Both the bias force for biased BD and the target weight for weight control are constructed from an approximation to the solution $u(x)$ of the Smoluchowski equation. A method for obtaining approximate solutions is given in [23, 22]. The method is to approximate $u(x)$ by $\hat{u}(\xi(x))$, where $\xi(x)$ is a reaction coordinate defined for every configuration. The PDE for $u(x)$ is converted to a two-point boundary value problem for $\hat{u}(\xi)$ based on the variational form of the boundary value problem, and $\hat{u}(\xi)$ is calculated by a Monte Carlo calculation and a one-dimensional integration.

Section 5 gives the results of experiments demonstrating the effectiveness of weight control. Experiments with *E. coli* SOD show speedups of 4.6 and 5.1. Experiments with a difficult artificial problem used in [7] show speedups of 55 and 31. The performance improvement is partly due to a decrease in the average duration of a trajectory and partly due to a reduction of variance.

2. Random walk methods. The underlying connection between SDEs and PDEs enables the random walk method to solve PDEs. This section reviews some helpful mathematical definitions and formulas for SDEs and the relationship between SDEs and PDEs. Most of the material here can be found in typical books on SDEs, such as [19, 4, 9, 16].

2.1. Review of SDEs. An N-dimensional *canonical* Wiener process $W(t) = W(t,\omega)$, $t \geq 0$, ω an outcome from a sample space, is defined as a random process with the following properties:

1. $W(0) = 0$.
2. For any integer k, reals $0 < t_1 < t_2 < \cdots < t_k$, and $x_1, x_2, \ldots, x_k \in \mathbb{R}^N$, the probability that $W(t_i) \in [x_i, x_i + \mathrm{d}x_i]$ for all $i = 1, 2, \ldots, k$ is

$$p(t_1, 0, x_1)p(t_2 - t_1, x_1, x_2)\cdots p(t_k - t_{k-1}, x_{k-1}, x_k)\mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}x_k,$$

   where

$$p(t, x, x') = (2\pi t)^{-N/2}\exp\left(\frac{-(x-x')^{\mathsf{T}}(x-x')}{2t}\right).$$

3. For any outcome ω, $W(t,\omega)$ (a function $\mathbb{R}^+ \to \mathbb{R}^N$) is continuous with respect to t.

A Wiener process $W(t)$ can be viewed as a family of Gaussian random variables parameterized by t having

$$\mathrm{E}(W(t)) = 0, \qquad \mathrm{E}(W(t)W(s)^{\mathsf{T}}) = \min(s,t)I,$$

where I is the $N \times N$ identity matrix. Computationally, a Wiener process can be

realized at a finite set of points $0 = t_0 < t_1 < t_2 < \cdots < t_{nsteps}$ by $W(t_n) - W(t_{n-1}) = \sqrt{t_n - t_{n-1}} Z^n$, where Z^n is a vector of independent standard Gaussian distributed random numbers, for which the probability density function (PDF) is

$$p_{\mathrm{g}}(z) = (2\pi)^{-N/2} \exp\left(-\frac{1}{2} z^{\mathrm{T}} z\right). \tag{2.1}$$

A single standard Gaussian random variable has mean 0 and variance 1.

A time homogeneous SDE has the form

$$\mathrm{d}X(t) = a\left(X(t)\right) \mathrm{d}t + B\left(X(t)\right) \mathrm{d}W(t), \tag{2.2}$$

where $X(t)$ is an N-dimensional stochastic process, $a(x)$ is an N-vector of real functions, $B(x)$ is an $N \times M$ matrix of real functions, and $W(t)$ is an M-dimensional canonical Wiener process. In this article, the Ito interpretation is used for (2.2), which for simplicity can be treated as the $\mathrm{d}t \to 0$ limit of the Euler–Maruyama scheme applied to (2.2):

$$X(t + \mathrm{d}t) - X(t) = a\left(X(t)\right) \mathrm{d}t + B\left(X(t)\right) \left(W(t + \mathrm{d}t) - W(t)\right). \tag{2.3}$$

Note that on the right-hand side, $a(x)$ and $B(x)$ are evaluated at time t. Another choice is to evaluate them at $t + \mathrm{d}t/2$, which results in the Stratonovich interpretation for the SDE.

Consider a function $\varphi(t, x) \in \mathrm{C}^{1,2}(\mathbb{R} \times \mathbb{R}^N)$ (which means both $\partial_s \varphi(t, x)$ and $\nabla_x \nabla_x^{\mathsf{T}} \varphi(t, x)$ are continuous functions, where ∇_x denotes a column vector of partial derivatives with respect to the components of x), and let $Y(t) = \varphi(t, X(t))$ be another stochastic process. The chain rule for an SDE is different from the chain rule for an ordinary differential equation (ODE). For an ODE, one needs only the first order Taylor expansion for $\varphi(t, x)$, namely, $\mathrm{d}Y(t) = \partial_t \varphi(t, X(t))\mathrm{d}t + \nabla_x^{\mathsf{T}} \varphi(t, X(t))\mathrm{d}X(t)$. However, the Ito interpretation of stochastic integrals leads to the *Ito formula*

$$\begin{aligned} \mathrm{d}Y(t) &= \partial_t \varphi(t, X(t))\mathrm{d}t + \nabla_x^{\mathsf{T}} \varphi(t, X(t))\mathrm{d}X(t) + \frac{1}{2}(\nabla_x \nabla_x^{\mathsf{T}} \varphi(t, X(t))) : (\mathrm{d}X(t)\mathrm{d}X(t)^{\mathsf{T}}) \\ &= (\partial_t + \mathcal{L}(X(t)))\varphi(t, X(t))\mathrm{d}t + (\nabla_x \varphi(t, X(t)))^{\mathsf{T}} B(X(t))\mathrm{d}W(t), \end{aligned} \tag{2.4}$$

where $\mathcal{L}(x)$ is the *characteristic operator* of (2.2),

$$\mathcal{L}(x)u(x) = \left(a(x)^{\mathsf{T}} \nabla_x + \frac{1}{2} B(x)B(x)^{\mathsf{T}} : \nabla_x \nabla_x^{\mathsf{T}}\right) u(x), \tag{2.5}$$

and $A : C$ means $\mathrm{tr}(A^{\mathsf{T}} C)$ for two matrices A and C. Here ∇_x operates only on the functions following the symbol; for example, ∇_x does not operate on $B(x)B(x)^{\mathsf{T}}$. Note the difference in the Ito formula is that $\mathrm{d}X(t)\mathrm{d}X(t)^{\mathsf{T}}$ gives a first order term $B(X(t))B(X(t))^{\mathsf{T}}\mathrm{d}t$, because formally $\mathrm{d}W(t)\mathrm{d}W(t)^{\mathsf{T}}$ is equivalent to $I\mathrm{d}t$, where I is the $M \times M$ identity matrix.

With the Ito formula, it is not hard to prove that for any function $\varphi(t, x) \in \mathrm{C}^{1,2}(\mathbb{R} \times \mathbb{R}^N)$

$$\lim_{h \downarrow 0} \frac{1}{h} \left(\mathrm{E}_{s,x} \varphi(s + h, X(s + h)) - \varphi(s, x)\right) = \left(\partial_s + \mathcal{L}(x)\right) \varphi(s, x), \tag{2.6}$$

where $\mathrm{E}_{s,x}$ denotes the expectation with respect to the probability law of the stochastic process governed by (2.2) with initial condition $X(s) = x$. This is the reason that $\mathcal{L}(x)$ is called the characteristic operator of (2.2). In some mathematics-oriented SDE books such as [16], the characteristic operator is actually defined by (2.6). With this equation, it is not hard to prove the formula for random walk methods in the next subsection.

