ADAPTIVE MULTILEVEL SPLITTING IN MOLECULAR DYNAMICS SIMULATIONS

David Aristoff, Tony Lelièvre, Christopher G. Mayne, Ivan Teo

HAL Id: hal-00959109
https://hal.inria.fr/hal-00959109

Submitted on 17 Mar 2014

ADAPTIVE MULTILEVEL SPLITTING IN MOLECULAR DYNAMICS SIMULATIONS [*], [**]

David Aristoff[1], Tony Lelièvre[2], Christopher G. Mayne[3] and Ivan Teo[4]

Abstract. Adaptive Multilevel Splitting (AMS) is a replica-based rare event sampling method that has been used successfully in high-dimensional stochastic simulations to identify trajectories across a high potential barrier separating one metastable state from another, and to estimate the probability of observing such a trajectory. An attractive feature of AMS is that, in the limit of a large number of replicas, it remains valid regardless of the choice of reaction coordinate used to characterize the trajectories. Previous studies have shown AMS to be accurate in Monte Carlo simulations. In this study, we extend the application of AMS to molecular dynamics simulations and demonstrate its effectiveness using a simple test system. Our conclusion paves the way for useful applications, such as molecular dynamics calculations of the characteristic time of drug dissociation from a protein target.

Introduction

In high-dimensional stochastic systems, one commonly encounters potential or entropic barriers so high that observing even a single crossing event would require a prohibitive amount of computational effort. The problem of sampling such rare events is of particular interest in molecular dynamics (MD), which involves extremely many degrees of freedom and requires very small time steps so that events of interest are frequently out of reach for equilibrium simulations. For example, the dissociation of small molecules from a substrate can have characteristic times on the order of seconds, compared to the usual MD time step size of 1 or 2 fs.

Various sampling techniques have been developed to mitigate high potential barriers between metastable states, allowing the sampling of trajectories that cross these barriers. One such technique is Adaptive Multilevel Splitting (AMS) [2,3], which has been successfully tested on Monte Carlo simulations. Thus, one interest of the current work is to test for the first time the validity of AMS applied to phase space Langevin dynamics. Unlike many of its peer techniques, AMS does not require prior knowledge of intermediate states and is valid for any reaction coordinate that describes the transition from the predefined initial and final states. Due to its versatility, AMS has promising prospects in biological MD simulations, where prior information about trajectories is often lacking due to experimental difficulties or sheer complexity of the system.

[1] Department of Mathematics, University of Minnesota, USA

[2] CERMICS, École des Ponts ParisTech, France

[3] Beckman Institute, University of Illinois at Urbana-Champaign, USA

[4] Beckman Institute, University of Illinois at Urbana-Champaign, USA

We applied the AMS algorithm to a simple MD test case involving an ion in a harmonic well, using the software NAMD [1], developed by the Theoretical and Computational Biophysics Group in the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign. In order to implement the algorithm, alterations to the current released version (2.9) of the software were necessary, as described herein. Direct long-time simulations of the test case, as well as an analytic derivation of key physical quantities, were performed along with AMS simulations for comparison. We found that AMS faithfully yielded results in agreement with measurements from the direct simulations and values obtained from the analytic calculation. In one of the simulations, AMS was able to sample events that were too rare for direct simulation to reproduce, and at a measured rate that agreed with the analytic calculation using a simple model. This study is an indication that AMS is a feasible sampling method in MD simulations, and is a precursor to simulations of more complicated biological systems and efforts to streamline the software for better efficiency.

After a brief description of the AMS algorithm, we present the software modifications which were necessary to implement AMS on NAMD. This is an opportunity to discuss the complexity of the algorithm, in terms of processor time and memory requirements. Finally, we discuss numerical results from the test simulations. The appendix is devoted to analytic computations on a simple model for the sake of comparison with the numerical results.

Description of AMS

Consider $(X_t)_{t \geq 0}$ a Markov process in $\mathbb{R}^d \setminus (A \cup B)$. Assume that the process follows almost surely continuous paths and is ergodic with respect to some equilibrium measure μ. In our simulations, X_t is the vector of current positions and velocities of a molecular system obeying Langevin dynamics, and μ is the Boltzmann distribution. The aim of AMS is to understand *reactive trajectories*, that is, paths of the process which start in some set $A \subset \mathbb{R}^d$ and then reach another set $B \subset \mathbb{R}^d$ without first returning to A. Define a *reaction coordinate* $\xi : \mathbb{R}^d \to \mathbb{R}$ which measures progress from A to B. Thus, it can be written

$$
\begin{aligned}
A &= \{x \in \mathbb{R}^d \,:\, \xi(x) \leq z_A\} \\
B &= \{x \in \mathbb{R}^d \,:\, \xi(x) \geq z_B\}
\end{aligned}
\tag{1}
$$

for some $z_A < z_B$. We assume ξ is differentiable and $\nabla \xi(x) \neq 0$ for all $x \in \mathbb{R}^d \setminus (A \cup B)$, so that the level sets of ξ are submanifolds of $\mathbb{R}^d \setminus (A \cup B)$. We will refer to the submanifold $\{x \in \mathbb{R}^d \,:\, \xi(x) = z\}$ as the *level z*.

