Extending the timescale in atomically detailed simulations

Alfredo E. Cárdenas

Department of Chemistry, University of South Florida, Tampa, Fl 33620, USA

Eric Barth

Department of Mathematics and Computer Science, Kalamazoo College, Kalamazoo, MI 49006, USA

INTRODUCTION

The current practice of molecular dynamics simulation dates back to the 1960's and the pioneering work on smooth potential models for monatomic fluids of Rahman[1] and Verlet[2]. In the 1970s, interest developed in applying molecular dynamics methods to more complicated molecular fluids such as water[3], molecular fluids with internal degrees of freedom[4] and large flexible molecules[5].

Given a potential energy function V (about which we have much more to say below) that models the interatomic forces in a molecular system with N atoms, the Newtonian equations of motion can then be expressed as

$$\mathbf{M}\ddot{\mathbf{r}} = \mathbf{F}(\mathbf{r}) := -\nabla_{\mathbf{r}} V(\mathbf{r}), \qquad [1]$$

where $\mathbf{M}$ is a diagonal mass matrix with diagonal

$$[m_1\ m_1\ m_1\ m_2\ m_2\ m_2 \ldots m_N\ m_N\ m_N],$$

and m_i the mass of the ith particle. The gradient $\nabla_{\mathbf{r}} V$ is the column vector of all partial derivatives with respect to particle positions. It is easily verified that the total energy

$$E = \frac{\dot{\mathbf{r}}^T \mathbf{M}\dot{\mathbf{r}}}{2} + V(\mathbf{r})$$

is constant along solutions of eq. [1]: $dE/dt = 0$. The system is simulated from initial positions and velocities $\mathbf{r}(t_0) = \mathbf{r}_0$, $\dot{\mathbf{r}}(t_0) = \dot{\mathbf{r}}_0$ often chosen randomly in accordance with some appropriate statistical ensemble.

Computer simulation of the system modeled by [1] requires some sort of time discretization scheme. The method proposed by Verlet propagated positions by

$$\mathbf{r}_i^{n+1} = -\mathbf{r}_i^{n-1} + 2\mathbf{r}_i^n + h^2\mathbf{M}^{-1}\mathbf{F}_i^n, \qquad [2]$$

and velocities using

$$\mathbf{v}_i^n = \left[\mathbf{r}_i^{n+1} - \mathbf{r}_i^{n-1}\right]/2h. \qquad [3]$$

Here the superscripts denote the indices of time steps, each of which is of size h, so

$$\mathbf{r}_i^n \approx \mathbf{r}_i(t_0 + nh),$$

and $\mathbf{F}_i^n = -\nabla_{\mathbf{r}_i} V(\mathbf{r}^n)$ is a Cartesian vector which gives the sum of forces acting on particle i due to interaction with other particles, evaluated at the point $\mathbf{r}^n$.

In his ground-breaking paper, Verlet noted the remarkable energy preservation properties of the integrator, reporting "…small irregularities in the total energy…but the error is of no consequence." The discretization method in eqs. [2–3], commonly known as the "Verlet Integrator", is accurate to second order in time, requires only one force evaluation per step, and is obviously *time-reversible*, which is part of the reason for its excellent stability in terms of near-conservation of energy. In fact it is now known that it is a more general symmetry preservation — the *symplectic* property[6] — of the Verlet method, viewed as an appropriate mapping of positions and momenta, that confers its excellent long-term energy stability[7-9]. For a thorough review of symplectic numerical methods, see the monograph of Sanz-Serna and Calvo[10]. The Verlet method is now regarded as the gold standard for time stepping schemes in molecular dynamics. In conformity with modern practice, and to anticipate the algorithmic development of multiple time step methods in the coming sections, we rewrite the Verlet method equivalently in the "velocity Verlet" form. The inside of the integration loop is given by

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \frac{h}{2}\mathbf{M}^{-1}\mathbf{F}_i^n,$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + h\mathbf{v}_i^{n+1/2}, \qquad [4]$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \frac{h}{2}\mathbf{M}^{-1}\mathbf{F}_i^{n+1}.$$

In contrast to the constant energy regime described above, it is sometimes desirable to perform simulations at a fixed temperature. This can be accomplished by the Langevin dynamics model[11]:

$$\mathbf{M}\ddot{\mathbf{r}} = \mathbf{F}(\mathbf{r}, \mathbf{v}, t) := -\nabla_{\mathbf{r}} V(\mathbf{r}) - \gamma \mathbf{v} + \mathbf{R}(t), \qquad [5]$$

where $\gamma > 0$ is a friction coefficient and $\mathbf{R}(t)$ is a vector of normally distributed random variables with zero mean and covariance $\left\langle R(t)R(t')^T \right\rangle = 2\gamma k_B T\mathbf{M}\delta(t-t')$, where k_B is Boltzmann's constant, T is the simulation temperature and δ is the Dirac delta function. A natural extension of discretization [4] gives the following time discretization scheme[12]:

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \frac{h}{2}\mathbf{M}^{-1}\mathbf{F}_i(\mathbf{r}^n, \mathbf{v}^{n+1/2}, t_n),$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + h\mathbf{v}_i^{n+1/2}, \qquad [6]$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \frac{h}{2}\mathbf{M}^{-1}\mathbf{F}_i(\mathbf{r}^{n+1}, \mathbf{v}^{n+1/2}, t_{n+1}).$$

The molecular dynamics potential

The interactions of polyatomic molecules are modeled by pair potentials, both Lennard-Jones and electrostatic, between all constituent atoms. Also, the model potential must maintain intramolecular geometries by including the "bonded" terms: bond lengths, bond angles, and dihedral angles. The result is the molecular modeling potential function which generally is of the form:

$$V(\mathbf{r}) = V^b + V^a + V^d + V^i + V^{LJ} + V^C, \qquad [7]$$

where V^b, V^a, V^d and V^i are sums over various pairs, triples and quadruples of spatially localized bonded groups of atoms representing bonds, angles, dihedral angles, and improper dihedral angles, respectively:

$$V^b = \sum_{\text{bonds}} V^b_{ij}, \qquad V^a = \sum_{\text{angles}} V^a_{ijk}, \qquad V^b = \sum_{\text{dihed}} V^d_{ijkl}, \text{etc.}$$

Similarly V^{LJ} is the sum of Lennard-Jones contributions for all pairs of atoms, and V^C is the sum of the Coulombic potential over all charge-charge interaction pairs.

$$V^{LJ} = \sum_{\text{allpairs}} V^{LJ}_{ij}, \qquad V^C = \sum_{\text{allpairs}} V^C_{ij}.$$

The functional forms of these terms vary widely. Representative examples can be found in work by a number of authors[13-17]. A simple, detailed model is presented in the MD Test Set project[18].

The MD potential is highly nonlinear, with many local minima. Minimization of the potential energy is a common task, but the nonpolynomial proliferation of local minima frustrates attempts to determine lowest energy states for modeled systems[19]. Also, the finite-time dynamics of a nonlinear multiple-minima system can become trapped in one potential energy well, which impedes conformational sampling. The terms in the potential represent interactions on a wide range of spatial scales (from bonds of length 1 Å=10^{-10} m, to Coulombic interactions which extend throughout the modeled system) and time scales (the fastest bonds have a period of 10 fs=10^{-14} s, while large scale conformational interconversions may occur on the scale of seconds). Time stepping algorithms such as the Verlet method [4] require a time step which is sufficiently short (0.5–1.0 fs) to resolve the fastest bonded motion, meaning that a computed trajectory which spans a time interval of one nanosecond (10^{-9} s) requires one million dynamics steps. As with the earliest molecular dynamics simulations, the great majority of the computational work is expended in computing the forces of interaction—for N particles the computational effort is $O(N^2)$. For simulations in which the only long-range force comes from the rapidly-decaying Lennard-Jones potential, this difficulty can be remedied by imposing *distance cutoffs*: the potential is approximated by zero for all atomic separations greater than a certain cutoff value r_c. In biomolecular simulations, the electrostatic $1/r^2$ forces are non-negligible even at large separations, making distance cutoffs unphysical and undesirable.

Thus, the MD potential function has several characteristics which have an impact on the performance of numerical methods: multiple minima, wide range of time and space scales, and long-range interactions between many particles.

For traditional molecular dynamics simulation using [4], the most important limiting aspect of the potential energy function involves the numerical stability of the integrator. While the computational resources required for a given numerical simulation could be lessened by increasing the length of each time step, stability of the time-stepping algorithm is limited by the molecular high-frequency vibrational modes, such as bond stretching. The fastest period relevant to biomolecules is around 10 fs (associated with O–H and N–H stretching, for example). Resolving these fast motions adequately dictates time steps of length 1 fs or less. Of course the slower and more computationally expensive force components are updated at each step, resulting in undesirable CPU limitation of simulation length and system size. Intense activity is ongoing on the problems of efficient time stepping and fast evaluation of non-bonded forces without distance cutoffs.

Multiple time steps

In multiple time step (MTS) time discretization methods the short range forces, which can change rapidly in time, are updated frequently and included in the numerical dynamics with small time steps. The long range forces are treated with larger steps in time, appropriate to the timescale on which they vary significantly. We will discuss later the fundamental impact the high-frequency force components have on MTS methods as well. In this chapter we trace the development of MTS methods and present a tutorial to show an elementary application of these techniques.

Reaction paths

A very different set of methodologies try to compute trajectories between two states of a molecular system. These “double-ended” algorithms, usually called reaction path approaches, are different from integrators of the Newton equations of motion [4] that only need the initial positions and velocities of the particles in the system. The boundary points can represent a reactant and product configuration or a transition or intermediate state and reactant (product) configuration. The calculated path provides a qualitative description of the structural changes as function of a parameter(s) (reaction coordinate(s)) that characterizes the reaction path. The path then represents a series of replicas of the molecular system $\{\mathbf{X}(s)\} = \{X(s_1), X(s_2), \cdots, X(s_N)\}$ parameterized according to a certain parameter *s*. In this notation, X represents the coordinates of the molecule in a given slice of the path.

Most of the reaction path approaches make use of a spatial step, and therefore are not affected by the time scale limitation of MD methods. However, for complex systems the ruggedness of the potential energy surface limits the applicability and accuracy of these paths because the number of conformations in the trajectory needs to increase. Another fundamental limitation is that these methods are not applicable to study molecular events for which very few details are known about the conformations of products or key intermediates. It is for those processes where theoretical approaches are more helpful.

In MD trajectories it is easy to extract dynamical properties by computing an average over time. The extraction of these properties is more difficult to do in reaction path approaches because trajectories are computed in configuration space. Dynamical information can be computed if an ensemble of many reactive trajectories is obtained. Determination of such an ensemble is not done in general (this can be done with transition path sampling). However, reaction paths methods are very useful in determination of rates and free energy profiles for fast but rare events that are very inefficient to probe using MD algorithms. This is the case because trajectories obtained with reaction path approaches filter out the waiting periods the system spends in the reactant wells, unlike the MD trajectories.

The reaction path methods, comprising a number of different theoretical formulations and algorithms, are difficult to describe by a common framework. The second part of this chapter reviews some of the path techniques developed in the past 10 years. All of these techniques describe the system with atomic detail and use a potential function of the form of Eq. [7]. Then, we will describe with more detail one of these methods based on a discretized formulation of the action of classical mechanics. In this action formalism (called SDEL) reaction paths are obtained linking two conformations of the system. These paths, parameterized according to arc length, can be obtained with a large length step. Therefore, this algorithm tries to solve the time scale limitation of normal MD simulations by using a boundary value formulation of the classical equations of motion. This method has been used to compute approximate paths for processes that are impossible to study using normal MD simulations. However, it is difficult to compute dynamical properties from these paths (for example, the time of the trajectory computed directly from the algorithm is underestimated by several orders of magnitude!). Still, the method can be used to determine large conformational changes that can be resolved by the trajectory. This chapter provides a tutorial section about how to run a program associated with this algorithm and review some recent applications and improvements.

MULTIPLE TIME STEP METHODS

In an effort to lengthen the feasible simulation timescale of molecular simulations, Streett, Tildesley and Saville introduced the multiple time step method to the MD literature in 1978[20]. These authors recognized that the components of the force which vary most rapidly, and hence require small time steps for numerical resolution, are typically associated with pair interactions at small separations. This is important because each of the N particles in the simulation has a small separation distance with only a few, say $k << N$, particles. Success of the multiple time step methodology depends on computing these $kN << N^2$ interactions at each step, i.e. at intervals of h in the numerical schemes given above, while computing the remaining $N(N-k)$ pair interactions [i] (associated with forces that vary more slowly) at longer time intervals τh. Before we discuss implementation details, it is desirable to state the general issues of multiple time step numerical methods. The central objectives are: (1) to devise a splitting of the systematic forces into a hierarchy of force classes based on the time interval over which they vary significantly, and (2) to incorporate these force classes into a numerical method in a way that both realizes enhanced computational efficiency and maintains stability and accuracy of the computed solution.

[i] In fact, the original authors presented the MTS method in the context of a distance truncated Lennard-Jones potential, so that the total number of computed interactions was somewhat less that N^2. For biological MD applications there is evidence that cutoffs can cause undesirable artifacts[21, 22]

Splitting the force

Streett and coauthors proposed a splitting of forces based on a distance parameter r_{split}. In the potential energy formalism we write

$$V(\mathbf{r}) = \sum_{|\mathbf{r_i}-\mathbf{r_j}|<\mathbf{r}_{\text{split}}} \mathbf{V_{ij}} + \sum_{|\mathbf{r_i}-\mathbf{r_j}|\geq\mathbf{r}_{\text{split}}} \mathbf{V_{ij}} \qquad [8]$$

$$= \mathbf{V}^{\text{fast}}(\mathbf{r}) + \mathbf{V}^{\text{slow}}(\mathbf{r}), \qquad [9]$$

with

$$\mathbf{F}^{\text{fast}}(\mathbf{r}) := -\nabla_{\mathbf{r}} V^{\text{fast}}(\mathbf{r}) \qquad \mathbf{F}^{\text{slow}}(\mathbf{r}) := -\nabla_{\mathbf{r}} V^{\text{slow}}(\mathbf{r}).$$

As a practical matter, particles will move in and out of the r_{split} sphere for a given particle over the course of a simulation. To avoid discontinuities which result from a particle suddenly changing classification from the slow to the fast force component, the force can be decomposed into fast and slow components using a switching function $S(r)$,

$$\mathbf{F}_i = \mathbf{F}_i^{\text{fast}} + \mathbf{F}_i^{\text{slow}} = -S(r)\nabla_{\mathbf{r_i}} V - (1-S(r))\nabla_{\mathbf{r_i}} V, \qquad [10]$$

where

$$S(r) = \begin{cases} 1, & r < r_{\text{split}} - \lambda \\ 1 + R^2(2R-3), & r_{\text{split}} - \lambda \leq r \leq r_{\text{split}} \\ 0, & r_{\text{split}} < r \end{cases} \qquad [11]$$

with $R = (r - (r_{\text{split}} - \lambda))/\lambda$. Here λ is a “healing length” over which the switching function S smoothly varies between one and zero. The form of the switching function is somewhat arbitrary, though sufficient smoothness is required.