2.2. A Feynman–Kac formula for elliptic PDEs. The solutions of many linear PDEs can be expressed as path integrals. Since path integrals are typically computed numerically by random walk simulation, path integral methods are also known as random walk methods.

A Dirichlet boundary condition is treated by stopping the random walk at the boundary. A *stopping time* is a random time τ, which depends only on the history of the trajectory up to time τ. A good example of a stopping time is the first exit time from domain Ω, $\tau_\Omega = \inf\{t : X(t) \in \partial\Omega\}$. In the following, the first exit time τ_Ω will be used as the stopping time.

The elliptic Dirichlet problem on domain $\Omega \subset \mathbb{R}^N$,

$$\begin{aligned} -\mathcal{L}(x)u(x) &= 0, \quad x \in \Omega, \\ u(x) &= f(x), \quad x \in \partial\Omega, \end{aligned} \tag{2.7}$$

with $\mathcal{L}$ defined by (2.5), has the solution

$$u(x) = \mathrm{E}_{0,x} f(X(\tau_\Omega)), \tag{2.8}$$

where $X(t)$ satisfies (2.2). The expectation is a path integral. The stochastic process defined by (2.2) induces a probability measure on the space of continuous functions $\mathrm{C}(\mathbb{R}^+, \mathbb{R}^N)$ much like the Wiener measure. The function space can be thought of as a "path" space, where each path is a continuous function. The integrand of the path integral is $f(X(t))$, which is a functional of the path $X(t)$. In the following we see a more general integrand for the path integral, but it is still a functional of the path.

The more general elliptic PDE on domain $\Omega \subset \mathbb{R}^N$,

$$\begin{aligned} (-\mathcal{L}(x) + q(x))u(x) &= g(x), \quad x \in \Omega, \\ u(x) &= f(x), \quad x \in \partial\Omega, \end{aligned} \tag{2.9}$$

has the solution

$$\begin{aligned} u(x) = \mathrm{E}_{0,x}\Bigg(& f(X(\tau_\Omega)) \exp\left(-\int_0^{\tau_\Omega} \mathrm{d}t' q(X(t'))\right) \\ & + \int_0^{\tau_\Omega} \mathrm{d}t\, g(X(t)) \exp\left(-\int_0^t \mathrm{d}t' q(X(t'))\right)\Bigg). \end{aligned} \tag{2.10}$$

This variant Feynman–Kac formula appears in [2].

It is worth mentioning that a Neumann boundary condition, the reflecting boundary condition, can be treated easily by the bumping of a Brownian particle at the boundary.

The representation in terms of path integrals is motivated by the relative simplicity of computing them numerically. To do this, we generate paths or trajectories with the probability density governed by (2.2) with specified initial condition and then

compute the expectation. Generating trajectories requires discretizing (2.2) with a finite step size. Similar to the forward Euler method for ODEs, the simplest numerical method for SDEs is the Euler–Maruyama method (2.3), which we write as

$$X^{n+1} = X^n + a(X^n)\Delta t + \sqrt{\Delta t} B(X^n) Z^{n+1}, \tag{2.11}$$

where $X^n \approx X(t_n)$, $t_{n+1} = t_n + \Delta t$, and Z^{n+1} is a vector of M independent standard Gaussian random numbers. The term $\sqrt{\Delta t} Z^{n+1}$ realizes the Wiener increment $W(t^{n+1}) - W(t^n)$. In practice, the integration step size Δt can vary. Of course, there are many higher order numerical methods for generating trajectories, e.g., in the books by Milstein [13] and Kloeden and Platen [9]. See also the excellent review by Higham [6]. Better control of the random walk is possible using a lattice [2] or exponential time-stepping [8].

2.3. A reaction probability problem. The techniques of this article and their effectiveness are illustrated by the problem of calculating the probability of reaction between a substrate and an enzyme for a diffusion-limited reaction. This probability can be used to calculate the rate constant [14, 24].

Physics and geometry of the problem. The enzyme is modeled as a rigid body at the origin and the substrate is composed of N spheres. The $3N$ Cartesian coordinates of the centers of the spheres $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N$ describe a configuration of the substrate, which is represented by a column vector x of dimension $3N$. The substrate is treated as a *Brownian particle* in $3N$-dimensional space. Configurations of the substrate that are close to the enzyme in some prescribed sense constitute a *set of reacted configurations* $\Omega_{\rm rc}$, which is an open subset of $\mathbb{R}^{3N}$, whose boundary $\partial\Omega_{\rm rc}$ is the *reaction surface*. If the Brownian particle diffuses to the reaction surface, a reaction happens and the motion terminates. A *center of the substrate* $\mathbf{r}_{\rm c}$ is chosen as a possibly weighted average of $\mathbf{r}_i$'s. Let Ω_b and Ω_q be the subsets of $\mathbb{R}^{3N}$ containing the configurations with $|\mathbf{r}_{\rm c}| \le b$ and $|\mathbf{r}_{\rm c}| \le q$, respectively, where b and q are large enough so that $\Omega_{\rm rc} \subset \Omega_b \subset \Omega_q$. Their boundaries are called the *b-surface* and the *q-surface*, respectively. The q-surface is also called the *escape surface*. The particle's movement is in the domain $\Omega = \Omega_q \setminus \Omega_{\rm rc}$. A force vector $F(x)$ of dimension $3N$ and a $3N \times 3N$ symmetric positive definite diffusion tensor $D(x)$ are defined in the domain Ω. A function $f(x)$ is defined on the boundary $\partial\Omega$ of Ω,

$$f(x) = \begin{cases} 0 & \text{if } x \in \partial\Omega_q, \\ 1 & \text{if } x \in \partial\Omega_{\rm rc}. \end{cases} \tag{2.12}$$

Finally, $k_{\rm B}$ is the Boltzmann constant and T the ambient temperature.

Define $u(x)$ to be the probability that a particle starting from x reacts rather than escapes. The problem of calculating $u(x)$ has two equivalent formulations: the expectation of an exit value for an SDE and the solution of an elliptic PDE. The equivalence is based on that between (2.7) and (2.8) for the elliptic Dirichlet problem.

SDE definition of the reaction probability. The reaction probability $u(x)$ is the path integral

$$u(x) = \mathrm{E}_{0,x} f(X(\tau_\Omega)), \tag{2.13}$$

where $X(t)$ is a trajectory of the Brownian particle governed by the SDE [3]

$$\mathrm{d}X = \left((\nabla^{\rm T} D(X))^{\rm T} + D(X) \frac{F(X)}{k_{\rm B} T} \right) \mathrm{d}t + \sqrt{2}\, D_{1/2}(X) \mathrm{d}W(t), \tag{2.14}$$

with initial condition $X(0) = x$, τ_Ω is the *first exit time* from domain Ω, $\mathrm{E}_{0,x}$ is the expectation with respect to the probability law for the random trajectory $X(t)$, $D_{1/2}$ is a matrix satisfying $D_{1/2}D_{1/2}^{\mathrm{T}} = D$, and $W(t)$ is a $3N$-dimensional canonical Wiener process.

