Long Time Dynamics of Complex Systems

RON ELBER,* AVIJIT GHOSH, AND
ALFREDO CÁRDENAS
*Department of Computer Science, Upson Hall 4130,
Cornell University, Ithaca, New York 14853*

Received July 11, 2001

ABSTRACT
Molecular dynamics trajectories of large biological molecules are restricted to nanoseconds. We describe a computational method, based on optimization of a functional, to extend the time of molecular simulations by orders of magnitude. Variants of our technique have already produced microsecond and millisecond trajectories. The large steps enable feasible computations of atomically detailed approximate trajectories. Numerical examples are provided: (i) a conformational change in blocked glycine peptide and (ii) helix formation of an alanine-rich peptide.

I. Introduction

Molecular dynamics (MD) simulations provide an atomically detailed description of complex systems on a wide range of temporal and spatial scales. Macroscopic phenomena, such as conduction, can be investigated with computer simulations based on microscopic interactions. Conduction requires extrapolation from microscopic modeling to large temporal and spatial scales. The focus of this Account is the extension of simulation time scales using recent methodology[1−3] to enable studies of biochemical phenomena not accessible to MD. Molecular processes in biology happen on time scales that range from femtoseconds to minutes and more.

Here, we use the term MD for a simulation technique that solves the classical equations of motion at the atomic level. We assume that an accurate (and usually empirical) potential is available that describes the interactions between the atoms. The dynamics on the empirical energy surface is described by classical mechanics (Newton's law). We exclude from the present discussion phenomenological models, with potentially smaller number of degrees of freedom, such as the Langevin equation. The discussion is limited to dynamical models and differential equations that directly follow from microscopic parameters. For example, the Langevin equation requires a friction constant. The friction constant, which significantly affects the dynamics, is not a microscopic parameter. We also do not discuss calculations that assume a clear separation of time scales such as the transition state theory or the "transition paths sampling".[4] The discussions are therefore limited to the most straightforward mode of calculations.

Despite the numerous successes of the MD approach for studying biochemical processes,[5] an important limitation stands out. The extrapolation to large temporal scales was proven problematic, and most practical applications have been limited to nanoseconds. As a result, it is not currently possible to study the atomically detailed dynamics of many interesting processes in biochemistry and biophysics. Examples are conformational transitions relevant for protein activity (e.g., the R-to-T transition in hemoglobin, tens of microseconds[6]) or ion permeation through membrane channels (microseconds for ion permeation through the gramicidin channel[7]). To understand the origin of this limit and why it was (and still is) so hard to fix, we will briefly review current algorithms for MD calculations. We then proceed with the description of a new promising approach to extend the time scales accessible to computer simulations that was developed in the authors' laboratory. Two examples are discussed at the end of the Account.

II. The Molecular Dynamics Approach

In the molecular dynamics approach we determine the time evolution of a molecular system. This time evolution is called a trajectory and is denoted by $\vec{X}(t)$, where $\vec{X}$ is the coordinate vector of all the atoms in the system and t is the time. The classical equations of motion that determine a trajectory are

$$M\frac{\mathrm{d}^2\vec{X}}{\mathrm{d}t^2} = -\frac{\mathrm{d}U}{\mathrm{d}\vec{X}} \tag{1}$$

where M is the mass matrix (diagonal for Cartesian coordinates) and U is the microscopic interaction potential. With two initial conditions, $\vec{X}(t=0)$ and $\vec{V}(t=0) \equiv \mathrm{d}\vec{X}/\mathrm{d}t|_{t=0}$, eq 1 can be solved in small time steps. A procedure to solve numerically eq 1, which is widely used in condensed phase simulations, is the Verlet algorithm,[8]

$$\begin{aligned} \vec{X}_{i+1} &= \vec{X}_i + \Delta t\cdot\vec{V}_i - M^{-1}\frac{\Delta t^2}{2}\frac{\mathrm{d}U}{\mathrm{d}\vec{X}}\bigg|_{\vec{X}=\vec{X}_i} \\ \vec{V}_{i+1} &= \vec{V}_i - M^{-1}\frac{\Delta t}{2}\left[\frac{\mathrm{d}U}{\mathrm{d}\vec{X}}\bigg|_{\vec{X}=\vec{X}_i} + \frac{\mathrm{d}U}{\mathrm{d}\vec{X}}\bigg|_{\vec{X}=\vec{X}_{i+1}}\right] \end{aligned} \tag{2}$$