In treating the full MD potential model, the forces due to bonded interaction potentials V^b, V^a, V^d and V^i are included in the fast component along with the fastest non-bonded forces. It is not uncommon for the two-scale splitting described here to be generalized in a direct way to a hierarchy of more than two classes. We should mention that efficiency gains cannot typically be realized from the simplest splitting of all: bonded forces comprising one class and non-bonded forces the other. The reason is that atomic collisions cause the short-range non-bonded forces to vary over roughly the same (short) timescale as the bonded forces. The details of force splitting can be rather complicated and system-dependent. Later in this work, we will address this important issue influencing efficient implementation of force-splitting in MTS methods.

Numerical integration with force splitting: Extrapolation vs. Impulse

The general plan for a multiple time step numerical method is that $\mathbf{F}_i^{\text{fast}}$ will be evaluated at every step of the integration at time increments h, while $\mathbf{F}_i^{\text{slow}}$ will be evaluated less frequently, typically at time increments τh where $\tau > 1$ is an integer. The key question is: how will $\mathbf{F}_i^{\text{slow}}$ be incorporated in the numerical dynamics? In the original work of Streett et al, the slow force on particle i was approximated by a truncated Taylor series at each step j, $0 < j < \tau$, between updates at steps t_n and $t_n + \tau h$:

$$\mathbf{F}_i^{\text{slow}}(t_n + jh) = \mathbf{F}_i^{\text{slow}}(t_n) + jh\dot{\mathbf{F}}_i^{\text{slow}}(t_n) + \frac{1}{2}(jh)^2\ddot{\mathbf{F}}_i^{\text{slow}}(t_n) + \cdots.$$

Formulas for the required time derivatives up to second order are given by Steet et al[20].

A natural simplification is to truncate the Taylor series after the constant term, resulting in a constant extrapolation of the slow force. The velocity Verlet method [4] can be easily modified to implement this constant extrapolation multiple time step method:

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \frac{h}{2}\mathbf{M}^{-1}(\mathbf{F}_i^{\text{fast}} + \mathbf{F}_i^{\text{slow}}),$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + h\mathbf{v}_i^{n+1/2},$$

$$\text{update } \mathbf{F}_i^{\text{fast}} \qquad [12]$$

$$\text{if } (n+1)\bmod \tau = 0, \quad \text{update } \mathbf{F}_i^{\text{slow}}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \frac{h}{2}\mathbf{M}^{-1}(\mathbf{F}_i^{\text{fast}} + \mathbf{F}_i^{\text{slow}}).$$

Note that the important feature is that the fast forces are computed each step, while the slow forces are computed τ times less frequently, with updates given by:

$$\mathbf{F}_i^{\text{fast}} = \mathbf{F}_i^{\text{fast}}(\mathbf{r}_i^{n+1}) \qquad \mathbf{F}_i^{\text{slow}} = \mathbf{F}_i^{\text{slow}}(\mathbf{r}_i^{n+1}).$$

However, the multiple time stepping puzzle is not so easily solved as this. The simplicity of this modification hides a potentially disastrous flaw! We recall that the Verlet method is popularly employed in virtually every molecular dynamics simulation done today because its geometric symmetry ensures that the total energy along computed solutions does not drift but remains essentially constant, respecting the underlying Newtonian physics of the model. By modifying the force updates in the multiple time step method given above, we have disrupted the symmetry of the original method. The result is that the energy will drift significantly and systematically. The situation is improved, but not solved, by using higher-order Taylor approximation for the slow forces and a higher-order integration scheme. Street et al used third order Taylor approximation of the slow forces and a high order Gear predictor-corrector integration method. In this way, the energy drift can be made small relative to the time step, so that relatively long simulations can be computed with less apparent problems from energy growth.

A new era in multiple time step methods arrived in the early 1990s when Grubmüller et al[23] and Tuckerman et al[24] independently published multiple time step methods that appeared to overcome the energy instability of extrapolation methods. The idea is to mimic the “kick-drift” nature of the velocity Verlet method itself. It is seen in [4] that the force supplies a “kick”, or impulse, in the first line, then the system “drifts” as the updated half-step velocity contributes to the position at the new step. The velocity Verlet method can be modified so that the slow force is also applied as an impulse:

$$\mathbf{v}_i^1 = \mathbf{v}_i^n + \frac{\tau h}{2}\mathbf{M}^{-1}\mathbf{F}_i^{\text{slow}}(\mathbf{r}_i^n),$$

$$\mathbf{r}_i^1 = \mathbf{r}_i^n$$

$$\text{for } j = 1 : \tau,$$

$$\mathbf{v}_i^{j+1/2} = \mathbf{v}_i^j + \frac{h}{2}\mathbf{M}^{-1}\mathbf{F}_i^{\text{fast}}(\mathbf{r}_i^j),$$

$$\mathbf{r}_i^{j+1} = \mathbf{r}_i^j + h\mathbf{v}_i^{j+1/2}, \qquad [13]$$

$$\mathbf{v}_i^{j+1} = \mathbf{v}_i^{j+1/2} + \frac{h}{2}\mathbf{M}^{-1}\mathbf{F}_i^{\text{fast}}(\mathbf{r}_i^{j+1}),$$

$$\text{end}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^{\tau}$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{\tau} + \frac{\tau h}{2}\mathbf{M}^{-1}\mathbf{F}_i^{\mathrm{slow}}(\mathbf{r}_i^{n+1}).$$

The modification amounts to the replacement of the middle step of [4] with an inner loop over the τ steps between slow force updates.

It can be shown that impulse multiple time step algorithms, such as the one shown here, can be formulated so as to preserve time-reversibility. As a result these methods can, for suitable choices of time step sizes, avoid systematic energy drift along computed trajectories. In the next section we discuss the question of feasible time step size. This question is the heart of the matter. In order to consume less computational power per unit of simulation time, successful multiple time step methods must combine force splitting approaches and time stepping algorithms which allow significantly lengthened time steps for the most computationally costly force components. This issue has been the focus of intense work over the past decade.

The fundamental limitation on the size of MTS time steps

Impulse MTS methods began to show considerable success in the mid-nineties, with published results reporting computational speed-up by a factor of five compared to traditional MD simulation[25, 26]. Two features emerged with regard to the practical time step sizes for MTS methods. The first was consistent with results reported by Street et al 20 years earlier: the size of the small time step — used to resolve the highest frequency motion in the system — needed to be somewhat smaller than the typical MD time step in order to maintain energy stability of the solution. In terms of overall computational efficiency this is of little practical concern because the forces being evaluated at each small step are assumed to be very inexpensive in CPU time. The second feature was more significant: computed solutions demonstrated systematic energy instability whenever the larger steps used to resolve slower force components exceeded 5 fs[27]. This is important because the possibility of further efficiency gains with MTS methods require that the slowest forces be updated much less frequently. The 5 fs barrier, which for a time seemed to have put a ceiling on further developments, came to be understood as a resonance artifact[28-30] coinciding with the half-period of bonds such as O-H. The impulses introduced into the dynamics at each large step excite the bonds and lead to catastrophic energy growth. This energy growth is seen initially in the fastest bonded energy. As a practical matter, growth in these energy components can provide an early signal of trouble in an MTS simulation.

One obvious remedy might be to choose time step lengths so as to avoid small integer multiples of half-periods of any oscillatory motion. However, it has been demonstrated[31] that the molecular dynamics potential gives rise to motion with a continuum of periods greater than or equal to 10 fs. Furthermore, the energy instability of impulse MTS methods becomes exponentially worse at larger multiples of the half-periods. This rules out the possibility that a fortuitously chosen assortment of impulse MTS time steps longer than 5 fs could yield stable computation.

A number of methods have been proposed to overcome the MTS time step barrier, including averaging methods[32] which “mollify” the impulse, allowing time steps of up to 6 fs while maintaining the favorably small energy drift attained by impulse MTS methods at time steps of 4 fs. We will omit the details of these time-stepping algorithms, but point to a reference[33] which explicitly provides implementation details.

A point of great interest is that extrapolation methods such as the one given above in equation [13] suffer from resonance artifacts to a lesser degree than impulse methods[34]. In particular, it has been demonstrated[35, 36] by eigenvalue analysis of extrapolation MTS methods that the severity of the instability does not grow with the largest MTS steps. Similar eigenvalue analysis shows that impulse methods suffer from increasingly severe energy instability with increasing MTS steps. This suggests that if the relatively mild instability of extrapolation methods could be somehow managed, the 5 fs time step barrier could be overcome. In the next section we discuss how extrapolation MTS methods, while unsuitable for simulations in which energy must be conserved, can achieve the goals of true long-time-step methods in the context of Langevin dynamics.

Langevin stabilization

The Langevin dynamics model of equation [5] has been employed to meet a variety of modeling objectives. As a fixed-temperature method, it provides a way to carry out numerical simulations which sample from the canonical ensemble of statistical mechanics[37]. Also, the stochastic forcing function can be viewed as a model of intermolecular collisions, making the Langevin model suitable as a surrogate for explicit solvent molecules[12]. A third way the versatile Langevin formalism has been utilized is to provide energy stabilization for simulations using numerical methods which may have desirable properties but suffer from energetic drift when applied to MD equations of motion[38, 39].

Langevin stabilization as an approach to multiple time step numerical integration was introduced by Barth and Schlick[35] in 1998. One particularly simple method described in that paper can be written easily as a modification of [13] subject to [5]:

$$\begin{gathered}
\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \frac{h}{2}\mathbf{M}^{-1}(\mathbf{F}_i^{\text{fast}} + \mathbf{F}_i^{\text{slow}}), \\
\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + h\mathbf{v}_i^{n+1/2}, \\
\text{update } \mathbf{F}_i^{\text{fast}} \\
\text{if } (n+1) \operatorname{mod} \tau = 0, \quad \text{update } \mathbf{F}_i^{\text{slow}} \\
\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n+1/2} + \frac{h}{2}\mathbf{M}^{-1}(\mathbf{F}_i^{\text{fast}} + \mathbf{F}_i^{\text{slow}}).
\end{gathered} \qquad [14]$$

where

$$\mathbf{F}^{\text{fast}}(\mathbf{r}, \mathbf{v}, t) := -\nabla_{\mathbf{r}} V^{\text{fast}}(\mathbf{r}(t)) - \gamma \mathbf{v}(t) + \mathbf{R}(t) \qquad \mathbf{F}_i^{\text{slow}} = \mathbf{F}_i^{\text{slow}}(\mathbf{r}^{n+1}, \mathbf{v}^{n+1/2}, t_{n+1}).$$

This method can naturally be extended to include a splitting of more than two force classes, and is amenable to other modifications and improvements such as moving the slow force update to be more symmetrically placed in the integration loop. For a three-class force splitting, it was shown[35] that for typical biomolecular systems, the MTS time steps (and hence the frequency of updating the slow forces) can be extended to 48 fs or more with resulting computational speedups of at least a factor of ten. The success of Langevin stabilization with extrapolation MTS methods has led to its use to achieve stable simulations at large time steps using mollified impulse methods as well[40-42]. A systematic comparison[43] of these methods shows that extrapolation methods hold some advantage among Langevin-stabilized MTS integrators in terms of stability at long time steps. On the other hand, the argument in favor of mollified impulse methods is that no stabilization is required at small time steps.

Further Challenges and Recent Advances

The success of Langevin-stabilized methods has yielded, for the first time, the opportunity to explore the full promise of MTS methods. As in any field of inquiry, solving one problem often clears the way for the emergence of several others, and MTS integration is no exception. As an example, the extrapolation method[35] was successfully used with 120 fs slow force update frequency in simulations of a DNA/polymerase system[44]. Remarkably, in this study computational gains were limited to a factor of 5, even though the longest-range forces were updated two orders of magnitude less frequently than the fastest forces. The explanation is likely due to the extreme sensitivity of DNA systems to electrostatic interactions, which requires that medium-range forces be treated with very short time steps in the range 1–2 fs. This points to a difficulty with the force splitting idea as applied to molecular dynamics of biological systems: The time and distance scales do not naturally fall into well separated categories which would give rise to easy identification of force classes. The result is that the highest frequency motion in a slow force class might be only marginally lower than the lowest frequency motion in a fast force class. This feature, more prominent in some systems than others, can severely limit the size of time step which can be used for any given force class, hence attenuating the hoped-for computational advantage of MTS methods.

During the decade that saw development of long time step MTS methods, work proceeded along another path on the fundamental problem of computationally costly force evaluations in MD. This research was aimed at the development of summation schemes, such as fast multipole[45-48] and Ewald summation[49-51], for faster evaluation of electrostatic energies and forces without distance cutoffs. The computational advantage over direct evaluation of long-range electrostatic forces is the reduction in the complexity of the task from $O(N^2)$ to $O(N \ln N)$. For solvated systems, fast summation methods have been successfully integrated into the framework of force-splitting in multiple time step integration methods[25, 52, 53]. While it is often the case that fast summation methods enjoy computational speedups compared to direct evaluation only for sufficiently large systems, it is precisely these large systems that would have exhibited the largest computational speedup for long time step MTS methods.

The current situation is then that large time step size allows the slowest forces to be evaluated very infrequently over the course of a multiple time step MD simulation. At the same time, these forces can be evaluated rather more cheaply than was possible several decades ago[ii]. As the long range forces become both cheaper to evaluate and this evaluation needs to be done less frequently during

[ii] There are some technical limitations to the conjoined use of MTS methods and fast summation[32, 54]

a simulation, our attention must go to the new cost leader in MD simulation: the medium-range forces. These forces are not especially slow (as in the DNA/polymerase simulation[44]), nor are they particularly amenable to efficient approximation by fast summation methods. They can and do require careful resolution with rather small time steps. Further efficiency gains will require new approaches to the medium-range forces. This issue is open to new research from a variety of fields: force-field modeling, force splitting techniques and time-stepping algorithms.

An MTS tutorial

In this section, we present a simple model problem to illustrate impulse and extrapolation MTS methods for simulations in the constant energy and constant temperature regimes. The models are implemented in MATLAB, with codes given at a supplementary web site[55].

The model consists of a pair of water molecules in the plane. The forces of interaction come from spring models of oxygen-hydrogen (O-H) bonds, as well as H-H springs to maintain the water model geometry. In a more realistic model, the angle between the two 0-H bonds would correspond to a more complicated potential function, but the three spring model suits our purposes here. The atoms of each molecule also interact, via electrostatic forces, with atoms of the other molecule. To avoid the complexity of treating the system in a way that takes the pressure of the system into account, we fix the two oxygen atoms in place. This is accomplished by simply making the model forces to be zero for these two atoms.

The system is represented graphically in figure 1. In the figure we indicate the central dipole vector for each molecule. In analyzing the results of simulations, we will monitor the motion of the angles θ indicated in the figure.