PDE definition of the reaction probability. The reaction probability $u(x)$ is the solution of the following boundary value problem for the steady state Smoluchowski equation:

$$\left(\nabla_x^{\mathsf{T}} D(x)\nabla_x + \frac{F(x)^{\mathsf{T}}}{k_{\mathrm{B}}T} D(x)\nabla_x\right) u(x) = 0, \qquad x \in \Omega$$
$$u(x) = f(x), \qquad x \in \partial\Omega. \tag{2.15}$$

When the force is conservative, a potential energy $U(x)$ can be defined such that $F(x) = -\nabla U(x)$, and (2.15) can be transformed to the quite different form

$$\exp\frac{U(x)}{k_{\mathrm{B}}T}\nabla_x^{\mathsf{T}} D(x) \exp\left(-\frac{U(x)}{k_{\mathrm{B}}T}\right)\nabla_x u(x) = 0. \tag{2.15$'$}$$

Average reaction probability on the *b*-surface. The formula for a rate constant is in terms of an average reaction probability on the b-surface. Given a distribution $p_b(x)$ of initial values x on the b-surface, the problem is to find the average $\bar{u}(b)$ of reaction probabilities $u(x)$ with respect to the distribution $p_b(x)$. The average $\bar{u}(b)$ is the expectation $\mathrm{E}f(X(\tau_\Omega))$, where $f(X(\tau_\Omega))$ is the exit value from domain Ω for a trajectory whose starting point x is distributed on the b-surface with probability density $p_b(x)$. This is illustrated in Figure 1.

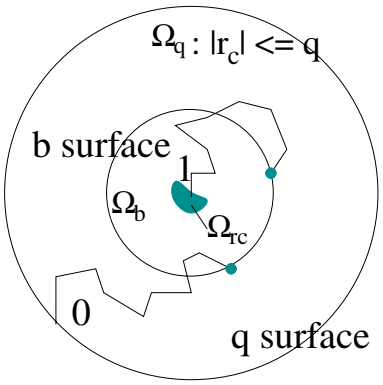

FIG. 1. *Trajectories begin on the b-surface and evolve until they reach either the reaction surface* Ω_{rc} *or the escape surface* Ω_q.

The density $p_b(x)$ [22] is such that the center $\mathbf{r}_{\mathrm{c}}$ has a uniform distribution on the $|\mathbf{r}_{\mathrm{c}}| = b$ spherical surface and the other degrees of freedom have a Boltzmann distribution.

Rather than solving the Dirichlet problem in (2.15) by a discretization method, we use the random walk method given by (2.13) and (2.14). In the following, a variance reduction method—biased BD—for the random walk method is derived and analyzed.

3. Variance reduction using biased BD. The biased BD realization of the Milstein importance sampling approach for (2.8) is derived here using a discrete formulation as a starting point. Also, comparatively elementary proofs are given for the fundamental results of Milstein.

3.1. Discrete biased BD. To motivate biased BD, we consider the reaction probability problem. Typically the probability of reaction $u(x)$ is very small, and a large number of trajectories will yield only a small number of reactions. Hence, the statistical error will be large relative to $u(x)$. The idea of biased BD is (i) to introduce an artificial bias force $F_{\rm b}(x)$ (in addition to the natural force $F(x)$) that increases the probability of reaction, and (ii) to divide the exit value by the factor by which the probability of the trajectory has been artificially inflated. Equivalently, the exit value is multiplied by a weight, which is the reciprocal of the probability inflation factor.

The discrete dynamics equation (2.11) is, again,

$$X^{n+1} = X^n + a(X^n)\Delta t + B(X^n)\sqrt{\Delta t}Z^{n+1},$$

and for the reaction probability problem

$$a(x) = (\nabla^{\rm T} D(x))^{\rm T} + D(x)\frac{F(x)}{k_{\rm B}T}, \qquad B(x) = \sqrt{2}D_{1/2}(x).$$

The addition of a bias force $F_{\rm b}(x)$ would add a term $D(x)F_{\rm b}(x)/(k_{\rm B}T)$ to the drift $a(x)$. It is useful, however, to consider the bias force as a modification to the random term and to write discrete biased BD as

$$X_{\rm b}^{n+1} = X_{\rm b}^n + a(X_{\rm b}^n)\Delta t + B(X_{\rm b}^n)\sqrt{\Delta t}\left(Z^{n+1} + \sqrt{\Delta t}c(X_{\rm b}^n)\right), \tag{3.1}$$

where for the reaction probability problem the "bias vector"

$$c(x) = D_{1/2}(x)^{\rm T} F_{\rm b}(x)/\left(\sqrt{2}k_{\rm B}T\right).$$

The calculation of the weight, which is to be discussed shortly, is actually valid for the more general discrete biased BD scheme

$$X_{\rm b}^{n+1} = X_{\rm b}^n + a(X_{\rm b}^n)\Delta t + B(X_{\rm b}^n)\sqrt{\Delta t}Z_{\rm b}^{n+1},$$

where $Z_{\rm b}^{n+1}$ is a random vector with an arbitrary PDF $p_{\rm b}(z)$ instead of a standard Gaussian $p_{\rm g}(z)$. The special case (3.1) corresponds to

$$Z_{\rm b}^{n+1} = Z^{n+1} + \sqrt{\Delta t}c(X_{\rm b}^n), \qquad p_{\rm b}(z) = p_{\rm g}\left(z - \sqrt{\Delta t}c(X_{\rm b}^n)\right). \tag{3.2}$$

Biased BD associates with each trajectory a weight, which is initially 1 and is adjusted as the trajectory develops to compensate for bias. Let $w^0 = 1$ and let w^n be the weight of a trajectory after the nth integration step. Consider the bias introduced by step $n+1$. A given value z of $Z_{\rm b}^{n+1}$ will be more probable than it should be by a factor $p_{\rm b}(z)/p_{\rm g}(z)$. Therefore, the weight of the trajectory is divided by $p_{\rm b}(Z_{\rm b}^{n+1})/p_{\rm g}(Z_{\rm b}^{n+1})$ at step $n+1$:

$$w^{n+1} = w^n p_{\rm g}(Z_{\rm b}^{n+1})/p_{\rm b}(Z_{\rm b}^{n+1}). \tag{3.3}$$

Thus, the product of the weight factor $p_{\rm g}(z)/p_{\rm b}(z)$ and the PDF $p_{\rm b}(z)$ for $Z_{\rm b}^{n+1}$ preserves a Gaussian distribution, as illustrated by Figure 2. By taking into account the weight for every integration step, the final statistical result of biased BD gives the same expectation as standard BD.