The AMS algorithm uses N replicas of the process along with a splitting technique to efficiently generate reactive paths. The algorithm, rephrased from [3], is as follows:

Algorithm 1. *Pick a number N of replicas, and choose z_{min} between z_A and z_B. Set the total number of branching points, M, to zero: $M = 0$. Then iterate the following:*
Initialization Step:
1. *Generate N independent samples of the (normalized) equilibrium distribution $\mu(A)^{-1}\mu|_A$ restricted to A.*
2. *Starting at these N sample points, evolve N replicas independently until each replica reaches the level z_{min}.*
3. *Continue evolving the N replicas until each replica reaches A or B.*
Branching Step:
4. *Let z_k be the maximum level reached by replica k, and let*

$$
\begin{aligned}
k^* &= \arg \min_{1 \leq k \leq N} z_k \\
z^* &= \min_{1 \leq k \leq N} z_k.
\end{aligned}
$$

5. *Kill the existing k^*th replica. Pick another replica uniformly at random, say replica k'. Let $x \in \mathbb{R}^d$ be the point at which replica k' first reaches the level z^*.*

6. *Start a new k^*th replica at the point x, and evolve the replica until it reaches either A or B. Update the total number of branching points: $M = M + 1$.*
7. *If all the replicas have reached B, stop. Otherwise, return to Step 4.*

The parameter $z_{\min}$ is chosen so that the time for the process to go from z_A to $z_{\min}$ is on average relatively small, while at the same time a replica starting at $z_{\min}$ can evolve for some time without immediately returning to z_A.

Future work may take advantage of parallelism by using a variant of AMS [2] in which $i \geq 1$ replicas are killed, as explained in the following remark:

Remark 1. *In the Branching Step, instead of killing and restarting just one replica, we can instead choose the i lowest z_k values and kill the corresponding replicas. Each of the killed replicas would then be restarted at a randomly chosen surviving replica, as per Step 6. It has been shown that the $i = 1$ case produced the best results [4]; however, the gain in computational speed from running i trajectories in parallel at each iteration may justify such a compromise in accuracy.*

Let p be the probability that, starting at the level $z_{\min}$ as in Step 2 above, the process reaches B before returning to A. After the algorithm is complete, we have the following estimate p_{AMS} of p:

$$p_{\mathrm{AMS}} := \left(1 - \frac{1}{N}\right)^M \tag{2}$$

where by definition M is the total number of branching points. Under ideal conditions, p_{AMS} is an unbiased estimate of p [4, 5], and moreover, the variance of p_{AMS}, $\mathrm{Var}(p_{\mathrm{AMS}})$, is, asymptotically as $N \to \infty$ [4, 6]:

$$\mathrm{Var}(p_{\mathrm{AMS}}) \sim \frac{-p_{\mathrm{AMS}}^2 \ln p_{\mathrm{AMS}}}{N}. \tag{3}$$

At this point, it is worth cautioning that the estimators p_{AMS} and $\mathrm{Var}(p_{\mathrm{AMS}})$ given above were derived on the assumption that the N trajectories obtained are completely independent. This is not true in the case of AMS, since trajectories are branched from one another. Thus, one should be careful in the interpretation of these estimators. One possible way to check the accuracy of the estimators is to run the AMS algorithm several times and inspect the variation in the results obtained.

AMS can also be used to estimate τ, the average time to go from ∂A to B, which is done as follows. Let T_1 be the average time for a replica in the Initialization Step to go from the level z_A to the level $z_{\min}$; similarly let T_2 be the average time for a replica in the Initialization Step to go from $z_{\min}$ to z_A without reaching z_B. T_1 and T_2 can be sampled from a direct simulation. Let T_3 be the average time for the process to go from the level $z_{\min}$ to z_B without ever returning to z_A; T_3 can be estimated by averaging over the reactive trajectories at the end of the algorithm. We then obtain the following estimate τ_{AMS} of τ:

$$\tau_{\mathrm{AMS}} := \frac{T_1 + T_2}{p} + T_1 + T_3. \tag{4}$$

See [3] for details and a discussion.