The index i is for the discrete time measured in steps of Δt. In a single cycle we use the coordinates and the

Ron Elber was born in Rehovot, Israel (1957), received his Ph.D. (1984) from the Hebrew University of Jerusalem in theoretical chemistry, and continued his research and teaching in computational biology thereafter. He was a professor at the Department of Chemistry, University of Illinois at Chicago, from 1987 to 1991 and at the Chemistry and Biology Institutes at the Hebrew University from 1992 to 1998. From 1999 to present, he is a Professor of Computer Science at Cornell University.

Avijit Ghosh was born in Birmingham, England, in 1970. He received his B.S. (1994) working with Professor B. M. Pettitt from the University of Houston and his Ph.D. (2000) from Columbia University, working with Professor R. Friesner. He moved to Cornell University as a NSF postdoctoral fellow in bioinformatics with Harold Scheraga and Ron Elber.

Alfredo Cárdenas was born in Caracas, Venezuela, in 1963. He received his B.S. degree in chemistry (1988) from Simón Bolívar University in Caracas and his Ph.D. degree in chemistry (2000) from the University of Pittsburgh, working with Professor Rob Coalson. He was also an Andrew Mellon predoctoral fellow of the Faculty of Arts and Sciences of the University of Pittsburgh (1998−2000). He joined Elber's group at Cornell University as a postdoctoral research associate in 2000.

* To whom correspondence should be addressed. Phone: (607) 255-7416. Fax: (607) 255-4428. E-mail: ron@cs.cornell.edu.

10.1021/ar010021d CCC: $22.00
Published on Web 02/26/2002

velocities at time i to generate the coordinates and the velocities at time $i + 1$. This process is repeated until the number of steps times the step size is equal to the time of interest.

The equation for the coordinate vector resembles a second-order Taylor expansion in the time step. To obtain an accurate solution, the time step must be small. The typical time step that is used in numerical solutions of eq 2 is a femtosecond (10^{-15} s). To reach a microsecond and watch ion migration through a channel, 10^9 femtosecond steps are required. Moreover, milliseconds (and 10^{12} femtosecond steps) are required to simulate gate opening in channels. The large number of steps necessitates tremendous resources and cannot be done in a routine way. The sequential solution of eq 2 is therefore limited in most applications to the nanosecond time scales mentioned earlier. Heroic efforts, using months of CPU time on a supercomputer, led to a microsecond trajectory.[9] However, these calculations are far from routine.

Is it possible to reduce the number of steps by increasing the step size? Most of the computational efforts are spent on computing the potential derivatives (the forces). A reduction in the number of steps will reduce the number of force evaluations and speed up the calculations. By increasing the size of the time step, we expect to trade accuracy for speed. Numerous studies in computational biochemistry are qualitative explorations along the reaction pathway and may not require the high accuracy of a femtosecond step. Unfortunately, as was shown by a series of studies,[10] an increase of the time step in the framework of eqs 1 and 2 is limited. While much has been learned, and numerous algorithms to speed up the calculation of the potential derivatives have been designed,[11] the actual increase in the time step was small. The increase of less than an order of magnitude is still too small to make possible the study of the processes mentioned before.

The reason for the apparent bound on the step size is stability. Some of the coordinates change rapidly in time (bond vibrations, atomic collisions), and it is necessary to use small steps to follow the rapid changes. If the basic time step is increased beyond a few femtoseconds, then the solution accumulates exponential errors. This means that the coordinates "blow up", and after a few iterations they do not resemble a known biomolecule. An example of a numerical "blow up" is shown in Figure 1 for the harmonic oscillator. When the time step is larger than a fifth of the period, the solution loses stability.

Since the rapid motions are the cause, efforts have been made to remove some of them. For example, the SHAKE algorithm freezes internal coordinates (bonds, angles, etc.) at their ideal values. The freeze removes fast motions from the integration scheme and generates approximate trajectories.[12] The removal of high-frequency motions is appealing since fast motions are expected to be less relevant at long times compared to slow degrees of freedom of the system. The filtering enables an increase of the time step by a factor of about 2. This increase is significant but still insufficient to cover the many orders of magnitude in time that are not accessible to MD.