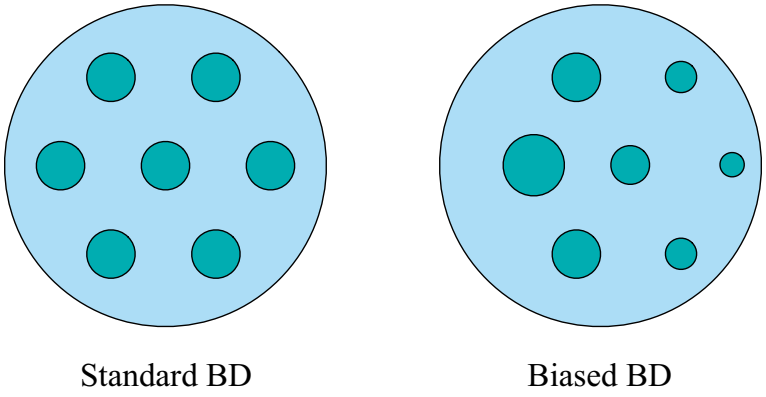

FIG. 2. *Biased BD weights the outcomes to compensate for the biased distribution.*

3.2. Continuous biased BD. Let the step size $\Delta t \to 0$, and (3.1) becomes the SDE

$$\mathrm{d}X_{\mathrm{b}}(t) = a\,(X_{\mathrm{b}})\,\mathrm{d}t + B\,(X_{\mathrm{b}})\,c\,(X_{\mathrm{b}})\,\mathrm{d}t + B\,(X_{\mathrm{b}})\,\mathrm{d}W(t). \tag{3.4}$$

Define $\Lambda^n = \ln w^n$, the logarithm of the current weight of the particle, and let $\Lambda(t)$ be the continuous extension of Λ^n under the zero step size limit. Equations (3.3) and (3.2) give

$$\begin{aligned}\Lambda^{n+1} - \Lambda^n &= \ln(p_{\mathrm{g}}(Z_{\mathrm{b}}^{n+1})/p_{\mathrm{b}}(Z_{\mathrm{b}}^{n+1})) \\ &= \frac{1}{2}\left((Z^{n+1})^{\mathrm{T}}Z^{n+1} - (Z^{n+1} + \sqrt{\Delta t}c(X_{\mathrm{b}}^n))^{\mathrm{T}}(Z^{n+1} + \sqrt{\Delta t}c(X_{\mathrm{b}}^n))\right) \\ &= -\frac{1}{2}c(X_{\mathrm{b}}^n)^{\mathrm{T}}c(X_{\mathrm{b}}^n)\Delta t - c(X_{\mathrm{b}}^n)^{\mathrm{T}}\sqrt{\Delta t}Z^{n+1}.\end{aligned}$$

Noting that $\sqrt{\Delta t}Z^{n+1} \to \mathrm{d}W(t)$ as $\Delta t \to \mathrm{d}t$, we have

$$\mathrm{d}\Lambda(t) = -\frac{1}{2}c(X_{\mathrm{b}})^{\mathrm{T}}c(X_{\mathrm{b}})\mathrm{d}t - c(X_{\mathrm{b}})^{\mathrm{T}}\mathrm{d}W(t). \tag{3.5}$$

Combining (3.4) and (3.5) gives

$$\begin{pmatrix} \mathrm{d}X_{\mathrm{b}} \\ \mathrm{d}\Lambda \end{pmatrix} = \begin{pmatrix} a\,(X_{\mathrm{b}}) + B\,(X_{\mathrm{b}})\,c\,(X_{\mathrm{b}}) \\ -\frac{1}{2}c(X_{\mathrm{b}})^{\mathrm{T}}c(X_{\mathrm{b}}) \end{pmatrix}\mathrm{d}t + \begin{pmatrix} B\,(X_{\mathrm{b}}) \\ -c(X_{\mathrm{b}})^{\mathrm{T}} \end{pmatrix}\mathrm{d}W(t). \tag{3.6}$$

Initially, $X_{\mathrm{b}}(0) = x$ and $\Lambda(0) = 0$.

In fact, (3.6) can be set as the starting point of biased BD.[1] The derivations in the following subsection are based on this equation. The most important theoretical results for biased BD are the following:

1. With a bias force present, the expectation of the suitably weighted exit value gives the unbiased estimate of the reaction probability.
2. There is an optimal bias force under which the variance of the weighted exit value is 0.

These results are stated more precisely as theorems in the following subsection.

[1] Milstein [13, Eq. (12.2)] uses w rather than Λ as the variable in the SDE. Equivalence of the two formulations can be shown using the Ito formula.

3.3. Variance of biased BD. The following theorem expresses the moments of the weighted exit value as solutions of elliptic boundary value problems. We apply this result to study the mean and the variance.

THEOREM 1. *Let $u_k(x)$ be the kth moment of the weighted exit value defined by*

$$u_k(x) = \mathrm{E}_{0,x}\left(f(X_\mathrm{b}(\tau_\Omega))\exp\Lambda(\tau_\Omega)\right)^k,$$

where $X_\mathrm{b}(t)$, $\Lambda(t)$ satisfies the biased BD (3.6), τ_Ω is the first exit time from domain $\Omega = \Omega_q \setminus \Omega_\mathrm{rc}$, and $f(x)$ is defined on $\partial\Omega$. Then $u_k(x)$ satisfies the PDE

$$\left(\mathcal{L}(x) - (k-1)c(x)^\mathsf{T}B(x)^\mathsf{T}\nabla_x + \frac{1}{2}(k-1)kc(x)^\mathsf{T}c(x)\right)u_k = 0, \qquad x \in \Omega,$$

with $\mathcal{L}(x)$ given by (2.5) and with boundary condition

$$u_k = f(x)^k, \qquad x \in \partial\Omega.$$

Proof. First, the boundary condition is satisfied because when a particle starts from a point x on the boundary, it exits immediately and the final weight is 1, which gives weighted exit value $f(x)^k$ on the boundary.

Second, we show that $u_k(x)$ satisfies the given PDE. Let $y = (x^\mathsf{T}, \lambda)^\mathsf{T}$, $Y(t) = (X_\mathrm{b}(t)^\mathsf{T}, \Lambda(t))^\mathsf{T}$, $f_k(y) = (f(x)\exp\lambda)^k$, and $\bar{u}_k(y) = \mathrm{E}_{0,y}f_k(Y(\tau_\Omega))$. Because of the Markov property of SDEs,

$$\mathrm{E}_{0,y}f_k(Y(\max\{\tau_\Omega, h\})) = \mathrm{E}_{0,y}\mathrm{E}_{h,Y(h)}f_k(Y(\max\{\tau_\Omega, h\})),$$

where $h > 0$ and $Y(t) = Y(\tau_\Omega)$ for $t \geq \tau_\Omega$. Equivalently,

$$\bar{u}_k(y) = \mathrm{E}_{0,y}\bar{u}_k(Y(h)).$$

So by (2.6)

$$0 = \lim_{h\downarrow 0}\frac{1}{h}\left(\mathrm{E}_{0,y}\bar{u}_k(Y(h)) - \bar{u}_k(y)\right) = \bar{\mathcal{L}}(y)\bar{u}_k(y), \tag{3.7}$$

where $\bar{\mathcal{L}}$ is the characteristic operator, (2.5), of (3.6),

$$\bar{\mathcal{L}}(x,\lambda) = \mathcal{L}(x) + c(x)^\mathsf{T}B(x)^\mathsf{T}\nabla_x(1-\partial_\lambda) - \frac{1}{2}c(x)^\mathsf{T}c(x)\partial_\lambda(1-\partial_\lambda), \tag{3.8}$$

with $\mathcal{L}$ as the characteristic operator of the original equation (2.2). Also, we can see from (3.6) that

$$\bar{u}_k(x,\lambda) = \mathrm{e}^{k\lambda}u_k(x). \tag{3.9}$$

The theorem follows from combining (3.7), (3.8), and (3.9). ∎

The preceding theorem with $k = 1$ shows that the expectation, $u(x) = u_1(x)$, of the weighted exit value $f(X_\mathrm{b}(\tau_\Omega))\exp\Lambda(\tau_\Omega)$ does not depend on the bias vector $c(x)$. This result is an assertion in Milstein's book [13, p. 131] based on Girsanov's theorem.