Software Modifications

To establish the feasibility of applying the AMS methodology to complex MD simulations, the initial implementation described herein is largely focused on resolving several technical challenges, namely, rapid prototype development, simulation scalability, data management, and method parallelization. A general outline of the AMS implementation and organization is shown schematically in Fig. 1. Under this scheme, the AMS control code works in conjunction with NAMD to set up, run, and analyze each simulation step. A pool of NAMD

instances utilize a shared file system to allow cross-process communication, enabling dynamic initialization and termination of simulations as guided by the AMS control logic.

As the centerpiece of the described implementation, the NAMD software package [1] was chosen for its highly efficient, scalable, and feature-rich MD engine that can run on a variety of platforms and is maintained at most supercomputing centers around the world. Two NAMD features of primary importance to AMS are the native "Colvars" implementation [7] and the embedded Tcl interpreter. The colvars module is leveraged to define and monitor the AMS reaction coordinate using the built-in collective variables, which are easily defined and can be combined to describe complex reaction coordinates. Access to the Tcl scripting interface, a feature unique to NAMD, allows a great deal of flexibility to rapidly develop and debug the AMS control code without requiring detailed knowledge of the internal workings of NAMD. Separating the AMS control from the NAMD "black box" requires only minor modifications to the NAMD source code–adding mechanisms for reading/writing of restart data and sequential trajectory files. Although these features were added specifically to enable the AMS method as described herein, they are of general utility to NAMD users and will be included in the NAMD 2.10 software release.
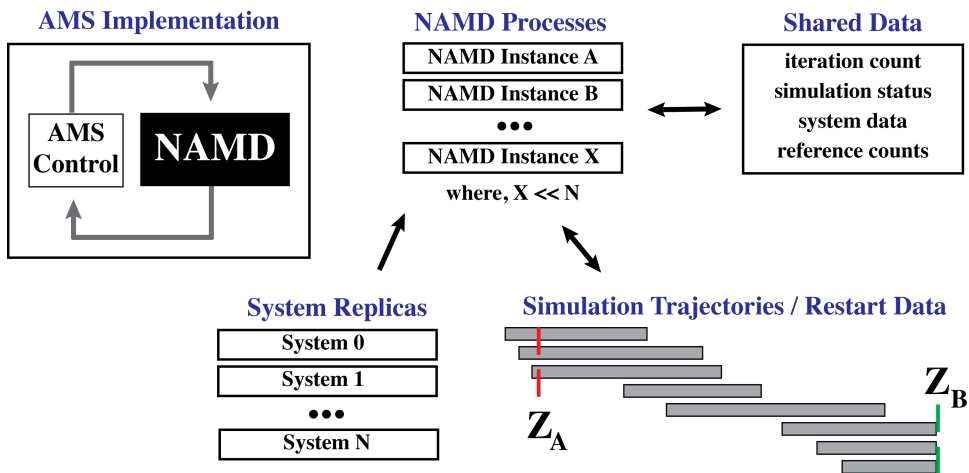

FIGURE 1. Schematic of AMS implementation using a file system to control multiple NAMD instances while monitoring data generated by the simulations.

From the outset, coupling the AMS methodology to MD simulations raised significant concerns regarding data management; MD is data-intensive (generates large positional trajectories and restart files), while AMS is highly duplicative (reactive trajectories are "copied" up to the branch point). Accordingly, the storage requirements of an AMS run are mitigated using two concurrent approaches. The first approach is centered on minimizing the number of simulation restart files that are stored. As described in the Branching Step of the algorithm, simulations are restarted at the *first* crossing of a particular isosurface of the reaction coordinate. In practical terms, this specification restricts potential restarts to frames in which the z-value is a new maximum observed value, as shown in Fig. 2. Each set of restart data (position, velocity, periodic cell) is stored using the frame number, preserving the timing information of the simulation. The largest reduction in storage requirements can be realized by suppressing the positional trajectory output (DCD files) without compromising the kinetic analysis (p_{AMS}, τ_{AMS}), albeit at the loss of continuity in the atomic detail for each reactive trajectory.