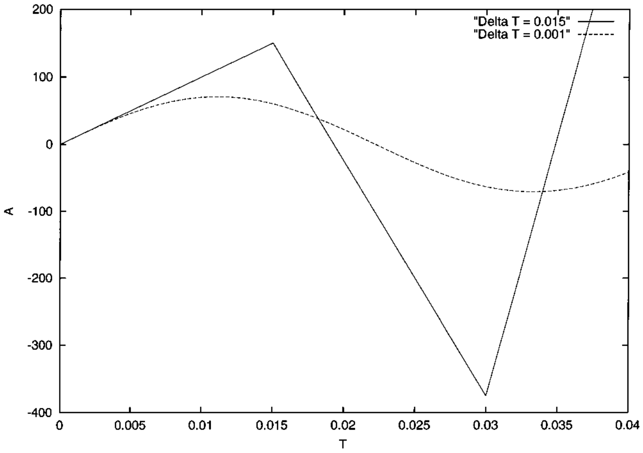

FIGURE 1. Demonstrating the numerical instabilities of the Verlet algorithm for the harmonic oscillator. The amplitude is plotted as a function of time. The oscillator period is 0.063. When the time step is set to about one-fourth of the period, the coordinates increases exponentially.

In SHAKE the fast degrees of freedom are identified first, and the contributions of these degrees of freedom are removed from the equations of motion. It is not simple to identify all the rapid motions, and the difficulties in the identification bound the size of the step even in algorithms that use SHAKE. Besides bond and angle vibrations, other fast motions are induced by collisions and are not covered by the SHAKE algorithm. Atoms that are close to each other define a "collision". They experience significant repulsive potential that forces them to depart rapidly. The identity of colliding atoms is constantly changing, making it difficult to choose the proper coordinates for tailored analysis. Special treatment of collisions has been attempted in the past but is inefficient to execute for more than one coordinate at a time.[13] In the context of the problems mentioned above, the backward Euler (BE) scheme[14] is an intriguing approach. The BE makes it possible to reduce the amplitudes of high-frequency motions with no need to identify them first. However, to recover energy that is lost in BE through the rapid modes, a Langevin equation with a phenomenological friction coefficient is used.

III. Functional Approach to Classical Mechanics

Equation 1 is not the only way to think about classical dynamics. An alternative is the use of functionals and actions. The classical action S_{cl} is[15]

$$S_{\mathrm{cl}} = \int_0^{\tau} L \, \mathrm{d}t \qquad L = \frac{1}{2}\dot{\vec{X}}^t \cdot M \cdot \dot{\vec{X}} - U(\vec{X}) \tag{3}$$

For convenience we define mass-weighted coordinates $\vec{Y} = M^{1/2}\vec{X}$, and rewrite the Lagrangian L as $L = (1/2)\dot{\vec{Y}}^t \cdot \dot{\vec{Y}} - U(\vec{Y})$. This substitution eliminates the need to carry around the mass. A trajectory, $\vec{Y}(t)$ ($t \in [0,\tau]$), is determined from eq 3 using the condition that S_{cl} is stationary with respect to variation in $\vec{Y}(t)$. The end points of the trajectory are fixed.[15] $\vec{Y}(t)$ that makes S_{cl} stationary solves also the classical equations of motion (eq 1). Equation 3 is not the only functional that describes classical dynamics. An alternative is (based on integration over the path length $d\vec{l} \equiv \dot{\vec{Y}}dt$[16])

$$S_l = \int_{\vec{Y}_1}^{\vec{Y}_2} \sqrt{2(E-U)}\, \mathrm{d}l \tag{4}$$

Instead of fixing the total time as in eq 3, the total energy of the system (E) is fixed. As before, the end points $\vec{Y}_1$ and $\vec{Y}_2$ are constrained. Fixing the energy instead of time is attractive since the energy can be estimated using equilibrium considerations. The total time is hard to estimate without kinetic data. Note that eq 4 also has an "equation of motion" associated with it, similar to the relationship between eqs 1 and 3.[16]