The model uses natural units: Angstroms, Kilocalories per Mole and Atomic Mass Units (AKMA). In this system of units, lengths and masses have obvious values (for example a hydrogen atom has approximately unit mass), but the resulting time unit is non-standard: 4.888821×10^{-14} seconds. For this reason, it is seen that in our examples, a time conversion is required so that simulation can be presented in time units of femtoseconds (fs).

To begin, we must state explicitly the potential energy function, using parameter values derived from the CHARMM potential model[14]:

$$V_{ij}^{\text{bond}} = \frac{k_{ij}}{2}\left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (y_i - y_j)^2} - L_{ij}\right)^2$$

and

$$V_{ij}^{\text{elec}} = \frac{q_i q_j}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (y_i - y_j)^2}},$$

where the spring constants and equilibrium lengths for H-H and O-H bonds are given by

$$k_{HH} = 450.0 \qquad k_{OH} = 450.0 \qquad L_{OH} = 0.9573 \qquad L_{HH} = 1.5139$$

and the electrostatic constants for hydrogen and oxygen are

$$q_H = 0.417 \qquad q_O = -0.834.$$

The total potential energy for the system is obtained by summing the V^{bond} terms for all six bonds, and V^{elec} for the eight electrostatic pairs. Note that because the oxygen atoms are not subjected to any force, we need not compute interactions between them, resulting in $3 \times 3 - 1 = 8$ non-bonded pairs. The forces are then computed as the negative gradient of V^{elec}. The form of the total forces can be seen in the MATLAB code twowaters.m. We have implemented the time-stepping algorithms [4–14]. Simulations using the various algorithms are coordinated and dispatched by the mfile dyntwowat.m.

High and low frequency motion: The importance of electrostatic forces

For typical configurations of the model system with the molecules separated by about 5 Å, it is seen that the forces due to the bonded interactions have magnitude on the order of 10^1, while the electrostatic forces have magnitude on the order of 10^{-3}. It is natural to wonder about the overall importance of the tiny electrostatic forces. We performed long MD simulations using [4] in the presence and absence of electrostatic forces in the model. The left view in figure 2 shows one position component of one of the molecules over a few dozen time steps — enough to capture several periods of the fastest motions due to the bond forces. It can be observed that the fastest oscillations have period of approximately 9 fs. The right view in figure 2 shows a much longer trajectory of the dipole angle θ. The presence of the electrostatic forces can be seen to result in very low frequency motions — several orders of magnitude slower than the fast bond-induced motions. Without electrostatic interaction, the dipole angle remains essentially constant, with small fluctuations due to molecular vibration. These low-frequency motions are typical of the biologically important motions in realistic biomolecular models. It is common to perform power spectrum analysis of MD trajectories in order to identify the constituent motions of the model system. Figure 3 shows the power spectrum for the dipole angle. Notice, for example, that the highest frequency peak at 0.114 fs^{-1} corresponds to motion with period 8.8 fs. Also note that in the left view the slowest motion appears as a frequency peak of 0.0001 fs^{-1}, which agrees with the trajectory in figure 2 with apparent period of 10000 fs. A MATLAB script for plotting power spectra is given in powerspectrum.m.

Behavior of impulse MTS method on the model system

The model system was designed to allow the simplest possible force splitting. The bonded forces constitute the fast force. The two molecules are constrained to be sufficiently distant from one another to avoid any close approaches, so the non-bond forces can be taken as the slow force. With this splitting, the impulse MTS method [12] was used with step sizes of $h = 0.5$ fs and various values of $\tilde{\tau}$. Figure 4 shows the RMSD energy error for the method versus the interval between slow force updates corresponding to integer values of τ between 1 and 20. It can be seen that the energy error from τ=1 (equivalent to velocity Verlet) to τ=8 (corresponding to slow force updates on intervals just shorter than half the period of the fastest motion) is remarkably constant. Unstable energy behavior begins for τ=9, and is especially notable at the period of the fastest motions. Power spectrum analysis shows that the low frequency motion is correctly resolved by the impulse MTS method for values of τ in the stable regime.

Behavior of Langevin-stabilized extrapolation method on the model system

Langevin Dynamics requires the calculation of a random force vector at every step. This is implemented for the current model in the MATLAB function dynlang1.m. The best value of the collision parameter γ is an open question. In the original work[35], the choice was 5/ps**.** For this model, we present results using a smaller collision parameter 2/ps. This corresponds to 0.1 in the units presented here. Power spectrum analysis shows that Langevin dynamics simulations on a given model exhibit essentially the same frequencies as constant-energy simulations do, but with broadening of the frequency peaks. This spreading of the peaks is in direct correlation with the magnitude of γ with larger values resulting in broader peaks. This effect can be seen in figure 5. We note that this is especially the case for the low frequency motion, where the stochastic forces can excite rotations in the dipole about the fixed atoms. This broadening can be viewed in two lights, depending on the nature and aims of the simulation. If detailed time-dependent dynamical information about the very low frequency motion is needed, the stochastic forces in Langevin dynamics blur the picture somewhat. On the other hand this can result in enhanced sampling of the low frequency motion, allowing simulations to capture this important motion on a shorter simulation time scale.

As we pointed out earlier, energy of the bonded terms is an especially sensitive detector of instability due to MTS resonance artifacts. Because total energy is not constant in Langevin simulations, bond energy can be used to monitor the stability of a Langevin trajectory. Figure 6 shows that the energy behavior of the

Langevin-stabilized extrapolation method is largely independent of τ, allowing for large MTS steps while maintaining the underlying behavior of the single-step Langevin method [6].

EXTENDING THE TIMESCALE: PATH METHODOLOGIES

In computational chemistry, molecular dynamics (MD) is the most widely used methodology to study the kinetic and thermodynamic properties of atomic and molecular systems[56-58]. These properties are obtained by solving the microscopic equations of motion [1] for the system under consideration. The multiple time steps algorithms discussed in the previous section increase the time scale that can be reached but the gain is still not enough for many processes. For many systems, like biomolecules, this simulation time is not enough to study large conformational changes or to study rare but important events.

Due to this limitation of traditional and multiple time step MD algorithms, a different approach can be considered. This different set of methodologies attempt to compute trajectories connecting conformations from the reactant state to conformations of the product state, i.e., the reaction path. Transition path sampling, MaxFlux, discrete path sampling, string methods and optimization of actions are examples of methodologies searching for these transition paths. In this section, we will review briefly the first four methods. Then, the theory and implementation of the action formalism will be described with more detail.

Transition path sampling

Transition path sampling (TPS) is a methodology that can be used to study slow activated processes. This technique, first developed by the Chandler group[59, 60] and further improved by Bolhuis et al.[61-64], is based on a polymer-like representation of the complete trajectory. TPS is an iterative method that starts by computing a dynamical pathway connecting conformations of the reactant and product state. This can be done using simpler methods that generate approximate trajectories connecting two boundary points.

Starting from this initial path, further trajectories are generated using an iterative strategy. Specifically, from the previous trajectory a configuration snapshot is taken and modified (Monte Carlo shooting method) in a manner consistent with the corresponding distribution ensemble. Usually the incorporated change is a momentum variation. Then, starting from this modified configuration, forward and backward trajectories are generated using MD. If this pathway connects the reactant and product state, this reactive trajectory is chosen to generate new ones. Iterating this process many times, an accurate sampling of trajectory space can be generated. From these trajectories, reaction mechanisms and transition states can be elucidated, and rate constants can be determined using the time derivative of the time correlation function[61, 65]:

$$k_{AB}^{TPS}(t) = \frac{dC(t)}{dt}, \qquad C(t) = \frac{\langle \mathrm{h}_{\mathrm{A}}(\mathbf{x}_0) h_B(\mathbf{x}_\mathbf{t}) \rangle}{\langle \mathrm{h}_{\mathrm{A}}(\mathbf{x}_0) \rangle} \quad \text{and}$$

$$k_{AB}^{TPS}(t)=\frac{\langle \dot{h}_B(t)\rangle_{A,H_B(T)}}{\langle h_B(t')\rangle_{A,H_B(T)}}\cdot C(t') \qquad [15]$$

in which $h_A(\mathbf{x})$and $h_B(\mathbf{x})$ are indicator functions that states if the system is the phase space regions *A* or *B*:

$$\begin{aligned} &h_A(\mathbf{x})=1\ ,\quad \text{if } \mathbf{x}\in A\,,\ \ \text{else}\ \ h_A(\mathbf{x})=0\,,\\ &h_B(\mathbf{x})=1\ ,\quad \text{if } \mathbf{x}\in B\,,\ \ \text{else}\ \ h_B(\mathbf{x})=0\,. \end{aligned} \qquad [16]$$

Here **x** is a phase space vector, $H_B(T)=\max_{0<t<T} h_B(\mathbf{x}_t)$, and $\langle\cdots\rangle_{A,H_B(T)}$ is an average over the ensembles of paths that start in *A* and go to *B* at least once during a fixed length *T*. An order parameter is introduced to describe the transition, and umbrella sampling can be used to compute the rate in Eq. [15].

The molecular processes studied with TPS are typically associated with a transition over a single significant barrier. TPS is more efficient than standard MD because the reactive trajectories (computed by TPS) are much shorter than the time it takes between successive transitions. Therefore, more (reactive) trajectories are computed with TPS than normal MD.

The methodology has been successfully applied to many systems, such as chemical reactions and conformational changes[62, 66-71]. However application of this algorithm to complex systems with rugged energy surfaces requires the identification of basin states separated by several barriers with different heights. For these systems, the assumption of time scale separation between the transition time and the incubation time is not easy to justify. For complex systems, the reactive trajectories can be long and the sampling will be limited by the time step used in the simulation. The definition of the reaction coordinate or physical descriptor that allows the identification of the different basin and transition states present during transitions of complex molecules can be cumbersome[72].

Maximization of the diffusive flux

This method (MaxFlux) is a time-independent algorithm that finds the path that maximizes the diffusive flux (or minimizes the mean first passage time) between two configurations at a given temperature. The algorithm is based on the work of Berkowitz and co-workers who derived the optimal transition connecting reactant and product using a variational principle[73]. If the transition is described as a stochastic process, the flux of particles along the optimal path is given by

$$j \propto \frac{\tan t}{\gamma \int \exp(\beta V) dl} \quad , \qquad [17]$$

where *V* is the potential of mean force of the system, γ an isotropic and spatially independent friction coefficient, $\beta = 1/k_B T$ and *dl* is an infinitesimal length element along the path. In MaxFlux, the line integral in the denominator of Equation [17] is minimized using a self-penalty walk method[74].

MaxFlux has been applied by the group of John Straub to study conformational transitions in peptides and aggregate formation[75, 76]. This approach can be used to describe slow processes controlled by diffusion. A difficulty in this description is the necessity to specify the phenomenological friction constant. The value of the friction constant strongly influences calculations of rates and affect the transition pathways. The maximization of equation [17] uses global optimization algorithms that are time-consuming and dependent on the initial guess for the pathway.

A temperature-dependent nudged-elastic-band (NEB) algorithm based also on the maximization of the flux was recently proposed[77, 78]. In this MaxFlux-NEB algorithm, based on the differential form of Equation [17], a discretized path is constructed with the different neighboring structures maintained equally spaced by the use of spring forces. Then, the path is minimized using a modified Verlet algorithm. This methodology has similar limitations as the MaxFlux approach.

Discrete path sampling and string method

Discrete path sampling (DPS)[79-82] is a methodology that samples paths along the potential energy surface (instead of the Gibbs free energy as in TPS). In this method, fast paths connecting local minima and transition state conformations are computed. The initial path connecting minima and transition states is computed using the nudged elastic band (NEB) method[83, 84] and the number of paths is increased by replacing a minimum in the path with a new minimum close to the original path. The new path is accepted and used to generate new paths if it is faster than the original path. The rate constants are computed using a harmonic approximation to the local density of states for each stationary point of the potential energy surface. Overall phenomenological rate constants can be extracted using master equations, kinetic Monte Carlo or graph transformations and transition state theory. The algorithm has been applied to a small pentapeptide[79] and the GB1 hairpin[80]. Reliance on statistical rate theory is one of the drawbacks of this methodology. A satisfactory sampling of stationary points of the potential energy for more complex systems can be difficult as well.

In the string method[85-89], based on the transition path theory (TPT)[90, 91], a transition tube in configuration space is constructed by performing a sampling of the equilibrium distribution of the system in a collection of hyperplanes. These hyperplanes are parameterized by a string connecting two metastable states.

The hyperplanes approximate the isocommittor surfaces (trajectories initiated at configurations on this surface have the same probability to reach the product state before reaching the reactant state). The transition tube represents a region in configuration space in which the transition occurs with high probability. The string is a curve normal to each hyperplane passing through the center of mass on each plane, and it defines the center of the transition tube. It is defined as

$$\varphi(\alpha) = \langle \mathbf{x} \rangle_{P_\alpha} , \qquad [18]$$

in which the average is restricted to equilibrium configurations on the hyperplane P_α (α is a parameter characterizing the hyperplanes).

The algorithm uses a variational principle to compute this string that satisfies[87]:

$$\hat{\mathbf{n}}(\alpha) \parallel \varphi'(\alpha) \quad , \qquad [19]$$

where $\hat{\mathbf{n}}(\alpha)$ is a unit vector normal to the hyperplane P_α and $\varphi'(\alpha)$ is the tangent vector of the string at α.

The string method is a more sophisticated procedure compared to MaxFlux and DPS but due to its inherent complexity has been used only for very simple systems such as alanine dipeptide[87].

Optimization of action

Another set of algorithms have been developed in the past few years in the group of Ron Elber based on the optimization of actions[92, 93]. In these methods an initial guess for the trajectory is generated connecting two boundary states, and the least action formalism is used to compute a finite-temperature trajectory.

The first formulation of this methodology was based on a discretized version of the classical action:

$$S[\mathbf{X}(t')] = \int_0^t L dt' ,$$

generating the Onsager-Machlup object function[94-98]:

$$S_{SDET}\left(\{\mathbf{X}_i\}_{i=1}^{N-1}\right) \equiv \sum_i \Delta t \left(\mathbf{M} \frac{\mathbf{X}_{i+1} + \mathbf{X}_{i-1} - 2\mathbf{X}_i}{\Delta t^2} + \frac{dV}{d\mathbf{X}_i} \right)^2 = \sum_i \Delta t \varepsilon_i^2$$

In these equations, **X** is the coordinate vector for the system, $\mathbf{M}$ is the diagonal mass matrix, *V* the potential energy, $\mathbf{X}_0$ and $\mathbf{X}_N$ are the fixed boundary conformations in the trajectory and ε_i is an error variable. This algorithm, called stochastic difference equation in time (SDET), has been used to compute *approximate* trajectories using a large time step *Δt* for long time events. These paths are obtained by sampling trajectory space using molecular dynamics or Monte Carlo according to a Gaussian distribution of errors (the term between parentheses corresponds to a finite difference version of the Newton's equation of motion [2]). Using similar time formalisms, Passerone and Parrinello[99, 100] and Bai and Elber[101] have computed *exact* trajectories for relatively short but rare processes.