Also, this theorem with $k = 2$,

$$\left(\mathcal{L}(x) - c(x)^\mathsf{T}B(x)^\mathsf{T}\nabla_x + c(x)^\mathsf{T}c(x)\right)u_2 = 0, \tag{3.10}$$

characterizes the variance, $u_2(x) - u(x)^2$, of the weighted exit value, showing that it depends on the bias vector. So it may be possible to reduce the variance by selecting a suitable bias vector.

The following is equivalent to Theorem 12.1 in Milstein's book [13].

THEOREM 2. *Assume $f(x) > 0$, $x \in \partial\Omega$, and that (2.7) has a unique solution $u(x) > 0$, $x \in \Omega$. Then*

$$c(x) = B(x)^{\mathsf{T}} \nabla_x \ln u(x) \tag{3.11}$$

is the optimal bias vector and gives zero variance to the weighted exit value,

$$\mathrm{Var}_{0,x} f(X_{\mathrm{b}}(\tau_\Omega)) e^{\Lambda(\tau_\Omega)} = u_2(x) - u(x)^2 = 0.$$

Proof. Substitute formula (3.11) for the optimal bias vector into (3.10), and we have $\mathcal{L}_2 u_2(x) = 0$, where

$$\mathcal{L}_2 = \mathcal{L} - \frac{(\nabla u(x))^{\mathsf{T}}}{u(x)} B(x) B(x)^{\mathsf{T}} \nabla_x + \frac{(\nabla u(x))^{\mathsf{T}}}{u(x)} B(x) B(x)^{\mathsf{T}} \frac{(\nabla u(x))}{u(x)}.$$

Using (2.7), it can be shown that for any twice differentiable function $v(x)$

$$\mathcal{L}_2 uv = \mathcal{L} uv - (\nabla u)^{\mathsf{T}} B B^{\mathsf{T}} \nabla v = u \mathcal{L} v + v \mathcal{L} u = u \mathcal{L} v.$$

Define $v = u_2/u$ and we have

$$\mathcal{L} v = \frac{1}{u} \mathcal{L}_2 uv = \frac{1}{u} \mathcal{L}_2 u_2 = 0, \quad x \in \Omega, \qquad v = f, \quad x \in \partial\Omega.$$

This has a unique solution $v = u$, so $u_2 = uv = u^2$ under the optimal bias vector, and the variance $u_2(x) - u(x)^2$ is zero. ∎

To apply this result to the probability reaction problem, we use $B(x) = \sqrt{2} D_{1/2}(x)$ and $c(x) = D_{1/2}(x)^{\mathsf{T}} F_{\mathrm{b}}(x)/(\sqrt{2} k_{\mathrm{B}} T)$ and thus obtain

$$F_{\mathrm{b}}(x) = 2 k_{\mathrm{B}} T \nabla_x \ln u(x).$$

The effect of having $u(x) = 0$ on the q-surface is that $F_{\mathrm{b}}(x)$ becomes infinitely repelling as the q-surface is approached. Strictly speaking, Theorem 2 requires that the boundary values $f(x)$ be greater than 0. Generalization to the case where $f(x)$ vanishes or changes sign is given in [13, Remark 12.1]. Also given there is the generalization to the more general Feynman–Kac formula given by (2.10).

Remark. It is possible to derive the optimal bias vector from discrete biased BD (3.1). Let $u_{\Delta t}(x)$ be the expected exit value for unbiased discrete BD. By considering a single step, we get that

$$u_{\Delta t}(x) = \int \mathrm{d}z \, u_{\Delta t}\Big(x + a(x)\Delta t + B(x)\sqrt{\Delta t} z\Big) p_{\mathrm{g}}(z). \tag{3.12}$$

Let us postulate the existence of perfect PDFs $p_{\mathrm{b}}(z)$ which for a trajectory starting at x give an exit value that is $u_{\Delta t}(x)$ with probability 1. Then the expected exit value after one step should be the same as that before the step:

$$(p_{\mathrm{g}}(Z_{\mathrm{b}})/p_{\mathrm{b}}(Z_{\mathrm{b}})) u_{\Delta t}(x + a(x)\Delta t + B(x)\sqrt{\Delta t} Z_{\mathrm{b}}) = u_{\Delta t}(x),$$

where Z_{b} is a random variable with PDF $p_{\mathrm{b}}(z)$. It is, indeed, possible for this to hold identically for Z_{b} by choosing

$$p_{\mathrm{b}}(z) = p_{\mathrm{g}}(z) u_{\Delta t}(x + a(x)\Delta t + B(x)\sqrt{\Delta t} z)/u_{\Delta t}(x). \tag{3.13}$$

That this actually defines a PDF is ensured by (3.12). Equation (3.11) follows by requiring that (3.13) hold asymptotically as $\Delta t \to 0$ for $p_{\mathrm{b}}(z) = p_{\mathrm{g}}(z - \sqrt{\Delta t} c(x))$.

4. Weight control and optimal weight. A weight control mechanism is described in the WEBD paper [7] involving a target weight which is dynamically determined. Here we use a static target weight and a different weight control algorithm.

4.1. Ideal weight. The theorem that follows shows that the weight satisfies

$$\exp \Lambda(t) = u(x)/u(X_{\mathrm{b}}(t)) \tag{4.1}$$

under the optimal bias vector. Note that $\exp \Lambda(t)$ is a deterministic function of coordinates $X_{\mathrm{b}}(t)$ and does not depend on the path it traverses. We call it the *ideal weight*. The ideal weight is central to the weight control algorithm for biased BD discussed in the next subsection. (For another case, that of zero bias vector, the weight of a particle is also deterministic, since it is always 1.)

Generally, if the bias vector is not the optimal one, or Δt is finite as in a numerical integration, the weight of a particle depends on the path.

THEOREM 3. *Under the optimal bias vector,*

$$\Lambda(t) = \ln u(x) - \ln u(X_{\mathrm{b}}(t)), \tag{4.2}$$

where $X_{\mathrm{b}}(t)$, $\Lambda(t)$ *satisfy* (3.6).

Proof. At $t = 0$, (4.2) is obviously satisfied.

Hence, we need only prove

$$\mathrm{d}\Lambda(t) = -\mathrm{d} \ln u(X_{\mathrm{b}}(t)).$$

The characteristic operator, (2.5), for $X_{\mathrm{b}}(t)$ in (3.4) is

$$\mathcal{L}_{\mathrm{b}} = \mathcal{L} + c^{\mathsf{T}} B^{\mathsf{T}} \nabla = \frac{1}{2} B B^{\mathsf{T}} : \nabla \nabla^{\mathsf{T}} + a^{\mathsf{T}} \nabla + c^{\mathsf{T}} B^{\mathsf{T}} \nabla.$$

By the Ito formula, (2.4),

$$\begin{aligned} \mathrm{d} \ln u(X_{\mathrm{b}}(t)) &= \mathcal{L}_{\mathrm{b}}(X_{\mathrm{b}}(t)) \ln u(X_{\mathrm{b}}(t)) \mathrm{d}t + (\nabla_x \ln u(X_{\mathrm{b}}(t)))^{\mathsf{T}} B(X_{\mathrm{b}}(t)) \mathrm{d}W(t) \\ &= \left(-\frac{1}{2} \frac{(\nabla u)^{\mathsf{T}}}{u} B B^{\mathsf{T}} \frac{\nabla u}{u} + \frac{1}{u} \mathcal{L} u + c^{\mathsf{T}} B^{\mathsf{T}} \frac{\nabla u}{u} \right) \mathrm{d}t + \frac{(\nabla u)^{\mathsf{T}}}{u} B \, \mathrm{d}W(t). \end{aligned}$$

Substitute $\mathcal{L}u = 0$, which is obtained from (2.7), and $B^{\mathsf{T}} \nabla u / u = c$, which is obtained from (3.11), and we have

$$\mathrm{d} \ln u = \frac{1}{2} c^{\mathsf{T}} c \mathrm{d}t + c^{\mathsf{T}} \mathrm{d}W(t) = -\mathrm{d}\Lambda,$$

which completes the proof. ∎

4.2. Weight control. The ideal weight defined previously is a perfect choice for the target weight; however, it is not practical. In the next subsection, we formulate the target weight in terms of an approximate reaction probability. With a target weight for each configuration and a lower and an upper weight tolerance, the weight control algorithm performs in the following way: At the beginning of each time step, the particle's weight is checked.