It is possible to design a numerical protocol to compute trajectories with prespecified boundary conditions ($\vec{Y}(0)$ and $\vec{Y}(t)$) that is based on the above definition of the actions. For example, a discrete version of the action in eq 3 is

$$S_{\mathrm{cl}} \cong \sum_{i=1,\ldots,N} \Delta t \left[\frac{(\vec{Y}_i - \vec{Y}_{i-1})^t\,(\vec{Y}_i - \vec{Y}_{i-1})}{2\Delta t^2} - U(\vec{Y}_i) \right] \tag{5}$$

where $\vec{Y}_0$ and $\vec{Y}_N$ are the fixed end points and the time of the trajectory is $N\Delta t$. By $\vec{Y}_i$ we denote the numerical approximation to $\vec{Y}(t_i)$. The finite difference expression provides an approximation for the kinetic energy. The plan is to optimize S_{cl} as a function of all the intermediate coordinates $\{\vec{Y}_j\}_{j=1}^{N}$ (time slices). The optimal path is the desired trajectory. Note, however, that the problem at hand is significantly more complex than the numerical solution outlined in eq 2. In eq 5 the whole trajectory is considered. The trajectory (a set of N time slices) is optimized to give a stationary S_{cl}. The dimensionality of the optimization problem is proportional to the number of steps. If the time step is a femtosecond and the total length of a trajectory is a nanosecond, then a million time slices are required *simultaneously* to perform the action optimization. This is a significant burden on any computer, as modern as it may be. The effort of computing an action with N time slices should be contrasted with a single or two structures that are required to progress the solution in eq 2.

How about stability? Can we use the action formulation and employ significantly larger time steps than before? Unfortunately, the stability of the numerical solution of the above action is comparable to that of the initial value difference equation (eq 2) (Ron Elber, unpublished results). One of the problems is that the action can change from having a minimum to having a maximum as a function of the step size, making the optimization difficult to perform. It is therefore no wonder that the optimization of the action in a straightforward way is not a widely used numerical approach to study classical dynamics. It seems that we are stuck with the initial value approach and the small time step.

IV. The Stochastic Difference Equation

The formulations we discussed so far are deterministic. Given initial coordinates and velocities, or given initial and final coordinates, there is a unique trajectory that solves the classical equations of motion. This statement is correct from an analytical point of view but becomes less obvious when finite computer accuracy is considered. In condensed phase simulations, the trajectories tend to be chaotic. This means that small changes in the initial conditions $\vec{Y}(0)$ and $\dot{\vec{Y}}(0)$ can drastically modify the calculated path. Since practical calculations always have (at least) truncation errors, slightly different starting values may yield profoundly different trajectories. A single trajectory is therefore an ill-defined entity.

If we were provided with the error $\vec{\epsilon}(t)$ of the numerical solution, then the "true" trajectory $\vec{Y}(t_i) = \vec{Y}_i + \vec{\epsilon}(t_i)$ could have been obtained. In the above formula, $\vec{Y}(t_i)$ is the exact trajectory and $\vec{Y}_i$ is the trajectory generated numerically. Of course, in practice the errors are unknown. The algorithms discussed so far ignored the errors, assuming that the time step is sufficiently small. In the stochastic difference equation (SDE) we model the errors statistically. In the present version we use a simple model of the errors (see below). It is expected that as time progresses and more experience is gathered on the properties of the solution, better modeling of the errors will emerge.

Consider the finite difference equation that defines an error function for the "true" trajectory $\vec{Y}(t)$:

$$\vec{\epsilon}(t) = \frac{\vec{Y}(t+\Delta t) + \vec{Y}(t-\Delta t) - 2\vec{Y}(t)}{\Delta t^2} + \left.\frac{\mathrm{d}U}{\mathrm{d}\vec{Y}}\right|_{\vec{Y}=\vec{Y}(t-\Delta t)} \tag{6a}$$

$$\vec{\epsilon}(l) = \frac{\vec{Y}(l+\Delta l) + \vec{Y}(l-\Delta l) - 2\vec{Y}(l)}{\Delta l^2} + \left.\frac{\mathrm{d}U/\mathrm{d}\vec{Y} - [(\mathrm{d}U/\mathrm{d}\vec{Y})\cdot\hat{e}_l]\hat{e}_l}{2[E - U(\vec{Y})]}\right|_{\vec{Y}=\vec{Y}(l-\Delta l)} \tag{6b}$$

Equation 6a is an approximation to eq 1. Equation 6b is not so well used, but similarly to eq 6a it can be derived by functional variation of an action (eq 4[16]). In eq 6b we used l to denote the path length and $\hat{e}_l$ to denote a unit vector parallel to the path direction. This vector is estimated by finite difference ($\hat{e}_l \cong [\vec{Y}(l+\Delta l) - \vec{Y}(l-\Delta l)]/(2\Delta l)$).