In this section a variant of the SDET algorithm will be described with more detail. In this more recent formulation called SDEL (for stochastic difference equation in length) the trajectory is parameterized as a function of its arc length and a unique path is obtained connecting the two boundary conformations[92, 93]. In this sense, the SDEL algorithm is similar to DPS and string methods because trajectories are computed in configuration space instead of the space parameterized by time like in normal MD, TPS and SDET algorithms.

Boundary value formulation in length

The SDEL algorithm allows the computation of atomically detailed trajectories connecting two known conformations of the molecule over long time scales. In contrast to normal and MTS molecular dynamics algorithms, step sizes can be easily increased two or three orders of magnitude without significative changes in many properties of the trajectory. The trade-off is that trajectories obtained with such a large step size are approximate: molecular motions that occur on a scale shorter than the step size are filtered out from the trajectories. Also, the initial and final configurations must be known because this is a boundary value algorithm. This means that the algorithm can not be used to predict the final conformation of a molecular system such as a protein. This confines the applicability of the algorithm to situations in which the initial and final configurations are known by experiment or modeling. This is not an unbearable limitation. In many chemical events, we are interested to determine how a system changes from a reactant state to a product state. For example, the algorithm can be used to describe folding mechanisms[102]: how a protein folds to its native conformation starting from an unfolded structure.

The Onsager-Machlup action methodology has a significative disadvantage: the total time of the trajectory is needed in advance. Also, low resolution trajectories do not approach a physical limit when the step size increases, in contrast to SDEL as will be shown below.

Similarly to the Onsager-Machlup action method, the SDEL algorithm is based on the classical action. However, in this case the starting point is the action *S* parameterized according to the length of the trajectory[103]:

$$S = \int_{\mathbf{Y}_u}^{\mathbf{Y}_f} \sqrt{2(E - V(\mathbf{Y}))}dl \qquad [20]$$

where $\mathbf{Y}_u$ and $\mathbf{Y}_f$ (lower and upper limits of integration) are the mass weighted coordinates ($\mathbf{Y} = \sqrt{\mathbf{M}}\mathbf{X}$) of the initial and final conformation of the system, respectively, *E* is the total energy and *V* is the potential energy of the system, and *dl* is an infinitesimal mass-weighted arc length element for the path connecting $\mathbf{Y}_u$ and $\mathbf{Y}_f$. Using the least-action principle of classical mechanics one obtains a classical trajectory connecting these two states of the system when a stationary solution for the action is computed, i.e., $\delta S/\delta \mathbf{Y} = 0$ (the action is not necessarily a minimum[103, 104]). These trajectories are calculated differently from usual MD simulations. First, the trajectory is obtained using double boundary conditions: the initial and final coordinates of the system are required as input. In contrast, in the MD algorithm the initial positions and velocities (usually chosen randomly from a Boltzmann distribution) are needed. Second, the trajectory in

Eq. [20] is parameterized as a function of length and not as a function of time. Finally, in the SDEL formulation the total energy of the trajectory is fixed. In contrast, in an MD trajectory the total time is fixed once the step size and the number of steps are constrained in the calculation.

Computation of exact trajectories using Eq. [20] is more expensive than in normal MD because the evaluation and optimization of the action entails the knowledge of the entire trajectory. However, if the aim is to obtain an approximate trajectory with a large step size between successive structures, optimization of Eq. [20] is a more feasible task and generates a more stable trajectory than in a straightforward MD algorithm. Numerically, an approximate trajectory is computed from Eq. [20] when a large step size is used *Δl >> dl.* Specifically, after replacing $dl \to \Delta l$ a discrete version of the action in Eq. [20] is obtained:

$$S \cong \sum_{i=0,\ldots,N} \sqrt{2(E - V(\mathbf{Y}_i))}\Delta l_{i,i+1} \qquad [21]$$

where the action is now a function of the coordinates of the *N* intermediate structures in the path, $\{\mathbf{Y}_i\}_{i=1}^{N}$ (with the coordinates of the structures $\mathbf{Y}_0 \equiv \mathbf{Y}_u$ and $\mathbf{Y}_{N+1} \equiv \mathbf{Y}_f$ held fixed), and $\Delta l_{i,i+1}$ is the mass-weighted distance separating consecutive structures in the trajectory ($\Delta l_{i,i+1} = /\mathbf{Y}_i - \mathbf{Y}_{i+1}/$). The trajectory that makes *S* stationary is determined by optimization. Thus, the optimized trajectory represents a sequence of structures connecting the initial and final state of the system. Explicitly, after optimization the following expression is obtained[103]:

$$\frac{\partial S}{\partial \mathbf{Y}_i} \cong \frac{\Delta^2 \mathbf{Y}_i}{\Delta l^2} - \frac{1}{2(E - V(\mathbf{Y}_i))}(\nabla V - (\nabla V \cdot \hat{\mathbf{e}}_i) \cdot \hat{\mathbf{e}}_i) = 0$$

[22]

with $\frac{\Delta^2 \mathbf{Y}_i}{\Delta l^2} = \frac{2\mathbf{Y}_i - \mathbf{Y}_{i+1} - \mathbf{Y}_{i-1}}{\Delta l^2}$ and $\hat{\mathbf{e}}_i \equiv \frac{\mathbf{Y}_{i+1} - \mathbf{Y}_{i-1}}{\left|\mathbf{Y}_{i+1} - \mathbf{Y}_{i-1}\right|}$,

where $\hat{\mathbf{e}}_i$ is a unit vector tangential to the path at slice *i*, and the length step *Δl* can be made a constant in the calculation and therefore independent of the index *i*. The first term of Eq. [22] (equivalent to the acceleration term in the Newton's equation of motion [1]) depends on the step size. At larger step sizes, this "acceleration" contribution becomes smaller. In the limit when this inertial term can be neglected, Eq. [22] becomes:

$$\frac{\delta S}{\delta \mathbf{Y}_i} \approx \frac{1}{2(E - V(\mathbf{Y}_i))}\left(\nabla V(\mathbf{Y}_i) - (\nabla V(\mathbf{Y}_i) \cdot \hat{\mathbf{e}}_i)\hat{\mathbf{e}}_i\right) = 0$$

$$\rightarrow \nabla V(\mathbf{Y}_i) - (\nabla V(\mathbf{Y}_i) \cdot \hat{\mathbf{e}}_i)\hat{\mathbf{e}}_i = 0 \qquad \forall i \tag{23}$$

Equation [23] generates a path in which the force is minimized in all directions but the tangential direction of the path. This is one of the definitions of the minimum energy path (MEP)[105]. This equation suggests that SDEL provides a physically meaningful trajectory even at low resolution (large step sizes).

To obtain an approximate trajectory using SDEL the following steps are used (Figure 7):

1. An initial guess for the path connecting the initial and final coordinates of the system is generated using a MEP algorithm, for example by self-penalty work (SPW)[74] or NEB[83, 84].

2. The total energy of the system *E* (needed to evaluate and optimize the action) can be estimated by averaging the total energy obtained using several MD simulations of the system at the temperature of interest. An alternative method is to identify the highest and lowest values of the potential energy for the

MEP trajectory, and then compute the average thermal energy at the top of the barrier (i.e., $E = V_{high} - V_{low} + ((3L-6)/2)k_B T$, where *L* is the number of atoms in the system).

3. A stationary solution for the action *S*, for which $\partial S/\partial \mathbf{Y}_i = 0$, is obtained by minimizing the square of the action gradient $\Theta = \sum(\partial S/\partial \mathbf{Y}_i)^2$ using a simulated annealing protocol (Figure 7). The evaluation of the potential energy, forces and hessians needed by the optimization algorithm is performed using the Amber/OPLS force field parameters including in the molecular simulation package MOIL[106].

4. Finally, the optimized trajectory is examined and its accuracy is estimated by the step size *Δl*. This step size should be small enough to provide a smooth representation of the path. The cut-off value *Δl$_c$* depends on the particularities of the system. If *Δl* > *Δl$_c$* , the trajectory is not accepted and more intermediate structures are added to the path. Then, steps 3 and 4 are repeated.

The function $\Theta = \sum(\partial S/\partial \mathbf{Y}_i)^2$ depends on the distances $\Delta l_{i,i+1}$ (Eq. [22]) which are computed as norms in Cartesian space. Therefore, it is important to remove overall translations and rotations from the structures along the trajectory. This can be done by imposing linear constraints using the Eckart conditions[107]:

$$\sum_{j=1,\ldots,L} \mathbf{y}_{ij} = 0 \qquad \sum_{\mathrm{j=1,\ldots,L}} \mathbf{y}^0_{\mathrm{ij}} \times \left(\mathbf{y}_{ij} - \mathbf{y}^0_{ij}\right) = 0 \qquad \forall \mathrm{i} \qquad [24]$$

where the vector $\mathbf{y}_{kl}$ is the mass-weighted Cartesian coordinate of atom *l* in structure *k*. The vectors $\mathbf{y}^0_{ij}$ represent the coordinates of a reference structure.

Equation [24] provides 6*N* linear constraints. We can denote these constraints by σ_{im} with *i* = 1, …, N and *m* = 1, …, 6.

The gradients of the constraints $\nabla\sigma_{il}$ and unit vectors in their directions $\boldsymbol{\eta}_{il}^{0} = \left(\nabla\sigma_{il} / |\nabla\sigma_{il}|\right)$ are coordinate-independent. Therefore, they only need be computed once at the beginning of the calculation. These unit vectors are not necessarily orthogonal for a single structure ($\boldsymbol{\eta}_{il}^{0} \cdot \boldsymbol{\eta}_{ik}^{0} \neq \delta_{lk}$). They can be orthogonalized using a Gram-Schmidt procedure[108]. We can denote this set of orthogonalized vectors by $\{\boldsymbol{\eta}_{il}\}_{i=1}^{N}$.

Let $\{\mathbf{Y}_i\}_{i=1}^{N}$ denote the set of variable coordinates of the current trajectory that satisfies the constraints. Let $\{\delta\mathbf{Y}_i^0\}_{i=1}^{N}$ be a proposed displacement of these coordinates during the optimization process to generate a new trajectory $\{\mathbf{Y}_i + \delta\mathbf{Y}_i^0\}_{i=1}^{N}$. The components of the displacement that satisfies the constraints are given by:

$$\delta\mathbf{Y}_i = \delta\mathbf{Y}_i^0 - \sum_{l=1,\ldots,6} \left(\delta\mathbf{Y}_i^0 \cdot \boldsymbol{\eta}_{il}\right)\boldsymbol{\eta}_{il} \qquad \forall \mathrm{i}. \qquad [25]$$

Then, a new trajectory with coordinates $\{\mathbf{Y}_i + \delta\mathbf{Y}_i\}_{i=1}^{N}$ satisfies the Eckart constraints.

The SDEL algorithm has been efficiently parallelized using message passing interface (MPI) libraries. In the parallelization scheme each node of a cluster of computers calculates the potential energy and derivatives for a particular segment of the path[109]. Inter-node communication is not heavy and the computation scales favorably with cluster size.

There are several advantages of SDEL compared with other methods:

1) *The trajectories can be computed at room temperature or any other temperature of interest and no bias potential is needed.* This differs from methods that use high temperature to accelerate the dynamics[110], or modify the potential energy function to drive the trajectory to a desired outcome[110-112].

2) *Both the boundary conditions and the length parameterization enable study of very slow processes*. This is demonstrated later.

3) *The algorithm is easy to run in parallel with no costly communication between processors*. Ordinary PC clusters can be used.

4) *As in any reaction path method all the trajectories are reactive*. This enhances the efficient use of computational resources. This is in contrast to initial value normal and MTS MD, in which many trajectories do not end at the desired state.

5) *The SDEL formulation is very general*. It is not limited to processes with large energy barriers, single barriers or with exponential kinetics. This makes SDEL more versatile than other reaction paths methods.

6) *The algorithm produces an interpolation between the minimum energy path (MEP) and a true classical trajectory*. Hence, even trajectories with low-resolution can be useful in qualitative reaction path studies.

But the algorithm has several disadvantages:

1) *The trajectories are approximate*. High frequency motions are not resolved. These motions can be important in certain dynamical events.

2) *The computations are expensive*. Computations of trajectories for systems with ~1000 atoms require a parallel resource of near 20 CPU-s at current processor speeds. However, cluster of computers of this size are becoming common in computational chemistry labs.

3) *The length formulation makes it difficult to estimate the timescale of the process*. Because of this limitation SDEL can provide information about the relative sequence of events but not absolute times. This is a limitation shared by all reaction path methods.

4) *Thermodynamic properties are approximate and quantitative kinetic properties are inaccessible*. The filtering out of high frequency modes due to the large step size affects calculations of thermodynamic properties and transition probabilities. However, it has been shown that enthalpic properties of slow variables are affected only slightly[93].

5) *The final solution depends on the initial guess for the trajectory*. For a large system, no global optimization protocol will generate the true minimum for the target function Θ in an acceptable time. In the applications of SDEL, the initial guess is an approximate MEP obtained with a self-penalty walk algorithm and most of the solutions obtained correspond to trajectories in local minimums near this initial guess. Implementation of more unbiased procedures, sampling trajectories connecting structures in configuration space, is a subject of ongoing research[113].

6) The current implementation of the algorithm uses implicit solvent model. This is not an essential limitation of SDEL and computations of trajectories with an atomistic description of the aqueous environment are possible. An assumption of time separation between the relaxation of the water molecules to equilibrium and the solvated molecule (for example, a protein) can make the calculations feasible. Then the configuration for the water molecules can be determined using thermal distribution for a fixed configuration of the molecule. A short MD simulation can be used to this end. This procedure was used before to include explicit water dynamics in the Onsager-Machlup action[96, 109]. However, inclusion of explicit solvent using this adiabatic approximation makes the computations slower for large systems. A simpler way to include the effect of explicit water/molecule interactions entails the extraction of configuration snapshots from SDEL trajectories. Then, these structures can be immersed into a box with explicit water molecules and MD simulations can be performed until equilibration is reached[114].

It is clear that the SDEL algorithm has very appealing advantages when it is applied to long time events. Meaningful trajectories can be obtained for processes that are difficult or impractical to study by initial-value formulations and other reaction path techniques. However, the lack of kinetic information makes SDEL an alternative to other algorithms that can provide transition probabilities, albeit with limited time scale, like TPS, DPS and string methods.

Use of SDEL to compute reactive trajectories: input parameters, initial guess, paralelization protocol

The SDEL algorithm is implemented in MOIL, a suite of molecular simulation programs developed in the group of Ron Elber. Linux and windows versions of the software package can be downloaded, free of charge from his group website at http://cbsu.tc.cornell.edu/software/moil/moil.html. The windows version has a graphics interface that is relatively easy to use. The linux version can only be run in a command mode after installation using a makefile command. The SDEL program can be run on a standalone computer or on a parallel cluster of PC's using MPI protocols. As it was mentioned before, the optimization of the action is computationally expensive. Therefore the use of a parallel computer is recommended for most applications dealing with systems with more than ~100 atoms. The inter-node communication required is low and the computation scales well with an increase of the number of nodes[109].