The second approach employs reference counting, a computer science framework for managing objects in memory, to curate the simulation data. As implied in Fig. 1, the simulation data (trajectories, restart data) are stored and manipulated as a collection of individual files. Each system replica maintains an ordered list of references to frames within a particular simulation file that, when stitched together, defines the reactive trajectory. The number of references to each file are maintained as part of the shared data, and reference counts

are incremented or decremented as trajectories are duplicated or discarded, respectively. When the reference count for a particular file reaches zero, the data is no longer relevant to any of the surviving reactive trajectories, and therefore, deleted from the file system. Using this framework both duplicate and discarded simulation data are minimized.
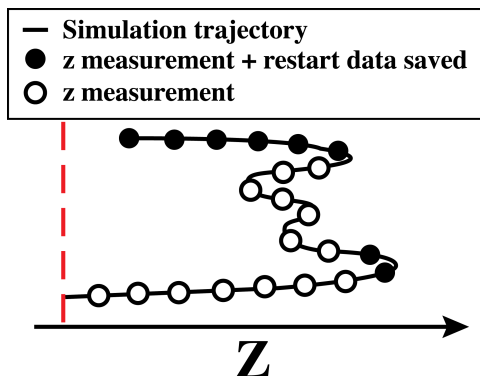

FIGURE 2. Illustration of a typical trajectory, with circles representing points at which the reaction coordinate z is measured. In the interest of data economy, saving to memory occurs only as required for the reconstruction of the reactive path, *i.e.* at the black circles.

Finally, a critical component of the current implementation is the "pseudo-parallelization" of the AMS method. The original codes used demonstrate the AMS method for 1D and 2D potential energy surfaces operated in a purely serial mode, waiting for each simulation to reach a termination criterion (return to state A or advance to state B) before starting a new replica (as in Step 6 of the algorithm). In the current implementation, we instead start a new replica as soon as the smallest maximum level among all running replicas surpasses the smallest maximum level among all stopped replicas. Thus, many replicas can be running concurrently in parallel. Fig. 3 depicts this strategy in which the least advanced trajectory, shown in green, represents a currently running replica. Once this replica has surpassed the threshold demarcated by the orange line, the least-progressed stopped replica (blue), can be restarted. Although the degree of parallelism and replica start times are unpredictable (an outcome of the stochastic MD process), in our experience, this design allows for significant parallelization in two key areas: as simulations rapidly progress along the reaction coordinate in areas of low energy, and when simulations make any degree of progress along the reaction coordinate (*e.g.*, high energy) but require a non-trivial amount of time to return to state A.

SIMULATIONS AND RESULTS

The test system consists of a 50 Å $\times$ 50 Å $\times$ 50 Å box of explicit water, ionized with 0.15 M potassium chloride such that the net charge is zero. We pick one particular K^+ ion, positioned initially at the origin. Henceforth, we refer only to the chosen ion and omit all other ions from our discussion. Our objective is to use AMS to evaluate the characteristic time taken for the ion to migrate from a point of distance z_A from the origin to a point z_B away from the origin, under the influence of a harmonic well potential centered on the origin. For this purpose, we identify the reaction coordinate z with the distance from the origin r.

CHARMM parameters for ion interactions were taken from Roux and coworkers [9] while the water molecules were characterized by the TIP3P water model [8]. Simulations were run with 1-fs time steps. Long range electrostatic forces were calculated using the particle mesh-Ewald (PME) method with a mesh density of about 1.5 Å^{-3}. Van der Waals forces were calculated using a 12 Å cutoff and a switching function starting at 10 Å. Force evaluations were performed at every time step. Periodic boundary conditions were imposed on the faces of the waterbox and Langevin dynamics was simulated with a temperature of 300 K and damping coefficient of
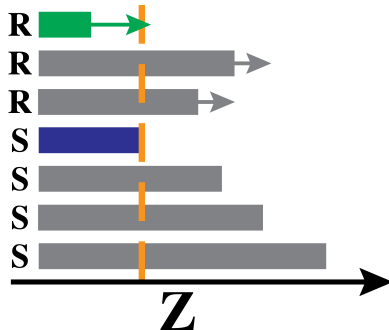
FIGURE 3. Illustration of "pseudo-parallelization" using a chart of maximum z attained in an example AMS simulation with seven replicas. The letters "R" and "S" label running and stopped replicas, respectively. When all running simulations have surpassed the level of the least-progressed stopped replica (shown in blue), an event denoted by the green replica crossing the orange threshold, the blue-colored replica can begin running even before any running replicas terminate.

TABLE 1. Parameters and results of AMS simulations.

k ((kcal/mol) Å^{-2})	z_A(Å)	z_{min}(Å)	z_B(Å)	M	p_{AMS}
0.01	8	10	22	253	0.079 ± 0.013
0.02	8	12	18	202	0.13 ± 0.02
0.08	5	9	15	474	$(4.5 \pm 1.2) \times 10^{-4}$

1 ps^{-1}. Pressure was maintained at 1 atm using a Nosé-Hoover Langevin piston with a damping timescale of 50 fs and a period of 200 fs.