1. If the weight falls below the range, its weight is doubled with 50% probability and the trajectory is terminated with 50% probability. A terminated trajectory contributes the value 0 to the expectation.

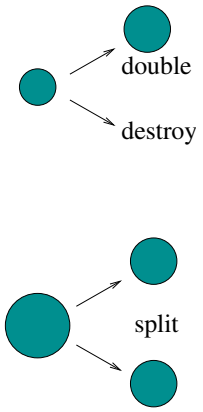

FIG. 3. *Mechanism for keeping weight within prescribed bounds.*

2. If the particle's weight is above the range, it splits into two particles, each with half the weight, and they are simulated separately. The final result is the sum of that for the two particles.
3. If the weight is within the range, the simulation continues.

This is illustrated in Figure 3. Thus the expectation of the outcome is unchanged in all cases. For each experiment, a particle could split many times, thus producing many reacted trajectories with different weights and many destroyed trajectories with zero weight. The weights of all reacted trajectories are summed to yield a single outcome for one experiment.

Without weight control, the particle's weight can grow without bound due to the imperfect choice of the bias vector and the approximation of the numerical integration method. With weight control, the weight of each Brownian particle is forced into a range. If the target weight is close to the ideal weight, a particle's weight is also close to the optimal weight.

In the weight control algorithm, the double or nothing strategy for a low weight particle halves the remaining simulation time while doubling the variance. The splitting strategy for a high weight particle doubles the remaining simulation time while halving the variance. The weight control algorithm is shown to be effective by the numerical tests, which indicates that the gain from reducing simulation time more than offsets the loss from increasing the variance for low weight particles, and the gain from reducing the variance more than offsets the loss from increasing the simulation time for high weight particles.

4.3. Construction of bias vector and target weight. A bias vector and a target weight are required by the algorithm. Of course, we do not know the optimal bias vector in (3.11) nor the ideal weight in (4.1). If we have an approximation of the reaction probability $u(x)$, we can construct the bias vector and target weight easily.

Motivated by WEBD is the use of a reaction coordinate $\xi(x)$, which we employ to construct an approximate solution having the form $\hat{u}(\xi(x))$. The reaction coordinate $\xi(x)$ measures nearness to reaction and is defined for every configuration. It is chosen so that $\xi(x) = \xi_{\text{rc}}$ exactly gives the reaction surface and $\xi(x) = \xi_q$ approximately gives the q-surface. To determine $\hat{u}(\xi))$ in the case of a conservative force, the Smoluchowski equation (2.15′) is expressed in variational form in terms of a functional of $u(x)$.

Minimization of this functional for the "test" function $u(x) \approx \hat{u}(\xi(x))$ yields the following formula [23, 22]:

$$\hat{u}(\xi) = \left(\int_{\xi_{\rm rc}}^{\xi_q} \mathrm{d}\xi' \rho(\xi')^{-1} \right)^{-1} \int_{\xi}^{\xi_q} \mathrm{d}\xi' \rho(\xi')^{-1},$$

where $\rho(\xi') = \int_{\Omega'} \mathrm{d}x\, \delta(\xi(x) - \xi')\ e^{-U(x)/(k_{\rm B}T)} (\nabla\xi(x))^{\mathsf{T}} D(x) \nabla\xi(x)$ is an integral on the manifold $\xi(x) = \xi'$. The density $\rho(\xi')$ might be chosen heuristically as illustrated in this article or by means of a Monte Carlo calculation as detailed in [23, 22].

The bias vector defined as in (3.11), but using $\hat{u}(\xi(x))$ for $u(x)$, might require numerical derivatives. The target weight logarithm $\lambda(x)$ according to (4.2) is

$$\lambda(x) = \ln \hat{u}(\xi(x^0)) - \ln \hat{u}(\xi(x)),$$

where x^0 is the starting point of a trajectory.

In the reaction probability problem, we use a different formula,

$$\lambda(x) = \ln \hat{u}_b - \ln \hat{u}(\xi(x)), \tag{4.3}$$

where $\hat{u}_b$ approximates an average of $\hat{u}(\xi(x))$ on the b-surface. The initial weight of a particle is still 1, while the initial target weight is $\hat{u}_b / \hat{u}(\xi(x^0))$. This change of target weight forces a particle to split or to be destroyed with some chance during the beginning of a trajectory based on the value $\hat{u}(\xi(x^0))$ at its starting point x^0. The target weight on the reaction surface is $\hat{u}_b$, which is independent of the starting point x^0. This target weight forces reacted particles to have about the same weight $\hat{u}_b$, which is proved to be effective by numerical experiments. The numerical experiments use a loose weight tolerance for the particle weight Λ^n,

$$-2 \le \Lambda^n - \lambda(X_{\rm b}^n) \le +1,$$

due to the very approximate nature of $\lambda(x)$.

5. Numerical testing. Here we briefly describe the rate constant computation for the enzyme–substrate system. A full description can be found in [22], and in [24] for the three-dimensional case.

The rate constant k is approximated using the domain truncation formula

$$\frac{1}{k} = \frac{1}{k_{\rm f}} + \int_q^{+\infty} \mathrm{d}r \frac{\exp\left(U_{\rm ext}(r)/(k_{\rm B}T)\right)}{4\pi r^2 d(r)},$$

and the Northrup–Allison–McCammon formula [14]

$$\frac{1}{k_{\rm f}} = \frac{1}{\bar{u}(b)} \int_b^q \mathrm{d}r \frac{\exp\left(U_{\rm ext}(r)/(k_{\rm B}T)\right)}{4\pi r^2 d(r)},$$

where r is the distance between the substrate and the enzyme, $b < q$ are two values of r, $d(r)$ is the relative diffusion coefficient between the pair of molecules, $U_{\rm ext}(r)$ is the intermolecular potential energy when the distance is large, and $\bar{u}(b)$ is the average reaction probability on the b-surface, as defined in section 2.3.