To compute a trajectory with SDEL, the two conformations at the boundaries are required. For example, to study the mechanism of folding of a protein, the initial unfolded conformation in the trajectory can be extracted from high temperature molecular dynamics simulations of the protein under consideration. The final folded structure in the trajectory can be taken from the protein data bank after equilibrium MD of the native configuration.

The total energy of the molecular system is also required by SDEL. The most common way to estimate this energy is by performing room temperature MD of the system, with the same solvation model and force-field parameters to be used during the SDEL run.

An initial guess for the trajectory connecting the two boundary states needs to be generated to evaluate the SDEL action, Θ. This is the most troublesome step of the algorithm. The SDEL action depends on $\sqrt{2(E-V(\mathbf{X}))}$. This suggests that each conformation in the initial trajectory must have a potential energy smaller than the total energy of the system. This prevents the use of simple methods to compute the initial trajectory like linear interpolation of the two boundary structures. In general, linear interpolation will generate intermediate structures with large potential energy due to steric repulsions.

In practice, this problem is solved by computing an approximate MEP connecting the two boundary states and using this path as initial guess for SDEL. MOIL contains a program called *chmin* that computes MEP using a simple self-penalty walk algorithm[74]. *Chmin* treats the path as a polymer chain, where each monomer is a copy of the molecular system at different times. The potential energy of the chain is the sum of the potential energies of each monomer, with the addition of a harmonic attraction term between nearest neighbors, and an exponential repulsion term between next nearest neighbors.

The potential energy for the structures $V(\mathbf{Y}_i)$ in the MEP should be analyzed before using this trajectory as initial guess for SDEL. Not only must the potential energy of every slice be lower than the total energy of the system but the potential energy should vary smoothly in the trajectory. It has been observed that steep peaks or decays of $V(Y_i)$ cause numerical instability during the optimization of the action.

The input file for SDEL, called path.inp in MOIL (Table 1), contains the names for the file (*rcrd*) with the initial guess trajectory (this is a binary file with extension *pth*), and the connectivity file used to extract the potential parameters for the molecule. The connectivity file (with extension *wcon*) is generated by running the program *conn* in Moil.

The SDEL target function for optimization in Moil is:

$$\Theta = \sum \left(\partial S / \partial \mathbf{Y}_i\right)^2 + \sum_{i=0}^{N} \gamma \left(\Delta l_{i,i+1} - \langle \Delta l \rangle\right)^2 \qquad [26]$$

with $\langle \Delta l \rangle$ the average value of the distances $\Delta l_{i,i+1}$ between structures in the path. The second term on the right side enforces equidistance of structures in the path. The parameter γ is a constant that can be adjusted in the input file (*gama* in path.inp) to optimize calculation efficiency.

A simulated annealing protocol can be used to optimize the target function subjected to the overall translation and rotation constraints (Equation [24]). We can denote the variable components of the initial guess for the trajectory $\left\{\mathbf{Y}_i^0\right\}_{i=1}^{N}$ and optimize the trajectory for K steps solving the second-order differential equation for the trajectory $\mathbf{Z} = \left\{\mathbf{Y}_i\right\}_{i=1}^{N}$:

$$\frac{d^2\mathbf{Z}}{d\tau^2} = -\nabla_{\mathbf{Z}}\Theta \qquad [27]$$

Using linear cooling with velocity scaling $\left|d\mathbf{Z}/d\tau\right|^2 = \mu(\theta - \tau)$, where τ is a fictitious time during the annealing run, θ is the total time (in practice, $\theta = K\Delta\tau$) and μ is a factor proportional to the initial temperature used at the beginning of an annealing cycle. In the input file, the flag *anne* instructs the program to use simulated annealing, *tmpr* gives the value of μ, and *dtop* is the time step $\Delta\tau$.

The SDEL program can also use a conjugate gradient Powell algorithm[115] to minimize the target function. This algorithm is more efficient searching for local minima. Therefore, it should be used only if it is believed that the initial guess for the trajectory is near the global minimum. To use conjugate gradient the *anne* flag must be removed in path.inp.

Other parameters that control the performance of a SDEL run (Table 1) are:

#ste = provides the total number of optimization steps.

list = gives the total number of steps in each cycle of optimization (the value of *K*). The program also writes useful information every *list* steps.

grid = this is the total number of structures in the trajectory, i.e., *grid = N+2.*

pdqe = provides the total energy for the molecular system in *Kcal/mol.*

gbsa = uses a generalized Born model for the solvent environment.

rmax, epsi, v14f, el14 are the values for the cutoff distance for nonbonded interactions, dielectric constant and 1-4 scaling for van der Waals and electrostatics interactions respectively.

proc = the number of nodes use

cpth = the program reads the trajectory coordinates (rcrd) using a binary path format.

action = instructs the program to continue execution.

During execution, the program generates log files with information about the status of the run on each node. The log file associated with the master node (called pth_out_0000.log) gives the gradient of Θ every *list* steps. Typically a value of this gradient of 10 *(Kcal/mol)*^{2} or less produces a convergent trajectory. Convergence can be analyzed by comparing intermediate results for the paths which are written after each cycle of optimization.

At the end of a run, the final trajectory is output (*file wcrd*) with path format. Moil contains several programs that can be used to analyze the trajectory (computation of radius of gyration, native contacts, secondary structure content, etc). The windows graphics interface of Moil can be used to visualize the trajectory. Also, the path format can be converted to a more conventional format like *dcd* using the *ccrd* program of Moil. *Dcd* files can be open by many molecular visualization programs like VMD[116].

The most common error message during a SDEL run is "*Our momentum is < 0*". This occurs any time the potential energy is larger than the total energy during the optimization process. Changes of the annealing parameters or the value of γ often fix this problem.

Examples of SDEL runs can be found in the directory moil.test in Moil.

Applications of the stochastic difference equation in length

The SDEL algorithm has been used to study the folding dynamics of several peptides and protein systems. In these applications the solvent environment has been treated implicitly using the Generalized Born model[117, 118]. The algorithm was first applied to study the folding of the B domain fragment of the *Staphylococcal* protein A[92]. This 60-residue three-helical protein has been studied by many groups using different computational strategies[102]. A recent experimental assessment of the transition state for this folding process suggests the difficulties of atomic simulations in capturing all the features observed in the experiment[119]. The results from SDEL were similar to the results obtained using high-temperature MD simulations showing early formation of the most stable helix. The experiment indicates that the other two helices are more involved in the early folding. This points to the existence of some energetic frustration during the folding of this protein[120].

SDEL was also used to study the coil-helix transition of an alanine-rich peptide[93], the conformational transition of sugar puckering in deoxyadenosine[121], polymerase P[122], and the B-Z DNA transition[123]. The coil to helix study[93] demonstrated several properties of SDEL trajectories, like the filtering of high frequency modes and the preservation of thermodynamics properties for slow degrees of freedom when the trajectory resolution is decreased.

An interesting application of SDEL examined the folding mechanism of cytochrome c[124]. The folding kinetics of cytochrome c has been extensively studied experimentally by a variety of techniques[125-129]. The analysis of SDEL folding trajectories was in agreement with several experimental observations: the collapse of the protein without formation of secondary structures followed by formation of the terminal helices before the middle helix (see upper side of Figure 8). Then a molten globule conformation is formed. The structural features of this molten globule conformation (lower right side of Figure 8) are in agreement with fluorescence energy transfer experiments[130]. Finally, more rearrangements occur before the protein folds to its native conformation (lower left side Figure 8).

The SDEL algorithm has also been applied to study the folding of more complicated systems, such as the wild type human Cu, Zn superoxide dismutase (SOD) dimer. SOD is a 153-residue, homodimeric, anti-oxidant enzyme that dismutates superoxide ion to hydrogen peroxide and oxygen[131]. It is an eightstrand, flattened, beta-barrel protein with one copper and one zinc ion per monomer[132]. This protein is involved in the Familial form of Amyotrophic Lateral Sclerosis (FALS).

A 1.8Å resolution Apo-SOD crystal structure (PDB 1HL4[133]) was used to generate SDEL trajectories[113] of monomer folding and dimerization (Figure 9). Initial analysis of a pair of trajectories showed a small population of folded but separated monomers. Interestingly, approximately 15-20% of each monomer's intra-subunit native contacts form when the subunit centers-of-mass are within a few angstroms of equilibrium position, with the rest of the native contacts forming when the monomers are farther away.

Recent advances and challenges

These applications demonstrate the potential of the SDEL algorithm as a tool to study conformational dynamics of large molecular systems such as peptides and proteins at longer timescales. This is the only algorithm from the methods discussed in this chapter that can be used to compute trajectories for such complex processes that take milliseconds or longer, such as the folding of cytochrome c or the SOD dimer. Although the trajectories are approximate they can provide structural data to explain experimental observations. If a more detailed and accurate description is required, snapshots taken from these SDEL trajectories can be used to extract thermodynamic information using MD, umbrella sampling or replica exchange methodologies.

The major limitation of the SDEL algorithm is the inaccessibility of absolute times and computation of rates. A promising algorithm called Milestoning has been proposed recently to overcome this difficulty[134]. This method computes a non-Markovian hopping between configuration space hyperplanes, the so-called "milestones". The assumption is that there is an equilibrium distribution on each milestone. The kinetics is obtained by starting an equilibrium configuration on a milestoning, and measuring the time distribution needed to reach the forward or backward milestones using short MD simulations. These time distributions are then used to compute the global kinetics through a non-Markovian model. The algorithm needs a reaction coordinate to define the hyperplanes. This reaction coordinate can be a MEP or a SDEL trajectory. The equilibrium sampling is performed in the neighborhood of the curvilinear path describing the reaction coordinate. This approach can also be used to compute free energy profiles along the reaction coordinate[135, 136]. The Milestoning method depends on the assumption that there is only one reaction coordinate (slow variable) in the system. The correctness of the approach can be assessed by monitoring the rate as a function of the separation between the milestones.

A similar algorithm is partial path transition interface sampling (PPTIS)[137]. This method is based on transition interface sampling (TIS)[61, 67], which maps the phase space of the system with many interfaces characterized by a one-dimensional reaction coordinate. In PPTIS rates are computed using a Markovian state model, i.e., assuming a loss of correlation during interface hopping. This algorithm is aimed at computing the kinetics of two-state exponential process in equilibrium.

Use of Milestoning or PPTIS could provide a way to recover the information that is lost when the high-frequency motions of the molecular system are filtered out by a SDEL trajectory. Hence, correct kinetic and thermodynamic properties can be extracted from the simulations. For very long and diffusive processes, like the ones associated with the folding of large proteins, computation of these properties still be challenging because the transitions between hyperplanes or interfaces will require longer MD simulations. At that point, a combination of MTS with these path methods could improve efficiency and speed.

CONCLUSION

Computational methods to extend the timescale of atomically detailed simulations have improved in the last 15 years. Accurate MTS simulations, with computational gains up to a factor of 10, have extended the applicability of molecular dynamics simulations. Additional refinements in the computation of medium-range forces could provide stable results with increasing speedups. However, it is apparent that fundamental limitations will prevent the extension of these algorithms to the range of timescales that are needed to study many processes of interest. On the other hand, reaction path approaches can be used to investigate slower processes by using different simplifying assumptions and approximations (for example, two-state transitions, equilibrium distributions along the path or removal of high-frequency motions). Therefore, path methodologies extend the range of application of computer simulation but with lost of accuracy and dynamical information compared with normal and MTS molecular dynamics simulations. At the end, some path techniques utilize MD to compute rate constants and recover part of the dynamical properties filtered out. In the future, use of MTS should improve the efficiency and practicality of these calculations.

ACKNOWLEDGEMENT

This material is based in part upon work supported by the National Science Foundation under Grant No. 0447294 to A.E.C. The same author would like to acknowledge the use of the services provided by Research Computing, University of South Florida, and the help of Shawn Hamm and Adam Clarke with some of the figures.

REFERENCES

1. A. Rahman, *Phys. Rev. A,* **136**, 405 (1964). Correlations in the motion of atoms in liquid argon.

2. L. Verlet, *Phys. Rev.,* **159**, 98 (1967). Computer "experiments" on classical fluids.

3. A. Rahman, F. H. Stillinger, *J. Chem. Phys.,* **55**, 3336 (1971). Molecular dynamics study of liquid water.

4. J. P. Ryckaert, G. Ciccotti, H. J. C. Berendsen, *J. Comp. Phys.,* **23**, 327 (1977). Numerical integration of the Cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes.

5. J. A. McCammon, B. R. Gelin, M. Karplus, *Nature,* **267**, 585 (1977). Dynamics of folded proteins.

6. R. D. Ruth, *IEEE Trans. Nucl. Sci.,* **30** 2669 (1983). A canonical integration technique.

7. F. M. Lasagni, unpublished work, 1990.

8. G. Benettin, A. Giorgilli, *J. Statist. Phys.,* **74**, 1117 (1994). On the Hamiltonian Interpolation of Near to the Identity Symplectic Mappings.

9. E. Hairer In *Backward analysis of numerical integrators and symplectic methods*, SCADE'93 conference, Auckland, New-Zealand, 1994; J.C.; Baltzer., Eds. Annals of Numer. Math.: Auckland, New-Zealand, 1994; pp 107.

10. J. M. Sanz-Serna, M. P. Calvo, *Numerical Hamiltonian problems*. Chapman and Hall, London, 1994.

11. R. J. Loncharich, B. R. Brooks, R. W. Pastor, *Biopolymers,* **21**, 523 (1992). Langevin dynamics of peptides: The frictional dependence of isomerization rates of N-Acetylalanyl-N'-Methylamide.

12. A. Brunger, I. Brooks, C. L., M. Karplus, *Chem. Phys. Lett.,* **105**, 495 (1982). Stochastic Boundary conditions for molecular dynamics simulations of ST2 water.

13. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, a. M. Karplus, *J. Comp. Chem.,* **4**, 187 (1983). CHARMM: A program for macromolecular energy, minimization, and dynamics calculations.

14. A. D. MacKerell Jr., M. B. D. Bashford, R. L. D. Jr., J. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. R. III, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, a. M. Karplus, *J. Phys. Chem.,* **102**, 3586 (1998). An all-atom empirical potential for molecular modeling and dynamics of proteins.

15. S. J. Weiner, P. A. Kollman, D. T. Nguyen, D. A. Case, *J. Comp. Chem.,* **7**, 230 (1986). An all atom force field for simulations of proteins and nucleic acids.

16. W. D. Cornell, P. Cieplak, C. I. Bayly, I.R. Gould, J. K.M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, P. A. Kollman, *J. Am. Chem. Soc.,* **117**, 5179 (1995). A second generation force field for the simulation of proteins and nucleic acids.

17. W. L. Jorgensen, J. Tirado-Rives, *J. Am. Chem. Soc.,* **1988**, 1657 (1988). The OPLS potential functions for proteins. Energy minimization for crystals of cyclic peptides and crambin.