Since the step, Δt or Δl, is finite, the "error" as defined in eqs 6 is not zero for the *exact* trajectory. Equations 6 are used to define an approximate trajectory $\{\vec{Y}_i\}_{i=0}^{N}$ such that

$$\vec{0} = \frac{\vec{Y}_{i+1} + \vec{Y}_{i-1} - 2\vec{Y}_i}{\Delta t^2} + \left.\frac{\mathrm{d}U}{\mathrm{d}\vec{Y}}\right|_{\vec{Y}=\vec{Y}_{i-1}} \tag{7a}$$

$$\vec{0} = \frac{\vec{Y}_{i+1} + \vec{Y}_{i-1} - 2\vec{Y}_i}{\Delta l^2} + \frac{\mathrm{d}U/\mathrm{d}\vec{Y}_{i-1} - [(\mathrm{d}U/\mathrm{d}\vec{Y}_{i-1})\hat{e}_l]\hat{e}_l}{2[E - U(\vec{Y}_{i-1})]} \tag{7b}$$

In the stochastic difference equation we compute the vectors $\{\vec{Y}_i\}_{i=1}^{N}$ and model the errors statistically in an

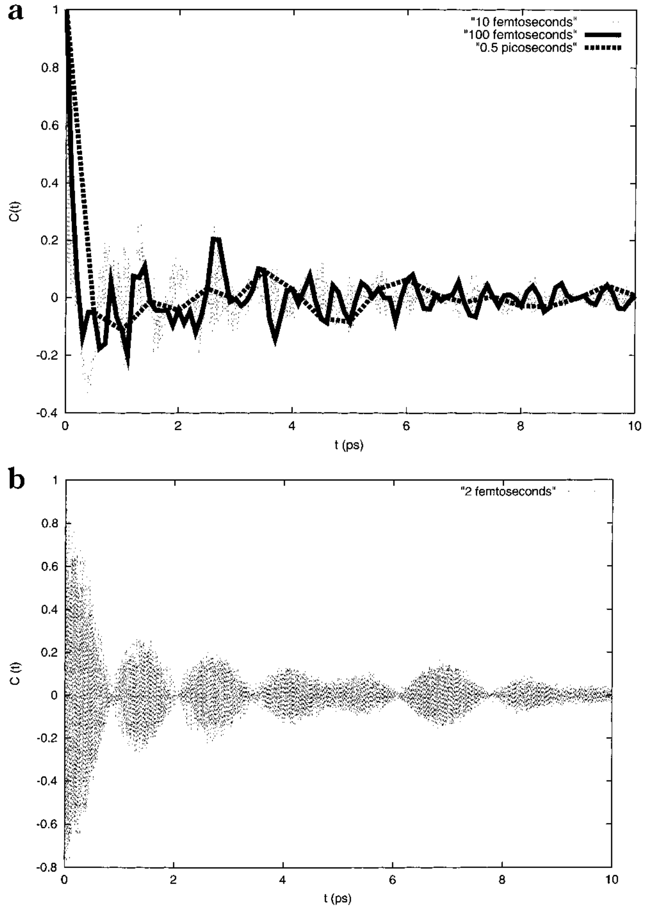

FIGURE 2. Error correlation function computed as an ensemble average over an exact trajectory: $C(t) \equiv \langle \vec{\epsilon}(t)\cdot\vec{\epsilon}(0)\rangle$. The error is a function of the time step. For sufficiently large time step ($\Delta t > 10$ fs), rapid decay of the correlation is observed (a). More significant correlation is observed for a 2 fs time step, but even then the correlation decays substantially after 10 ps (b).

attempt to estimate the deviation from $\vec{Y}(t)$. We sample an ensemble of plausible trajectories using eq 7 with "allowed" errors that are determined by the model. Based on numerical experimentation, we have chosen the following (Gaussian) model for $\vec{\epsilon}(t)$:[1]

$$\langle \vec{\epsilon}(t)\rangle = 0 \qquad \langle \vec{\epsilon}(0)\cdot\vec{\epsilon}(t)\rangle = \sigma^2\,\delta(t) \tag{8}$$

The error function (eq 6), which was extracted for a small system (a dipeptide) and a larger system (folding of C peptide in explicit solvent), indeed supports this model (see Figures 2−5 in ref 1). A few comments:

(i) We assume that the errors are independent, that they are distributed uniformly in time, and that their distribution (at one time slice) can be described with a single width parameter. This is (of course) a simplification, and refinements may follow. For example, it is possible to make the error dependent on the type of the coordinates or on the time.