5.1. *E. coli* SOD with and without weight control. The enzyme CuZn SOD converts toxic O_2^- ions to oxygen and hydrogen peroxide. Examined here is *E. coli* SOD 1eso, whose coordinates are from the Protein Data Bank (http://www.rcsb.org/pdb). Partial charges for atomic positions of SOD are from the standard CHARMM force field [1] incorporated into UHBD. The substrate O_2^- is modeled by a sphere of radius 1.5 Å and a charge of $-$e, which does BD in ionized solvent with SOD fixed in space. The electrostatic potential surrounding SOD is obtained from a numerical solution of the Poisson–Boltzmann equation on a three-dimensional grid. The atoms of the substrate and enzyme are modeled as nonpenetrating hard spheres of specified radius. Reaction is said to occur if O_2^- is within 7 Å of the copper atom of SOD. When O_2^- has a hard sphere contact with a nonreacting region of SOD, one simply retries the integration step with another Gaussian distributed random vector. The diffusion tensor is

$$D = \left(\frac{k_{\mathrm{B}}T}{6\pi\eta a_1} + \frac{k_{\mathrm{B}}T}{6\pi\eta a_{\mathrm{mol}}} \right) I,$$

where temperature $T = 300\,\mathrm{K}$, water viscosity $\eta = 0.89\,\mathrm{g\,m^{-1}s^{-1}}$, the hydrodynamic radius of O_2^- $a_1 = 2.05$ Å, and that of SOD $a_{\mathrm{mol}} = 25$ Å.

The simulations compute the average reaction probability on the b-surface and then use the NAM formula with $b = 80$ Å and $q = 400$ Å to obtain the rate constant. The origin is chosen to be the geometric center of the atoms of SOD. Biased BD uses for a reaction coordinate ξ the distance between the O_2^- ion and the copper atom of SOD. The bias force and target weight are constructed as described in section 4.3. A heuristic choice for the density function based on an analytical solution of (2.15) for three-dimensional pure diffusion is given by $\rho(\xi) = \xi^2$, for which $\hat{u}(\xi) = (1/\xi - 1/\xi_q)/(1/\xi_{\mathrm{rc}} - 1/\xi_q)$, where $\xi_{\mathrm{rc}} = 7$ Å and $\xi_q = 405.26545$ Å. An approximate solution $\hat{u}(\xi)$ is also computed with the Monte Carlo method described in [23, 22].

Results are given in Table 1. The value t_{needed} is the performance metric; it is an estimate of the CPU hours needed by a Pentium III 1GHz machine needed to attain a statistical error of 5% with 95% confidence. The variance for N_{trials} trajectories is estimated to be $\mathrm{Var}\zeta/N_{\mathrm{trials}}$, where $\mathrm{Var}\zeta$ is a calculated estimate of the variance of the average reaction probability $\bar{u}(b)$. With 95% confidence the calculated estimate $\mathrm{E}\zeta$ is in error by less than $1.96 \times \sqrt{\mathrm{Var}\zeta/N_{\mathrm{trials}}}$. Assume N_{trials} is chosen so that this is equal to $0.05 \times \bar{u}(b)$, where $\bar{u}(b)$ represents the best estimate obtained by any means possible. Then

$$t_{\mathrm{needed}} = \left(\frac{1.96}{0.05} \right)^2 \times \frac{\mathrm{Var}\zeta}{\bar{u}(b)^2} \times t_{\mathrm{step}} \times \bar{n}_{\mathrm{steps}}/3600, \tag{5.1}$$

where t_{step} is the CPU seconds per integration step of the SDE and $\bar{n}_{\mathrm{steps}}$ is the average number of integration steps per trajectory.[2] The value $\bar{u}(b)$ in the formula is obtained as an average of the last four $\mathrm{E}\zeta$ values with weights proportional to the reciprocals of the corresponding four $\mathrm{Var}\zeta$ values. The value N_{trials} in the table is the number of trajectories started from the b-surface, N_{destroy} is the number of times a trajectory is terminated, N_{split} is the number of times a trajectory is split, and the value t_{actual} is the actual unnormalized total CPU time. The cost of computing the Monte Carlo approximation of the densities is 11 minutes of the CPU time.

[2] For unbiased BD, $\bar{u}(b) - \bar{u}(b)^2$ is used in place of $\mathrm{Var}\zeta$.

TABLE 1

Rate constant and computing cost for SOD 1eso with and without weight control. heur: density function $\rho(\xi) = \xi^2$. mc: density function from Monte Carlo simulation. bias: biased BD without weight control. bswt: biased BD with weight control.

	Unbiased	heur.bias	heur.bswt	mc.bias	mc.bswt
t_{step}(s)	2.58×10^{-6}	5.50×10^{-6}	6.91×10^{-6}	5.59×10^{-6}	6.83×10^{-6}
$\bar{n}_{\text{steps}}$	1.96×10^{5}	6.14×10^{5}	1.01×10^{5}	1.57×10^{5}	8.78×10^{4}
N_{trials}	200000	200000	200000	200000	200000
N_{destroy}	N/A	N/A	187408	N/A	215294
N_{split}	N/A	N/A	13	N/A	55130
t_{actual}(h)	28.2	187.8	38.9	48.7	33.3
Eζ	1.73×10^{-3}	1.74×10^{-3}	1.74×10^{-3}	1.74×10^{-3}	1.75×10^{-3}
Varζ	1.73×10^{-3}	6.86×10^{-5}	7.20×10^{-5}	1.46×10^{-4}	4.16×10^{-5}
$k(\text{M}^{-1}\text{s}^{-1})$	1.71×10^{8}	1.71×10^{8}	1.71×10^{8}	1.72×10^{8}	1.72×10^{8}
$\Delta k(\text{M}^{-1}\text{s}^{-1})$	1.79×10^{7}	3.57×10^{6}	3.66×10^{6}	5.21×10^{6}	2.78×10^{6}
t_{needed}(h)	124.4	32.9	7.1	18.0	3.5

The ratio of N_{split} to $2N_{\text{destroy}}$ is highly unbalanced for the heuristic bias force, indicating a poor approximate solution obtained heuristically. As a result the weight control forces a particle's weight into an improper range and does not reduce the variance; however, it does reduce the average number of integration steps per trajectory significantly.

The rate constant of bovine *B. taurus* and shark *P. glauca* SOD determined experimentally is $3.92 \times 10^9\ \text{M}^{-1}\text{s}^{-1}$ [17] compared to a calculated value of $5 \times 10^9\ \text{M}^{-1}\text{s}^{-1}$ [18].

5.2. Model protein with and without weight control. The model problem discussed here originates in [15] and is later used [7] as a test problem to show the speedup of the WEBD algorithm. In the solvent there are two types of proteins, each a sphere of radius 18 Å. To define a reaction condition, imagine that the center of a 17 Å × 17 Å square is attached tangentially to each sphere with vertices labeled A, B, C, and D. Reaction occurs if at least 3 of the 4 pairs of vertices A_1A_2, B_1B_2, C_1C_2, D_1D_2 are within 2 Å.

The NAM formula is used with $b = 80$ Å and $q = 400$ Å to obtain the rate constant. The reaction coordinate ξ is defined as the third smallest distance between 4 pairs of vertices, which is the same as that used in [7]. Configuration space is six-dimensional, and near the reaction boundary the volume of configuration space $\rho(\xi)$ as a function of ξ is proportional to ξ^5. However, the 3 orientational degrees of freedom are limited in their range, and once ξ gets large enough this range is exhausted and further growth in $\rho(\xi)$ is proportional to ξ^2. For a heuristic density function we choose $\text{d}\ln\rho(\xi)/\text{d}\ln\xi = a(\xi)$, where $a(\xi)$ is a step function that takes the value 5 for $2 < \xi < 30$, the value 4 for $30 < \xi < 40$, the value 3 for $40 < \xi < 50$, and the value 2 for $50 < \xi < 600$. Also tested is a density function from a Monte Carlo simulation.

Complete details of this test problem are lengthy and the reader is referred to [22, 23] and to the references therein. Also, the problem described here is outside the class of problems defined in section 2.3, but the modifications are fairly straightforward.