18. E. Barth, B. Leimkuhler, S. Reich, *Lecture Notes in Computational Science and Engineering,* **24**, 73 (2002). A Test Set for Molecular Dynamics.

19. T. Schlick, *Optimization methods in computational chemistry*. VCH Publishers, New York, 1992; Vol. 3.

20. W. B. Streett, D. J. Tildesley, G. Saville, *Mol. Phys.,* **35**, 639 (1978). Multiple time step methods in molecular dynamics.

21. R. R. Gabdoulline, C. Zheng, *J. Comp. Chem.,* **16**, 1428 (1995). Effects of the cutoff center on the mean potential and pair distribution functions in liquid water.

22. M. Saito, *J. Chem. Phys.,* **101**, 4055 (1994). Molecular dynamics simulations of proteins in solution: Artifacts caused by the cutoff approximation.

23. H. Grubmuller, H. Heller, A. Windemuth, K. Schulten, *Mol. Sim.,* **6**, 121 (1991). Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions.

24. M. Tuckerman, B. J. Berne, G. J. Martyna, *J. Chem. Phys.,* **97**, 1990 (1992). Reversible Multiple Time Scale Molecular-Dynamics.

25. R. Zhou, B. J. Berne, *J. Chem. Phys.,* **103**, 9444 (1995). A new molecular dynamics method combining the reference system propagator algorithm with a fast multipole method for simulating proteins and other complex systems.

26. M. Watanabe, M. Karplus, *J. Phys. Chem.,* **99**, 5680 (1995). Simulations of macromolecules by multiple time-step methods.

27. J. J. Biesiadecki, R. D. Skeel, *J. Comp. Phys.,* **109**, 318 (1993). Dangers of multiple time step methods.

28. T. Bishop, R. D. Skeel, K. Schulten, *J. Comp. Chem.,* **18**, 1785 (1997). Difficulties with multiple time stepping and the fast multipole algorithm in molecular dynamics.

29. Q. Ma, J. A. Izaguirre, R. D. Skeel, *SIAM J. Sci. Comput.,* **24**, 1951 (2003). Verlet-I/r-RESPA is limited by nonlinear instability.

30. T. Schlick, M. Mandziuk, R. D. Skeel, K. Srinivas, *J. Comp. Phys.,* **139**, 1 (1998). Nonlinear resonance artifacts in molecular dynamics simulations.

31. T. Schlick, E. Barth, M. Mandziuk, *Ann. Rev. Biophys. Biomol. Struc.,* **26**, 179 (1997). Biomolecular dynamics at long timesteps: Bridging the timescale gap between simulation and experimentation.

32. J. A. Izaguirre, S. Reich, R. D. Skeel, *J. Chem. Phys.,* **110**, 9853 (1999). Longer time steps for molecular dynamics.

33. J. A. Izaguirre, Qun Ma, T. Matthey, J. Willcock, T. Slabach, B. Moore, G. Viamontes, *Lecture Notes in Computational Science and Engineering (LNCSE),* **24**, 146 (2002). Overcoming instabilities in Verlet-I/r-RESPA with the mollified impulse method.

34. E. Barth, T. Schlick, *J. Chem. Phys.,* **109**, 1633 (1998). Extrapolation versus Impulse in Multiple-Timestepping Schemes: Linear Analysis and Applications to Newtonian and Langevin Dynamics.

35. E. Barth, T. Schlick, *J. Chem. Phys.,* **109**, 1617 (1998). Overcoming stability limitations in biomolecular dynamics: Combining force splitting via extrapolation with Langevin dynamics in LN.

36. A. Sandu, T. Schlick, *J. Comp. Phys.,* **151**, 74 (1999). Masking Resonance Artifacts in Force-Splitting Methods for Biomolecular Simulations by Extrapolative Langevin Dynamics.

37. M. P. Allen, D. J. Tildesley, *Computer simulation of liquids*. Oxford Science Publications, Oxford, 1987.

38. C. S. Peskin, T. Schlick, *Comm. Pure Appl. Math.,* **42**, 1001 (1989). Molecular Dynamics by the Backwards-Euler Method.

39. G. Zhang, T. Schlick, *J. Comp. Chem.,* **14**, 1212 (1993). LIN: A New Algorithm Combining Implicit Integration and Normal Mode Techniques for Molecular Dynamics.

40. J. A. Izaguirre, D. P. Catarello, J. M. Wozniak, R. D. Skeel, *J. Chem. Phys.,* **114**, 2090 (2001). Langevin stabilization of molecular dynamics.

41. Q. Ma, J. A. Izaguirre In *Long time step molecular dynamics using targeted Langevin stabilization*, ACM Symposium on Applied Computing SAC 03, 2003; 2003; pp 178.

42. R. D. Skeel, J. A. Izaguirre, *Mol. Phys.,* **100**, 3885 (2002). An Impulse Integrator for Langevin Dynamics.

43. J. A. Izaguirre, in *Multiscale Computational Methods in Chemistry and Physics*, A. Brandt, K. B., and J. Bernholc, Ed. IOS Press: Amsterdam, 2001; Vol. 177. Langevin Stabilization of Multiscale Mollified Molecular Dynamics.

44. T. Schlick, L. Yang, in *Multiscale Computational Methods in Chemistry and Physics*, A. Brandt, J. B. a. K. B., Ed. Amsterdam, 2001; Vol. 177. Long-Timestep Biomolecular Dynamics Simulations: LN Performance on a Polymerase Beta / DNA System.

45. L. Greengard, V. Rokhlin, *J. Comp. Phys.,* **73**, 325 (1987). A fast algorithm for particle simulations.

46. A. W. Appel, *SIAM J. Sci Stat. Comput.,* **6**, 85 (1985). An efficient program for many-body simulations.

47. J. Barnes, P. Hut, *Nature,* **324**, 446 (1986). A hierarchical O(N log N) force calculation algorithm.

48. Z.-H. Duan, R. Krasny, *J. Comp. Chem.,* **21**, 1 (2000). An adaptive treecode for computing nonbonded potential energy in classical molecular systems.

49. T. Darden, D. York, L. Pedersen, *J. Chem. Phys.,* **98**, 10089 (1993). Particle mesh Ewald: an N*log(N) method for computing Ewald sums.

50. Z.-H. Duan, R. Krasny, *J. Chem. Phys.,* **113**, 3492 (2000). An Ewald summation based multipole method.

51. R. W. Hockney, J. W. Eastwood, *Computer simulation using particles*. McGraw-Hill, New York, 1981.

52. P. Procacci, M. Marchi, *J. Chem. Phys.,* **104**, 3003 (1996). Taming the Ewald sum in molecular dynamics simulations of solvated proteins via a multiple time step algorithm.

53. X. Qian, T. Schlick, *J. Chem. Phys.,* **116**, 5971 (2002). Efficient Multiple Timestep Integrators with Distance-Based Force Splitting for Particle-Mesh-Ewald Molecular Dynamics Simulations.

54. D. Barash, L. Yang, X. Qian, T. Schlick, *J. Comp. Chem.,* **24**, 77 (2003). Inherent Speedup Limitations in Multiple Timestep Particle Mesh Ewald Algorithms.

55. E. Barth http://kzoo.edu/~barth/mtsreview

56. T. Hansson, C. Oostenbrink, W. F. van Gunsteren, *Curr. Opin. Struct. Biol.,* **12**, 190 (2002). Molecular dynamics simulations.

57. M. Karplus, *Acc. Chem. Res.,* **35**, 321 (2002). Molecular dynamics simulations of biomolecules.

58. M. Karplus, J. Kuriyan, *Proc. Natl. Acad. Sci. U. S. A.,* **102**, 6679 (2005). Molecular dynamics and protein function.

59. P. G. Bolhuis, D. Chandler, C. Dellago, P. L. Geissler, *Annu. Rev. Phys. Chem.,* **53**, 291 (2002). Transition path sampling: Throwing ropes over rough mountain passes, in the dark.

60. P. G. Bolhuis, C. Dellago, D. Chandler, *Faraday Discussions*, 421 (1998). Sampling ensembles of deterministic transition pathways.

61. T. S. van Erp, D. Moroni, P. G. Bolhuis, *J. Chem. Phys.,* **118**, 7762 (2003). A novel path sampling method for the calculation of rate constants.

62. P. G. Bolhuis, *Proc. Natl. Acad. Sci. U. S. A.,* **100**, 12129 (2003). Transition-path sampling of beta-hairpin folding.

63. T. S. van Erp, P. G. Bolhuis, *J. Comput. Phys.,* **205**, 157 (2005). Elaborating transition interface sampling methods.

64. P. G. Bolhuis, *J. Phys.: Condens. Matter,* **15**, S113 (2003). Transition path sampling on diffusive barriers.

65. C. Dellago, P. G. Bolhuis, F. S. Csajka, D. Chandler, *J. Chem. Phys.,* **108**, 1964 (1998). Transition path sampling and the calculation of rate constants.

66. J. Juraszek, P. G. Bolhuis, *Proc. Natl. Acad. Sci. U. S. A.,* **103**, 15859 (2006). Sampling the multiple folding mechanisms of Trp-cage in explicit solvent.

67. P. G. Bolhuis, *Biophys. J.,* **88**, 50 (2005). Kinetic pathways of beta-hairpin (Un)folding in explicit solvent.

68. P. G. Bolhuis, D. Chandler, *J. Chem. Phys.,* **113**, 8154 (2000). Transition path sampling of cavitation between molecular scale solvophobic surfaces.

69. C. Dellago, P. G. Bolhuis, D. Chandler, *J. Chem. Phys.,* **108**, 9236 (1998). Efficient transition path sampling: Application to Lennard-Jones cluster rearrangements.

70. M. F. Hagan, A. R. Dinner, D. Chandler, A. K. Chakraborty, *Proc. Natl. Acad. Sci. U. S. A.,* **100**, 13922 (2003). Atomistic understanding of kinetic pathways for single base-pair binding and unbinding in DNA.

71. P. L. Geissler, C. Dellago, D. Chandler, *Phys. Chem. Chem. Phys.,* **1**, 1317 (1999). Chemical dynamics of the protonated water trimer analyzed by transition path sampling.

72. R. Radhakrishnan, T. Schlick, *Proc. Natl. Acad. Sci. U. S. A.,* **101**, 5970 (2004). Orchestration of cooperative events in DNA synthesis and repair

mechanism unraveled by transition path sampling of DNA polymerase beta's closing.

73. M. Berkowitz, J. D. Morgan, J. A. McCammon, S. H. Northrup, *J. Chem. Phys.,* **79**, 5563 (1983). Diffusion-controlled reactions: A variational formula for the optimum reaction coordinate.

74. R. Czerminski, R. Elber, *Int. J. Quantum Chem.*, 167 (1990). Self-Avoiding Walk between 2 Fixed-Points as a Tool to Calculate Reaction Paths in Large Molecular-Systems.

75. S. H. Huo, J. E. Straub, *J. Chem. Phys.,* **107**, 5000 (1997). The MaxFlux algorithm for calculating variationally optimized reaction paths for conformational transitions in many body systems at finite temperature.

76. J. E. Straub, J. Guevara, S. H. Huo, J. P. Lee, *Acc. Chem. Res.,* **35**, 473 (2002). Long time dynamic simulations: Exploring the folding pathways of an Alzheimer's amyloid A beta-peptide.

77. R. Crehuet, M. J. Field, *J. Chem. Phys.,* **118**, 9563 (2003). A temperature-dependent nudged-elastic-band algorithm.

78. R. Crehuet, A. Thomas, M. J. Field, *J. Mol. Graph. Model.,* **24**, 102 (2005). An implementation of the nudged elastic band algorithm and application to the reaction mechanism of HGXPRTase from Plasmodium falciparum.

79. D. A. Evans, D. J. Wales, *J. Chem. Phys.,* **119**, 9947 (2003). The free energy landscape and dynamics of met-enkephalin.

80. D. A. Evans, D. J. Wales, *J. Chem. Phys.,* **121**, 1080 (2004). Folding of the GB1 hairpin peptide from discrete path sampling.

81. D. J. Wales, *Int. Rev. Phys. Chem.,* **25**, 237 (2006). Energy landscapes: calculating pathways and rates.

82. D. J. Wales, T. V. Bogdan, *J. Phys. Chem. B,* **110**, 20765 (2006). Potential energy and free energy landscapes.

83. G. Henkelman, G. Johannesson, H. Jónsson, in *Theoretical Methods in Condensed Phase Chemistry*, Schwartz, S. D., Ed. Kluwer Academic: Dordrecht, 2000; Vol. 5, pp 269. Methods for finding saddle points and minimum energy paths.

84. S. A. Trygubenko, D. J. Wales, *J. Chem. Phys.,* **120**, 2082 (2004). A doubly nudged elastic band method for finding transition states.

85. E. Weinan, W. Q. Ren, E. Vanden-Eijnden, *J. Phys. Chem. B,* **109**, 6688 (2005). Finite temperature string method for the study of rare events.

86. E. Weinan, W. Q. Ren, E. Vanden-Eijnden, *Chem. Phys. Lett.,* **413**, 242 (2005). Transition pathways in complex systems: Reaction coordinates, isocommittor surfaces, and transition tubes.

87. W. Ren, E. Vanden-Eijnden, P. Maragakis, W. N. E, *J. Chem. Phys.,* **123**, 134109 (2005). Transition pathways in complex systems: Application of the finite-temperature string method to the alanine dipeptide.

88. L. Maragliano, A. Fischer, E. Vanden-Eijnden, G. Ciccotti, *J. Chem. Phys.,* **125**, 024106 (2006). String method in collective variables: Minimum free energy paths and isocommittor surfaces.

89. L. Maragliano, E. Vanden-Eijnden, *Chem. Phys. Lett.,* **426**, 168 (2006). A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations.

90. W. E, E. Vanden-Eijnden, *J. Stat. Phys.,* **123**, 503 (2006). Towards a theory of transition paths.

91. P. Metzner, C. Schutte, E. Vanden-Eijnden, *J. Chem. Phys.,* **125**, 084110 (2006). Illustration of transition path theory on a collection of simple examples.

92. R. Elber, A. Ghosh, A. Cardenas, in *Bridging the time scale gap*, Nielaba, P.; Mareschal, M.; Ciccotti, G., Eds. Springer Verlag: Berlin, 2002. Stochastic Difference Equation as a tool to compute long time dynamics.

93. A. E. Cardenas, R. Elber, *Biophys. J.,* **85**, 2919 (2003). Atomically detailed Simulations of helix formation with the stochastic difference equation.

94. S. Machlup, L. Onsager, *Phys. Rev,* **91**, 1512 (1953). Fluctuations and Irreversible Process .2. Systems with Kinetic Energy.

95. L. Onsager, S. Machlup, *Phys. Rev,* **91**, 1505 (1953). Fluctuations and Irreversible Processes.

96. R. Elber, J. Meller, R. Olender, *J. Phys. Chem. B,* **103**, 899 (1999). Stochastic path approach to compute atomically detailed trajectories: Application to the folding of C peptide.