The simulations were carried out in a series of steps. First, the system was minimized over 1000 time steps before being equilibrated for 5 ns with the ion fixed at the origin. Next, $N = 100$ replicas of the system were initialized and run independently, with the ion free to diffuse but under the influence of a spherical harmonic potential $U(r) = \frac{1}{2}kr^2$. Due to the spherical symmetry of the system, we can omit Step 1 of the algorithm, starting with Step 2 instead, without fear of introducing bias. Each replica is run until the ion reaches z_{min}. Step 2 and the subsequent step of the algorithm were performed for three different values of k, with $k = 0.01$ (kcal/mol) Å^{-2}, 0.02 (kcal/mol) Å^{-2}, 0.08 (kcal/mol) Å^{-2}. After the preparation steps above, the resulting states of the replicas were then fed into three sets of simulations.

The first set of simulations follows the AMS algorithm described above. The probability p_{AMS} (Eq. (2)) calculated by the AMS simulation set corresponds to that of the ion, initially at z_{min}, diffusing to z_B without first visiting the sphere $A = \{r : r < z_A\}$. The simulation parameters, number of AMS iterative steps M, and the corresponding p_{AMS} values with error estimates given by the square root of the variance (Eq. (3)) are tabulated in Table 1.

The second set of simulations consisted of direct 10-ns equilibrium runs on each of the 100 replicas. The long sampling time allowed us to measure the times T_1, T_2 and T_3 by averaging over the times taken for the ion to travel between z_A and z_{min} and from z_{min} to z_B. These time values are required to calculate the AMS prediction of τ as per Eq. (4). The trajectories obtained also provided direct measurements of τ and p, given by τ_{sim} and p_{sim}, respectively. τ_{sim} is the average of the measured times taken for the ion starting at z_A to reach z_B in the direct simulations. p_{sim} is measured using the formula $p_{sim} = \frac{n(\text{success})}{n(\text{success})+n(\text{fail})}$ where $n(\text{success})$ and $n(\text{fail})$ are, respectively, the number of trajectories starting from z_{min} that reach z_B before z_A, and the number of trajectories starting from z_{min} that reach z_A before z_B. With the exception of (*), the aforementioned

TABLE 2. Direct measurement of variables required for AMS, with the exception of T_3 for $k = 0.08\,(\text{kcal/mol})\,\text{Å}^{-2}$. The latter was calculated by reconstructing the reactive path trajectories obtained from the AMS algorithm itself.

k ((kcal/mol) Å^{-2})	n(success)	n(failure)	p_{sim}	T_1 (ns)	T_2 (ns)	T_3 (ns)	τ_{sim} (ns)
0.01	105	1136	0.092 ± 0.009	0.035 ± 0.005	0.051 ± 0.008	0.14 ± 0.01	0.94 ± 0.09
0.02	112	734	0.13 ± 0.01	0.13 ± 0.01	0.046 ± 0.004	0.060 ± 0.005	1.0 ± 0.1
0.08	-	-	-	0.34 ± 0.03	0.021 ± 0.001	$0.54 \pm 0.05^{*}$	-

quantities, listed in Table 2, were obtained through direct simulation. (*) was measured from reconstructions of the reactive trajectories generated by the AMS algorithm. In the cases where these direct measurements were possible, we were able to compare the p and τ values with those obtained from the AMS runs; however, we were unable to sample enough "success" events for the $k = 0.08\,(\text{kcal/mol})\,\text{Å}^{-2}$ case. For the purpose of validating the AMS results in the latter case, we performed an analytic calculation on a simple model, described in Appendix A. As a verification of the value of β in the model, we measured in the direct simulations the time-averaged equilibrium ion distribution by histogramming the position of the ion at every 100-fs interval over the entire trajectories of all the replicas. The equilibrium distributions obtained for each k are shown in Figure 4 with the theoretical expected Boltzmann distribution given by $p_B \propto r^2 \exp(-kr^2/2k_BT)$. Note that they compare favorably with the Boltzmann distribution curve, justifying the use of $\beta = 1/k_BT$ in the analytic calculations.
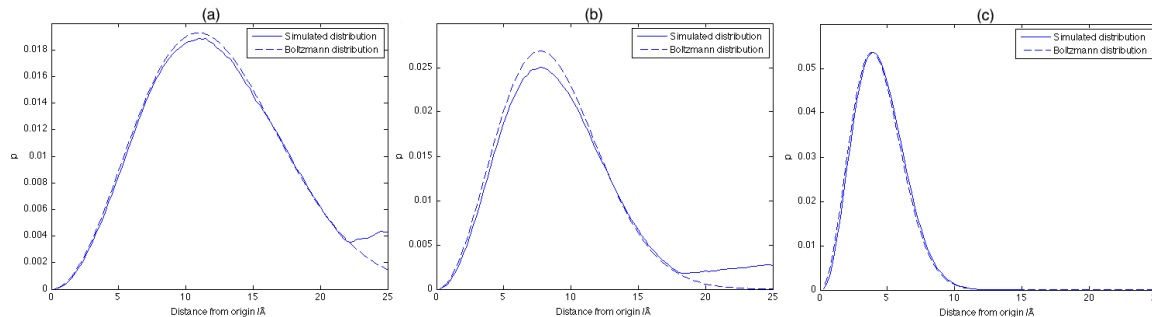