Table 2 shows the test results, where the parameters have the same meaning as before. The starred entries for unbiased BD are theoretical estimates [7, 23].

TABLE 2

Rate constant and computing cost for model protein with and without weight control. heur: density function $\mathrm{d}\ln\rho(\xi)/\mathrm{d}\ln\xi = 2,\ldots,5$. *mc: density function from Monte Carlo simulation. bias: biased BD without weight control. bswt: biased BD with weight control.*

	Unbiased	heur.bias	heur.bswt	mc.bias	mc.bswt
$t_{\mathrm{step}}(\mathrm{s})$	4.49×10^{-6}	7.69×10^{-6}	8.85×10^{-6}	7.68×10^{-6}	8.83×10^{-6}
$\bar{n}_{\mathrm{steps}}$	$3.15\times10^{5}*$	1.94×10^{5}	1.48×10^{5}	1.37×10^{5}	1.33×10^{5}
N_{trials}		200000	200000	200000	200000
N_{destroy}	N/A	N/A	542762	N/A	566803
N_{split}	N/A	N/A	501488	N/A	504419
$t_{\mathrm{actual}}(\mathrm{h})$		82.8	73.0	58.4	65.3
$\mathrm{E}\zeta$		4.95×10^{-6}	8.66×10^{-6}	5.48×10^{-6}	8.52×10^{-6}
$\mathrm{Var}\zeta$		6.41×10^{-8}	4.06×10^{-9}	4.92×10^{-8}	3.40×10^{-9}
$k(\mathrm{M}^{-1}\mathrm{s}^{-1})$		1.05×10^{5}	1.83×10^{5}	1.16×10^{5}	1.80×10^{5}
$\Delta k(\mathrm{M}^{-1}\mathrm{s}^{-1})$		2.35×10^{4}	5.91×10^{3}	2.06×10^{4}	5.41×10^{3}
$t_{\mathrm{needed}}(\mathrm{h})$	70487.1 *	1667.3	30.4	735.3	23.5

The expectations are low for biased BD without weight control. Moreover, the confidence intervals are disjoint. Perhaps there should be trajectories with larger weights that do not materialize for biased BD without weight control.

REFERENCES

[1] B. R. BROOKS, R. E. BRUCCOLERI, B. D. OLAFSON, D. J. STATES, S. SWAMINATHAN, AND M. KARPLUS, CHARMM: *A program for macromolecular energy, minimization, and dynamics calculations*, J. Comput. Chem., 4 (1983), pp. 187–217.

[2] F. M. BUCHMANN AND W. P. PETERSEN, *Solving Dirichlet problems numerically using the Feynman-Kac representation*, BIT Numer. Math., 43 (2003), pp. 519–540.

[3] D. L. ERMAK AND J. A. MCCAMMON, *Brownian dynamics with hydrodynamic interactions*, J. Chem. Phys., 69 (1978), pp. 1352–1360.

[4] C. W. GARDINER, *Handbook of Stochastic Methods*, 2nd ed., Springer-Verlag, Berlin, 1985.

[5] P. GRASSBERGER, *"Go with the Winner" Algorithms*, invited presentation, Workshop on Multiscale Computational Methods in Chemistry, Eilat, Israel, 2000.

[6] D. J. HIGHAM, *An algorithmic introduction to numerical simulation of stochastic differential equations*, SIAM Rev., 43 (2001), pp. 525–546.

[7] G. A. HUBER AND S. KIM, *Weighted-ensemble Brownian dynamics simulations for protein association reactions*, Biophys. J., 70 (1996), pp. 97–110.

[8] K. M. JANSONS AND G. D. LYTHE, *Efficient numerical solution of stochastic differential equations using exponential timestepping*, J. Statist. Phys., 100 (2000), pp. 1097–1109.

[9] P. E. KLOEDEN AND E. PLATEN, *Numerical Solution of Stochastic Differential Equations*, Appl. Math. (N.Y.) 23, Springer-Verlag, New York, 1992.

[10] M. MELCHIOR AND H. ÖTTINGER, *Variance reduced simulations of stochastic differential equations*, J. Chem. Phys., 103 (1995), pp. 9506–9509.

[11] M. MELCHIOR AND H. ÖTTINGER, *Variance reduced simulations of polymer dynamics*, J. Chem. Phys., 105 (1996), pp. 3316–3331.

[12] G. MILSTEIN AND J. SCHOENMAKERS, *Monte Carlo construction of hedging strategies against multi-asset European claims*, Stochastics Stochastics Rep., 73 (2002), pp. 125–157.

[13] G. N. MILSTEIN, *Numerical Integration of Stochastic Differential Equations*, Math. Appl. 313, Kluwer Academic, Dordrecht, The Netherlands, 1995.

[14] S. H. NORTHRUP, S. A. ALLISON, AND J. A. MCCAMMON, *Brownian dynamics simulation of diffusion-influenced bimolecular reactions*, J. Chem. Phys., 80 (1984), pp. 1517–1524.

[15] S. H. NORTHRUP AND H. P. ERICKSON, *Kinetics of protein–protein association explained by Brownian dynamics computer simulation*, Proc. Natl. Acad. Sci. USA, 89 (1992), pp. 3338–3342.

[16] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, 5th ed., Springer-Verlag, Berlin, 1998.

[17] F. Polticelli, M. Falconi, P. O'Neill, R. Petruzelli, A. Galtieri, A. Lania, L. Calabrese, G. Rotillio, and A. Desideri, *Molecular modeling and electrostatic potential calculations on chemically modified Cu, Zn superoxide dismutases from Bos taurus and shark Prionace glauca: Role of Lys134 in electrostatically steering the substrate to the active site*, Arch. Biochem. Biophys., 312 (1994), pp. 22–30.

[18] A. Rojnuckarin, D. R. Livesay, and S. Subramaniam, *Bimolecular reaction simulation using weighted ensemble Brownian dynamics and the University of Houston Brownian dynamics program*, Biophys. J., 79 (2000), pp. 686–693.

[19] Z. Schuss, *Theory and Applications of Stochastic Differential Equations*, John Wiley and Sons, New York, 1980.

[20] W. Wagner, *Unbiased Monte Carlo evaluation of certain functional integrals*, J. Comput. Phys., 71 (1987), pp. 21–33.

[21] W. Wagner, *Monte Carlo evaluation of functionals of solutions of stochastic differential equations. variance reduction and numerical examples*, Stochastic Anal. Appl., 6 (1988), pp. 447–468.

[22] G. Zou, *Robust Biased Brownian Dynamics for Rate Constant Calculation*, Ph.D. thesis, University of Illinois at Urbana-Champaign, 2002; Department of Computer Science Report 2294, University of Illinois at Urbana-Champaign, 2002; available online at http://www.cs.uiuc.edu/research/reports.html.

[23] G. Zou and R. D. Skeel, *Robust biased Brownian dynamics for rate constant calculation*, Biophys. J., 85 (2003), pp. 2147–2157; also available online at http://bionum.cs.uiuc.edu/p.html.

[24] G. Zou, R. D. Skeel, and S. Subramaniam, *Biased Brownian dynamics for rate constant calculation*, Biophys. J., 79 (2000), pp. 638–645.

[25] D. M. Zuckerman and T. B. Woolf, *Dynamic reaction paths and rates through importance-sampled stochastic dynamics*, J. Chem. Phys., 111 (1999), pp. 9475–9484.