97. R. Olender, R. Elber, *J. Chem. Phys.,* **105**, 9299 (1996). Calculation of classical trajectories with a very large time step: Formalism and numerical examples.

98. K. Siva, R. Elber, *Proteins: Struct., Funct., Genet.,* **50**, 63 (2003). Ion permeation through the gramicidin channel: Atomically detailed modeling by the Stochastic Difference Equation.

99. D. Passerone, M. Parrinello, *Phys. Rev. Lett.,* **8710**, 108302 (2001). Action-derived molecular dynamics in the study of rare events.

100. D. Passerone, M. Ceccarelli, M. Parrinello, *J. Chem. Phys.,* **118**, 2025 (2003). A concerted variational strategy for investigating rare events.

101. D. Bai, R. Elber, *J. Chem. Theory Comput.,* **2**, 484 (2006). Calculation of point-to-point short-time and rare trajectories with boundary value formulation.

102. J. E. Shea, M. R. Friedel, A. Baumketner, in *Reviews in Computational Chemistry*, Lipkowitz, K. B.; Cundari, T. R.; Griller, V. J., Eds. John Wiley and Sons, Inc.: New York, 2006; Vol. 22, pp 169. Simulations of Protein Folding.

103. L. D. Landau, E. M. Lifshitz, *Mechanics*. Butterworth-Heinenann, Oxford, 1976.

104. R. Elber, in *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology*, Sringer-Verlag: Berlin, 2006; Vol. 1, pp 435. Calculation of Classical trajectories with Boundary Value Formulation.

105. A. Ulitsky, R. Elber, *J. Chem. Phys.,* **92**, 1510 (1990). A New Technique to Calculate Steepest Descent Paths in Flexible Polyatomic Systems.

106. R. Elber, A. Roitberg, C. Simmerling, R. Goldstein, H. Y. Li, G. Verkhivker, C. Keasar, J. Zhang, A. Ulitsky, *Comput. Phys. Commun.,* **91**, 159 (1995). Moil - a Program for Simulations of Macromolecules.

107. R. Elber, *J. Chem. Phys.,* **93**, 4312 (1990). Calculation of the potential of mean force using molecular dynamics with linear constraints: An application to a conformational transition in a solvated dipeptide.

108. R. Czerminsky, R. Elber, *J. Chem. Phys.,* **92**, 5580 (1990). Reaction path study of conformational transitions in flexible systems: application to peptides.

109. V. Zaloj, R. Elber, *Comput. Phys. Commun.,* **128**, 118 (2000). Parallel computations of molecular dynamics trajectories using the stochastic path approach.

110. A. F. Voter, F. Montalenti, T. C. Germann, *Annu. Rev. Mat. Res.,* **32**, 321 (2002). Extending the time scale in atomistic simulation of materials.

111. S. Izrailev, S. Stepaniants, B. Isralewitz, D. Kosztin, H. Lu, F. Molnar, W. Wriggers, K. Schulten, in *Computational Molecular Dynamics: Challenges, Methods, Ideas*, Deuflhard, P.; Hermans, J.; Leimkuhler, B.; Mark, A. E.; Reich, S.; Skeel, R. D., Eds. Springer-Verlag: Berlin, 1998; pp 39. Steered Molecular Dynamics.

112. J. Schlitter, M. Engels, P. Kruger, E. Jacoby, A. Wollmer, *Mol. Sim.,* **10**, 291 (1993). A targeted molecular dynamics simulation of conformational change: Application to the T↔ R transition in insulin.

113. A. E. Cardenas, unpublished work, 2007.

114. F. Pitici, R. Elber, Computer simulations of folding of cytochrome c variants. In *Biophysical Society 50th Annual Meeting*, Salt Lake City, Utah, 2006.

115. W. H. Press, B. P. Flannery, S. A. Teukosky, W. T. Vetterling, *Numerical Recipes*. Cambridge University Press, Cambridge, 1986.

116. W. Humphrey, A. Dalke, K. Schulten, *J. Mol. Graph.,* **14**, 33 (1996). VMD: Visual molecular dynamics.

117. V. Tsui, D. A. Case, *Biopolymers,* **56**, 275 (2000). Theory and applications of the generalized Born solvation model in macromolecular Simulations.

118. G. D. Hawkins, C. J. Cramer, D. G. Truhlar, *Chem. Phys. Lett.,* **246**, 122 (1995). Pairwise Solute Descreening of Solute Charges from a Dielectric Medium.

119. S. Sato, T. L. Religa, V. Daggett, A. R. Fersht, *Proc. Natl. Acad. Sci. U. S. A.,* **101**, 6952 (2004). Testing protein-folding simulations by experiment: B domain of protein A.

120. P. G. Wolynes, *Proc. Natl. Acad. Sci. U. S. A.,* **101**, 6837 (2004). Latest folding game results: Protein A barely frustrates.

121. K. Arora, T. Schlick, *Chem. Phys. Lett.,* **378**, 1 (2003). Deoxyadenosine sugar puckering pathway simulated by the stochastic difference equation algorithm.

122. K. Arora, T. Schlick, *J. Phys. Chem. B,* **109**, 5358 (2005). Conformational transition pathway of polymerase ss/DNA upon binding correct incoming substrate.

123. W. Lim, Y. P. Feng, *Biopolymers,* **78**, 107 (2005). Applying the stochastic difference equation to DNA conformational transitions: A study of B-Z and B-A DNA transitions.

124. A. E. Cardenas, R. Elber, *Proteins: Struct., Funct., Genet.,* **51**, 245 (2003). Kinetics of cytochrome C folding: Atomically detailed simulations.

125. H. Roder, G. A. Elove, S. W. Englander, *Nature,* **335**, 700 (1988). Structural Characterization of Folding Intermediates in Cytochrome-C by H-Exchange Labeling and Proton Nmr.

126. G. A. Elove, A. F. Chaffotte, H. Roder, M. E. Goldberg, *Biochemistry,* **31**, 6876 (1992). Early Steps in Cytochrome-C Folding Probed by Time-Resolved Circular-Dichroism and Fluorescence Spectroscopy.

127. L. Pollack, M. W. Tate, N. C. Darnton, J. B. Knight, S. M. Gruner, W. A. Eaton, R. H. Austin, *Proc. Natl. Acad. Sci. U. S. A.,* **96**, 10115 (1999). Compactness of the denatured state of a fast-folding protein measured by submillisecond small-angle x-ray scattering.

128. S. J. Hagen, W. A. Eaton, *J. Mol. Biol.,* **297**, 781 (2000). Two-state expansion and collapse of a polypeptide.

129. S. Akiyama, S. Takahashi, T. Kimura, K. Ishimori, I. Morishima, Y. Nishikawa, T. Fujisawa, *Proc. Natl. Acad. Sci. U. S. A.,* **99**, 1329 (2002). Conformational landscape of cytochrome c folding studied by microsecond-resolved small-angle x-ray scattering.

130. J. G. Lyubovitsky, H. B. Gray, J. R. Winkler, *J. Am. Chem. Soc.,* **124**, 14840 (2002). Structural features of the cytochrome c molten globule revealed by fluorescence energy transfer kinetics.

131. J. S. Valentine, P. J. Hart, *Proc. Natl. Acad. Sci. U. S. A.,* **100**, 3617 (2003). Misfolded CuZnSOD and amyotrophic lateral sclerosis.

132. R. W. Strange, S. V. Antonyuk, M. A. Hough, P. A. Doucette, J. S. Valentine, S. S. Hasnain, *J. Mol. Biol.,* **356**, 1152 (2006). Variable metallation of

human superoxide dismutase: Atomic resolution crystal structures of Cu-Zn, Zn-Zn and as-isolated wild-type enzymes.

133. R. W. Strange, S. Antonyuk, M. A. Hough, P. A. Doucette, J. A. Rodriguez, P. J. Hart, L. J. Hayward, J. S. Valentine, S. S. Hasnain, *J. Mol. Biol.,* **328**, 877 (2003). The structure of holo and metal-deficient wild-type human Cu, Zn superoxide dismutase and its relevance to familial amyotrophic lateral sclerosis.

134. A. K. Faradjian, R. Elber, *J. Chem. Phys.,* **120**, 10880 (2004). Computing time scales from reaction coordinates by milestoning.

135. R. Elber, *Biophys. J-Biophys. Lett.*, in press (2007). A milestoning study of the kinetics of an allosteric transition: Atomically detailed simulations of deoxy scapharca hemoglobin.

136. A. M. A. West, R. Elber, D. Shalloway, *J. Chem. Phys.*, in press (2007). Extending molecular dynamics timescales with milestoning: Example of complex kinetics in a solvated peptide.

137. D. Moroni, P. G. Bolhuis, T. S. van Erp, *J. Chem. Phys.,* **120**, 4055 (2004). Rate constants for diffusive processes by partial path sampling.

Figures and table captions

Figure 1: The planar two-water model system.

Figure 2: The model exhibits motion across a wide range of frequencies. The left view shows a short trajectory of one component of the model system showing the period of the fastest motion to be approximately 8.8 fs. The right view shows the trajectory of one dipole angle. Note that there is a low frequency motion with period approximately 10 ps.

Figure 3: Power spectrum analysis for the dipole angle of one molecule. In the right view, frequency peaks can be seen corresponding to several fast motions, including the fastest of period 1/(0.114/fs)= 8.8 fs. The left view shows an enlarged region at the low-frequency end of the scale. The slow motion due to long-range electrostatics can be observed with period 1/(0.0001/fs) = 10000 fs.

Figure 4: Sensitivity of impulse MTS method to slow force update interval. The energy error is essentially unchanged from Velocity Verlet at update interval up to 4 fs. For larger update intervals, energy error is erratic, with a notable jump at the period of the fastest motion.

Figure 5: Power spectrum analysis of Langevin dynamics simulation of the model system. Frequency peaks are seen to be in qualitative agreement with constant-energy simulations, but somewhat broadened.

Figure 6: Average bond energy versus slow force update interval for Langevin-stabilized extrapolation MTS method. These average energies, taken from long simulations, show no sensitivity to slow force update interval.

Figure 7: Simplified view of the optimization of a trajectory. The circles denote configuration snapshots taken along the initial guess for the trajectory (dotted line). The first and last circles are the fixed boundaries of the path. The solid line is the resulting trajectory after optimization of the target function Θ. The arrows indicate the direction of the gradient of the target function for that particular coordinate slice.

Figure 8: Ribbon view of cytochrome c at four different positions along one of the folding trajectories.

Figure 9: Snapshots of SOD Trajectory. The arrow indicates the direction of folding. The top structure is the initial unfolded conformation. The second structure is an intermediate structure in the trajectory with partial secondary structure formation. The bottom structure is the folded dimer. The monomer size and separation are not shown to scale to enhance clarity. The center of mass distances in the first two snapshots[116] are 280Å and 50Å greater than the equilibrium dimer distance, respectively. The disulfide bridge between Cys57 and Cys146 was conserved during the simulations.

Table 1: Example of input file (path.inp) for SDEL run.

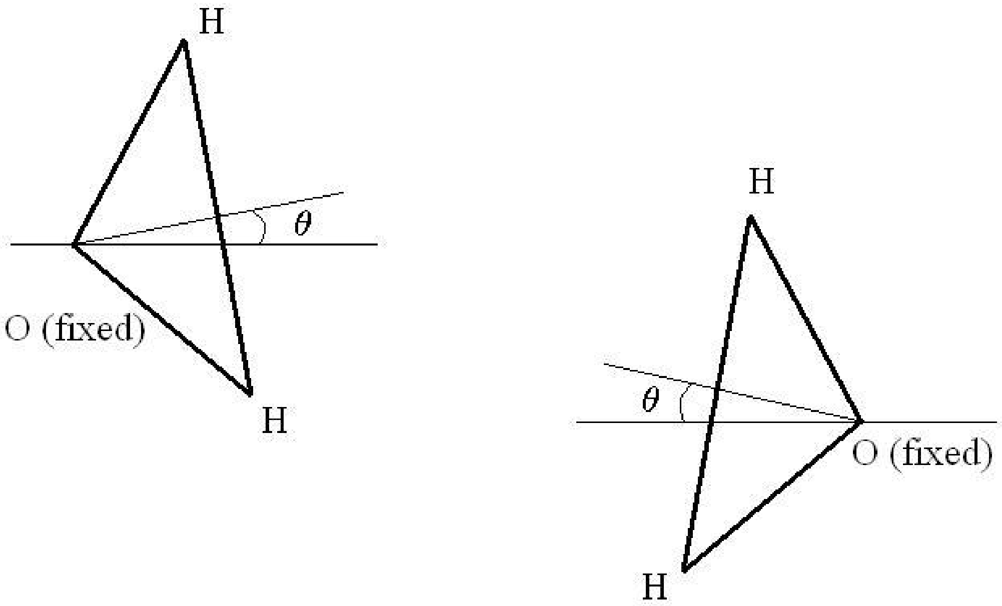

Fig. 1

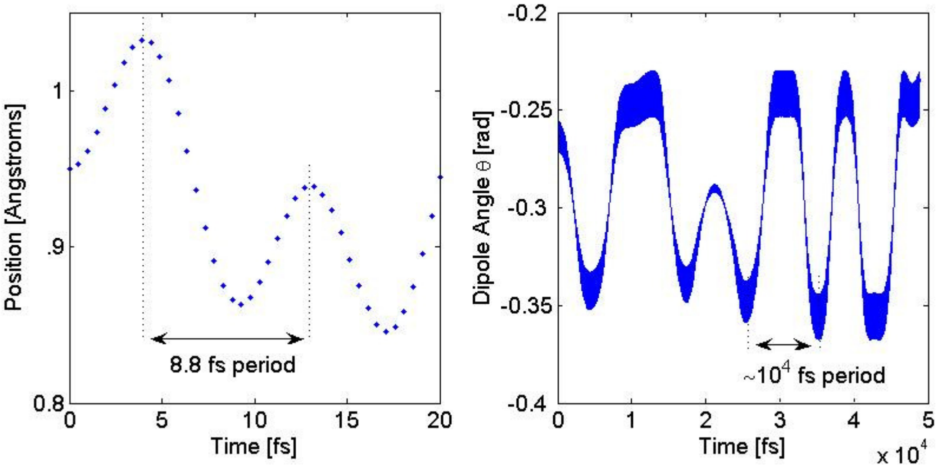
Position [Angstroms]
1
.9
0.8
0
5
10
15
20
Time [fs]
8.8 fs period
Dipole Angle θ [rad]
-0.2
-0.25
-0.3
-0.35
-0.4
0
1
2
3
4
5
Time [fs]
x 10^4
~10^4 fs period