FIGURE 4. Normalized equilibrium distribution curves obtained from simulation and Boltzmann distribution, with (a) $k = 0.01\,(\text{kcal/mol})\,\text{Å}^{-2}$, (b) $k = 0.02\,(\text{kcal/mol})\,\text{Å}^{-2}$, (c) $k = 0.08\,(\text{kcal/mol})\,\text{Å}^{-2}$. The sudden rise at the tail end near 20 Å in (a) and (b) is an artifact of the periodic boundary conditions used in the simulation.

The analytic model also requires the diffusion coefficient of the ion in water, D, to be specified. In the third set of simulations, we measure the local diffusion coefficient at various points in the system for the case $k = 0.08\,(\text{kcal/mol})\,\text{Å}^{-2}$, using an existing method, which is described together with the results in Appendix B. In the analytic calculation in Appendix A, we assume the result, $D = 254 \pm 12\,\text{Å}^2/\text{ns}$, to be constant in space and valid for the other values of k. The uncertainty in τ_{analytic} is due to the uncertainty in D being carried forward.

Using the relevant parameters, we obtain theoretical values, p_{analytic} and τ_{analytic}, for p and τ, respectively, with the analytic derivation given in Appendix A. The estimates from the AMS calculation τ_{AMS} of τ are also obtained using Eq. (4) with values for T_1, T_2 and T_3 from Table 2. Table 3 lists the results from the AMS calculation, direct simulations, and theoretical calculations for comparison. Table 3 shows that the AMS results

TABLE 3. Comparison of p and τ values obtained from AMS, direct simulation, and analytic calculations.

k ((kcal/mol) Å^{-2})	p_{AMS}	p_{sim}	p_{analytic}	τ_{AMS} (ns)	τ_{sim} (ns)	τ_{analytic} (ns)
0.01	0.079 ± 0.013	0.092 ± 0.009	0.084	1.3 ± 0.3	0.94 ± 0.09	0.99 ± 0.05
0.02	0.13 ± 0.02	0.13 ± 0.01	0.13	1.5 ± 0.2	1.0 ± 0.1	1.17 ± 0.06
0.08	$(4.5 \pm 1.2) \times 10^{-4}$	-	3.7×10^{-4}	800 ± 200	-	1040 ± 50

compare favorably with those of the direct simulation and analytic calculation. Values for both p and τ agreed within the error bounds.

Conclusion

Our findings show that AMS can be feasibly applied in MD simulations, providing accurate measurements that agree with both direct simulations and analytic calculations. The accuracy of the AMS measurements in these test cases are impressive in light of the savings in computational cost. Each AMS simulation took about 3 days to complete, in contrast to one direct simulation, which required about 0.5 days on each of 100 replicas, for an equivalent processing time of around 50 days.

Furthermore, there is still much room for improvement in the efficiency of the NAMD implementation of AMS. The AMS algorithm used in the reported test cases did not utilize the "pseudo-parallelization" scheme. The current implementation also is an early Tcl-based prototype which relies on a cumbersome file read/write system. Streamlining of the code, together with enhancement in parallelization through "pseudo-parallelization" or a parallel variant of AMS (refer to Remark 1), will significantly increase gains in efficiency. These improvements will eventually be required as AMS is used for practical applications in biology, which are more challenging than the simple test cases in this study.

The complexity of biological systems may require more than a straightforward increase in software efficiency. For example, one application of AMS in MD would be the measurement of dissociation rates, such as that of a drug molecule from its protein target. However, protein mechanics can be complex, sometimes involving coordinated movements along several degrees of freedom. It is not yet clear what the limitations of AMS are when applied to such systems. One might expect the need to identify sophisticated reaction coordinates, or to use multiple AMS runs with different reaction coordinates, when handling complex systems.