Fig. 2

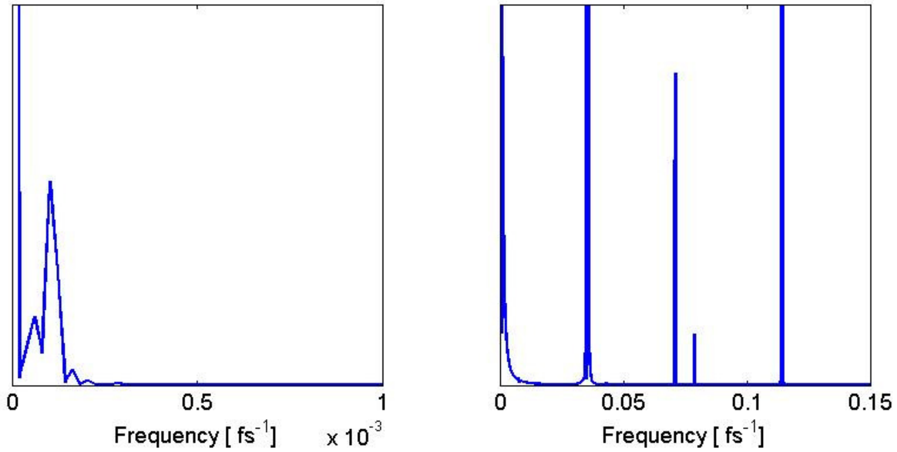

Fig. 3

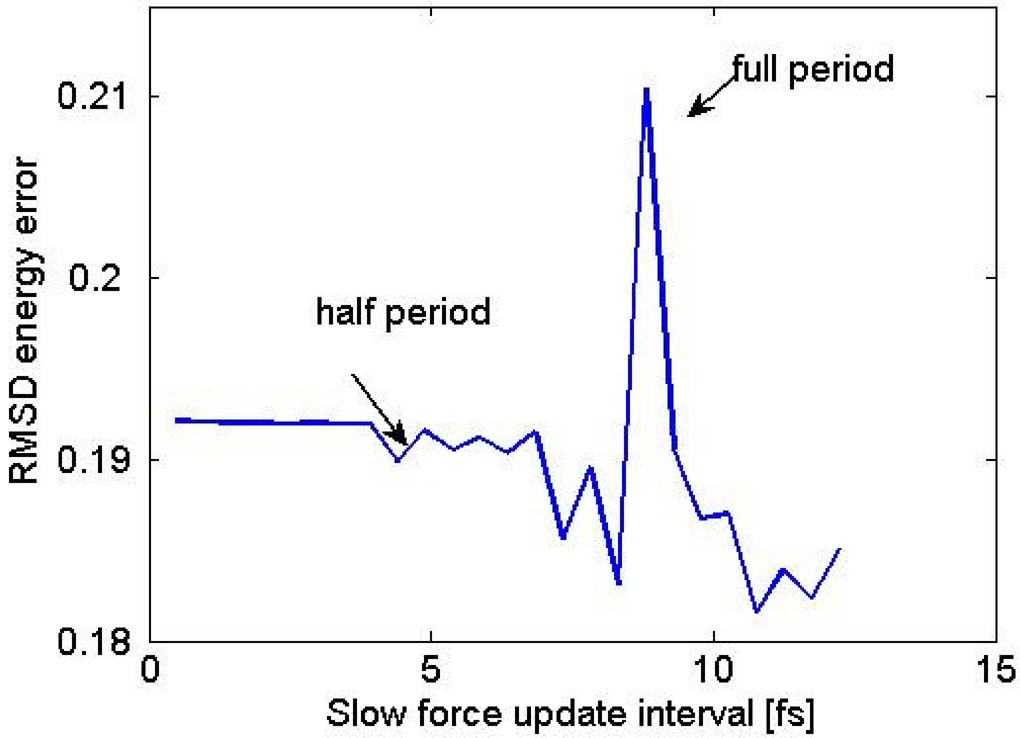
full period
half period
RMSD energy error
0.21
0.2
0.19
0.18
0
5
10
15
Slow force update interval [fs]

Fig. 4

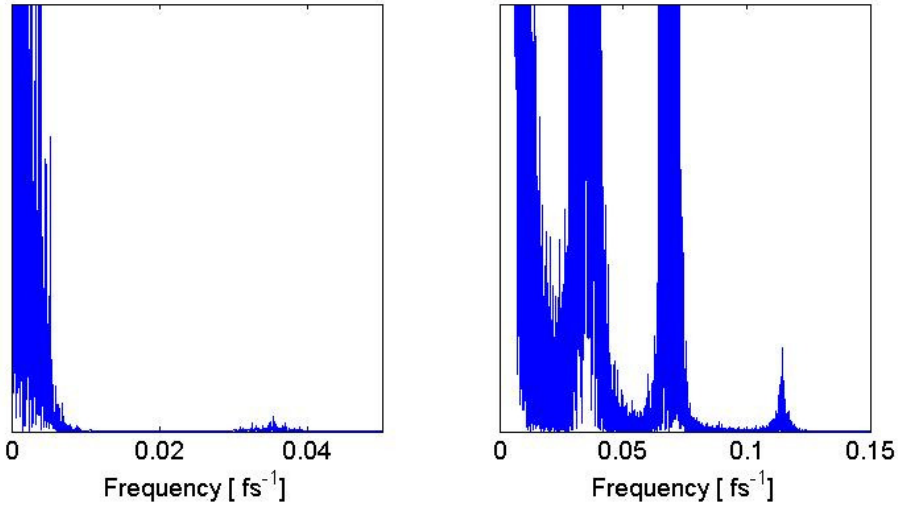

Fig. 5

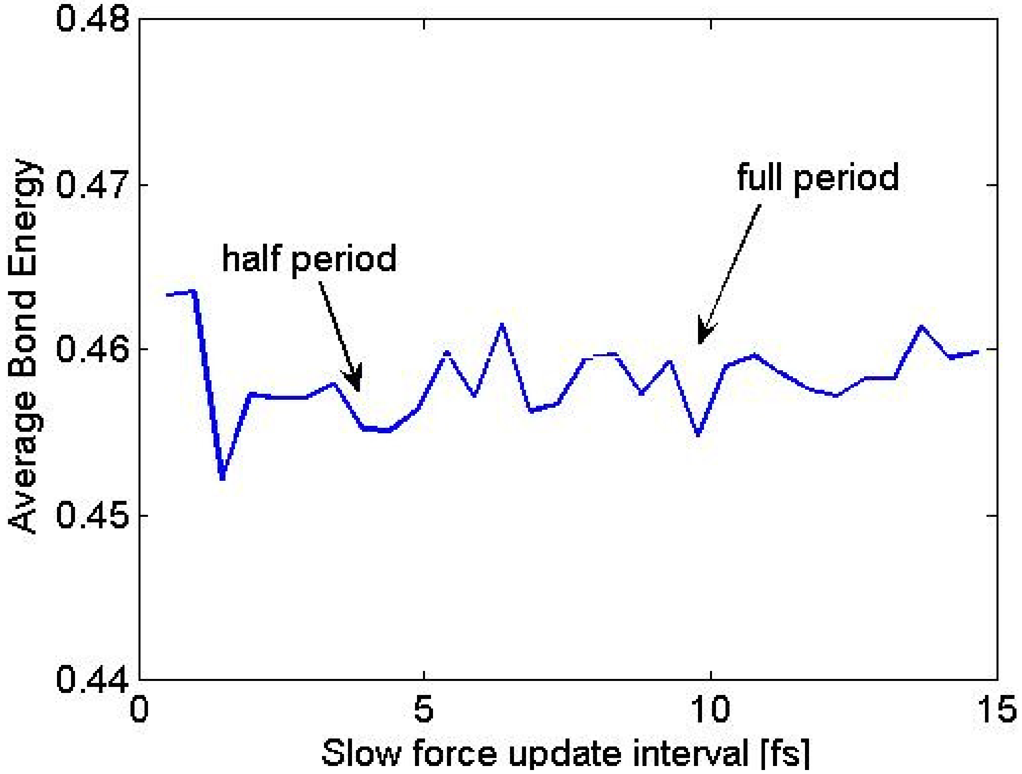
0.48
0.47
0.46
0.45
0.44
Average Bond Energy
full period
half period
0
5
10
15
Slow force update interval [fs]

Fig. 6

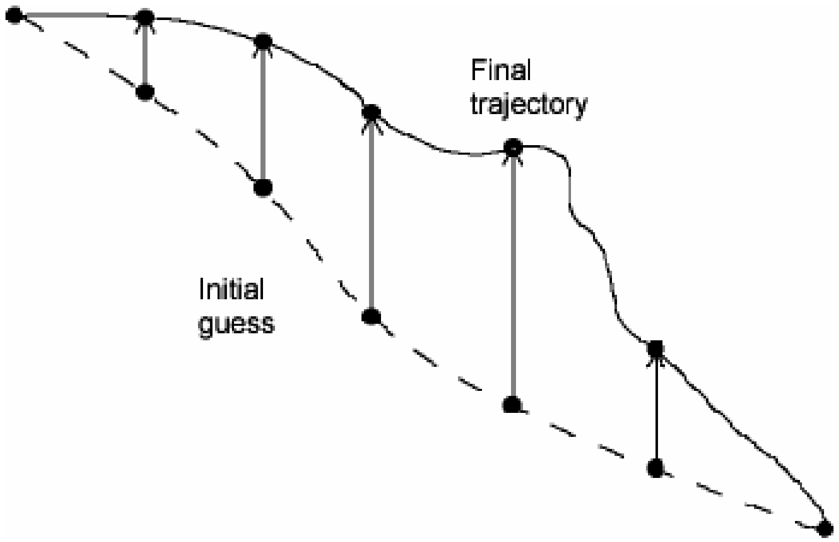

Figure 7

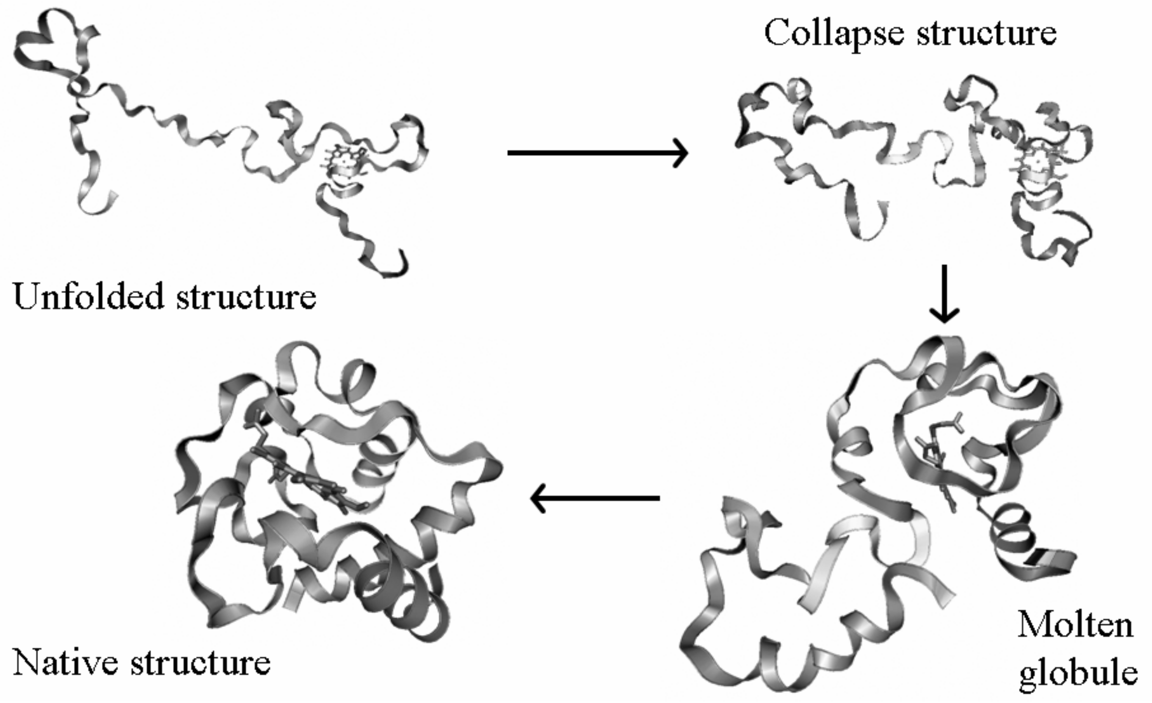
Collapse structure
Unfolded structure
Native structure
Molten globule

Figure 8

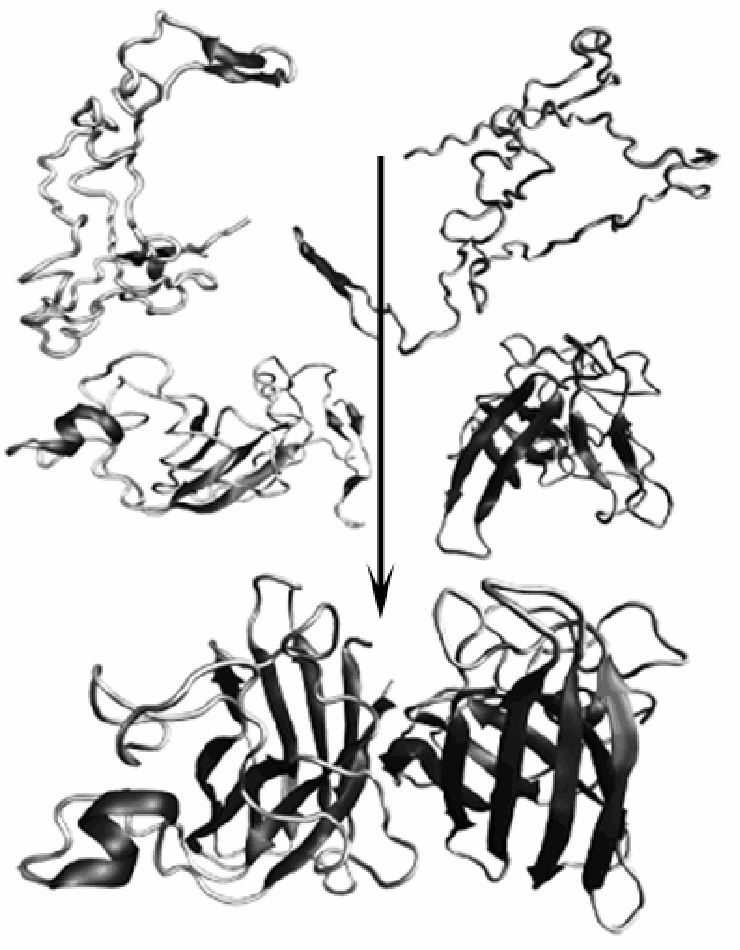

Figure 9

```
file conn name=(val.wcon) unit=10 read
file rcrd name=(valmin200.pth) bina unit=14 read
file wcrd name=(valpath.PTH) bina unit=12 wovr
#ste=5000 list=500
gama=2000.0 grid=200 pdqe=-42.2 gbsa
rmax=9999. epsi=1. v14f=8. el14=2. cpth
proc=10
tmpr=30000.0
dtop=1.0d-4
anne
action
```

Table 1