A. Analytic Results of Test System

In this section we derive analytic expressions for p and τ in the case where $\xi(x) = |x|$ and the process $(X_t)_{t \geq 0}$ evolves in $\mathbb{R}^3$ according to the SDE:

$$\gamma \, dX_t = -kX_t \, dt + \sqrt{2\gamma\beta^{-1}} \, dW_t.$$

Thus, the process is governed by zero-mass Langevin dynamics with the friction coefficient γ and harmonic potential energy $V(x) = -k|x|^2/2$. We also recall the definition of A and B in Eq. (1).

Furthermore, let L be the generator of the process $(X_t)_{t \geq 0}$, defined for suitable functions f by

$$Lf(x) = -k\gamma^{-1}x \cdot \nabla f(x) + (\gamma\beta)^{-1}\Delta f(x).$$

We first prove an analytic formula for τ:

Theorem A.1. *Let $u_0(r)$ be the average time for the process $(X_t)_{t \geq 0}$ to reach the level z_B, starting at the level r. Then,*

$$u_0(r) = D^{-1} \int_r^{z_B} s^{-2} e^{\beta k s^2/2} \left(\int_0^s t^2 e^{-\beta k t^2/2} \, dt \right) ds \,,$$

where D is the diffusion coefficient, given by

$$D = (\gamma\beta)^{-1}.$$

In particular,

$$u_0(z_A) = \tau,$$

where τ is the average time for the process to go from ∂A to B.

Proof. Pick $z \in (0, z_B)$, and let the process $(X_t)_{t\geq0}$ be reflected at the level z and absorbed at the level z_B. Define $u_z(x)$ as the average time for $(X_t)_{t\geq0}$ to be absorbed, given that $X_0 = x$ and $z < |x| \leq z_B$. It is well known that $u_z(x)$ is the solution to:

$$\begin{cases} Lu_z(x) = -1, & \text{if } z < |x| < z_B \\ \nabla u_z(x) \cdot x = 0, & \text{if } |x| = z \\ u_z(x) = 0, & \text{if } |x| = z_B \end{cases}$$

Putting this equation in spherical coordinates, applying spherical symmetry and using $D = (\gamma\beta)^{-1}$, we get

$$\begin{cases} -D\beta k r u_z'(r) + Dr^{-2}\frac{d}{dr}\left(r^2 u_z'(r)\right) = -1, & \text{if } z < r < z_B \\ u_z'(z) = 0, \quad u_z(z_B) = 1 \end{cases}$$

where now u_z is a function of r. Re-writing the above expression gives

$$\left(-\beta k r + 2r^{-1}\right) u_z'(r) + u_z''(r) = -D^{-1}.$$

Using the integrating factor $r^2 \exp(-\beta k r^2)$ and the reflecting boundary condition we get

$$u_z'(r) = r^{-2}e^{\beta k r^2/2} \int_r^z D^{-1}s^2 e^{-\beta k s^2/2} \, ds.$$

Integrating again, using the absorbing boundary condition and finally letting $z \to 0$, we obtain:

$$u_0(r) = D^{-1} \int_r^{z_B} s^{-2}e^{\beta k s^2/2} \left(\int_0^s t^2 e^{-\beta k t^2/2} \, dt\right) ds.$$

$\square$

We turn now to an analytic expression for p:

Theorem A.2. *Let $v(r)$ be the probability that the process $(X_t)_{t\geq0}$ reaches the level z_B before z_A, starting at the level r. Then*

$$v(r) = \left(\int_{z_A}^{z_B} s^{-2}e^{\beta k s^2/2} \, ds\right)^{-1} \int_{z_A}^{r} s^{-2}e^{\beta k s^2/2} \, ds.$$

In particular,

$$v(z_{min}) = p,$$

where p is the probability for the process to reach B before A, starting at the level z_{min}.

Proof. Let $v(x)$ be the probability for the process $(X_t)_{t\geq0}$ to hit the level z_B before z_A, given that $X_0 = x$ and $z_A \leq |x| \leq z_B$. It is well known that v is the solution to

$$\begin{cases} Lv(x) = 0, & \text{if } z_A < |x| < z_B \\ v(x) = 0, & \text{if } |x| = z_A \\ v(x) = 1, & \text{if } |x| = z_B \end{cases}$$

Using spherical coordinates as above we get

$$\begin{cases} -krv'(r) + \beta^{-1}r^{-2}\frac{d}{dr}\left(r^2 v'(r)\right) = 0, & \text{if } z_A < r < z_B \\ v(z_A) = 0, \quad v(z_B) = 1 \end{cases}$$

Thus

$$(\beta kr - 2r^{-1})v'(r) = v''(r)$$

and so

$$\log v'(r) = \beta kr^2/2 - 2\log r + C_1.$$

with C_1 a constant. Finally

$$v(r) = C_2 \int_0^r s^{-2} e^{\beta ks^2/2} \, ds + C_3$$

with C_2, C_3 constants. Using the boundary conditions, we obtain

$$v(r) = \left(\int_{z_A}^{z_B} s^{-2} e^{\beta ks^2/2} \, ds\right)^{-1} \int_{z_A}^r s^{-2} e^{\beta ks^2/2} \, ds.$$

$\square$

Remark 2. *We remark that by spherical symmetry, the time τ and the probability p do not depend on the initial distributions of the process on ∂A and at the level z_{min}, respectively.*

B. Determination of Diffusion Coefficient D for Analytic Model

The analytic model requires the local diffusion coefficient D as one of two parameters. D was measured using a formula due to Roux *et. al.* [10] and simplified by Hummer [11], given as follows:

$$D(\mathbf{X} = \langle \mathbf{X} \rangle) = \frac{1}{3} \frac{(\langle \delta \mathbf{X}(t) \cdot \delta \mathbf{X}(t) \rangle)^2}{\int_0^\infty \langle \delta \mathbf{X}(t) \cdot \delta \mathbf{X}(0) \rangle \, dt}, \tag{5}$$

where $\mathbf{X}$ is the Cartesian coordinates of the ion, $\langle \ldots \rangle$ denotes ensemble average (in practice the quantity measured as an average over time) and D is the local diffusion coefficient at position $\langle \mathbf{X} \rangle$.

In accordance with the procedure described in Hummer, a potassium ion was allowed to diffuse in the system under the influence of both the harmonic well potential of constant $k = 0.08$ kcal/mol, and an additional restraining harmonic potential with constant k_r centered at points of radius $r_0 = 0, 10, 20$ Å away from the origin. It is later found that D does not depend on the local potential gradient, hence we assume that the value of D obtained is also valid for other k values. Starting from a state with the ion near r_0, the system was run at equilibrium for 10 ns with data taken every 10-fs interval. Eq. (5) was then used to calculate the value of D at the respective points. The runs were repeated for $k_r = 0.1, 0.3, 0.6$ kcal/mol Å^2. The results are as follows:

k_r ((kcal/mol) Å^{-2})	r_0 (Å)	D (Å^2/ns)
0.1	0	256
0.1	10	246
0.1	20	245
0.3	0	272
0.3	10	269
0.3	20	256
0.6	0	247
0.6	10	234
0.6	20	258

Taking the mean and standard deviation gives $D = 254 \pm 12$ Å^2/ns.

References

[1] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, Scalable molecular dynamics with NAMD, *J. Comput. Chem.* **26** (2005), pp. 1781–1802.

[2] F. Cérou and A. Guyader, Adaptive multilevel splitting for rare event analysis, *Stoch. Anal. Appl.* **25**(2) (2007), pp. 417–443.

[3] F. Cérou, A. Guyader, T. Lelièvre, and D. Pommier, A multiple replica approach to simulate reactive trajectories, *J. Chem. Phys.* **134**(5) (2011), pp. 054108.

[4] A. Guyader, N. Hengartner, and E. Matzner-Løber, Simulation and Estimation of Extreme Quantiles and Extreme Probabilities (2010).

[5] M. Rousset et al, Adaptive Multilevel Splitting: unbiased estimators and new algorithmic variants, *in preparation*.

[6] F. Cérou, P. Del Moral, T. Furon, and A. Guyader, Sequential Monte Carlo for rare event estimation, *Stat. Comput.* **22**(3) (2012), pp. 795–808.

[7] G. Fiorin, M. L. Klein, and J. Hénin, Using collective variables to drive molecular dynamics simulations, *Molecular Physics* **111**(22-23) (2013), pp. 3345–3362.

[8] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, Comparison of simple potential functions for simulating liquid water, *J. Chem. Phys.* **79**(2) (1983), pp. 926–935.

[9] D. Beglov and B. Roux, Finite Representation of an Infinite Bulk System: Solvent Boundary Potential for Computer Simulations, *J. Chem. Phys.* **100**(12) (1994), pp. 9050–9063.

[10] T. B. Woolf T B and B. Roux, Conformational Flexibility of o-Phosphorylcholine and o-Phosphorylethanolamine: A Molecular Dynamics Study of Solvation Effects, *J. Am. Chem. Soc.* **116**(13) (1994), pp. 5916–5926.

[11] G. Hummer, Position-dependent diffusion coefficients and free energies from Bayesian analysis of equilibrium and replica molecular dynamics simulations, *New J. Phys.* **7**(1) (2005), pp. 34.