(ii) The error function depends on the step size. There is a "minimal size" of a step for which the correlations decay rapidly, as suggested by eq 8. Our statistical model will not work for step sizes that are "too small" (Figure 2) since the trajectory is produced accurately for most motions. The few modes that are not accurate are correlated.

(iii) There is a proof[2] that the filtering of high-frequency modes ($\omega > \pi/\Delta t$) is the major difference between $\{\vec{Y}_i\}_{i=1}^{N}$ and $\vec{Y}(t)$. In the approximate trajectory, $\{\vec{Y}_i\}_{i=1}^{N}$, the rapid motions are removed. In contrast to the SHAKE algorithm, the filtering is done in an automated way, and thus it is not necessary to identify the rapid motions first.

Item (iii) mimics the SHAKE algorithm mentioned earlier. However, there are also difficulties. For example, in a double-well system there are two types of rapid motions. One rapid motion corresponds to oscillations within a well. The second fast motion is the transition over a barrier from one well to the next. Filtering of all the high-frequency modes also removes infrequent but fast motions such as transitions over high-energy barriers. As a result, the SDE describes better diffusive behavior and motions over low barriers, in which rapid and rare transitions do not make significant contributions to the dynamics.

Is it possible to recover some of the rapid motions, or to model them? This is the point where the errors enter the game. Accepting the model of eq 8, we ask, "What is the probability of obtaining error $\vec{\epsilon}_i$ at the i time slice?" The error is used in the statistical sense to get the optimal (but approximate) trajectory closer to the exact result. The exact trajectory includes the high-frequency components that are modeled in the SDE as Gaussian noise,

$$\mathrm{d}P(\vec{\epsilon}_i) \propto \exp\left[-\frac{\epsilon_i^{\,2}}{2\sigma^2}\right]\mathrm{d}\vec{\epsilon}_i \tag{9}$$

Since the errors are assumed to be independent of time, we can also write

$$\mathrm{d}\bar{P}(\vec{\epsilon}_1, ..., \vec{\epsilon}_n) \cong \prod \mathrm{d}P(\vec{\epsilon}_i) \tag{10}$$

Equation 10 is the probability density for a trajectory as a function of the sampled errors. It is convenient to express the probability density as a function of the coordinates and not the errors (the relation is given in eq 6a). We also use the observation[1] that the Jacobian $\mathrm{d}\vec{\epsilon}_j/\mathrm{d}\vec{X}_i$ is a constant to get

$$\mathrm{d}P(\vec{Y}_1, ..., \vec{Y}_N) \propto \exp\left[-\frac{1}{2\sigma^2}\sum_i\left(\frac{\vec{Y}_{i+1}+\vec{Y}_{i-1}-2\vec{Y}_i}{\Delta t^2}+\left.\frac{\mathrm{d}U}{\mathrm{d}\vec{Y}}\right|_{\vec{Y}=\vec{Y}_{i-1}}\right)^2\right]\prod_k \mathrm{d}\vec{Y}_k \tag{11}$$

A trajectory that maximizes the probability density in eq 11 minimizes the "action":

$$S_{\mathrm{SDET}} = \Delta t\sum_i \epsilon_i^{\,2} = \Delta t\sum_i\left(\frac{\vec{Y}_{i+1}+\vec{Y}_{i-1}-2\vec{Y}_i}{\Delta t^2}+\left.\frac{\mathrm{d}U}{\mathrm{d}\vec{Y}}\right|_{\vec{Y}=\vec{Y}_{i-1}}\right)^2 \tag{12}$$

S_{SDET} in eq 12 is optimized with simulated annealing, and alternative trajectories are sampled according to the weight σ^2. The subscript SDET stands for "stochastic difference equation with respect to time". At first sight it is not obvious that eq 12 is easier to optimize in compared to the classical action(s). A second look, however, suggests a few differences. First, we now have a stochastic model for the errors, which makes it easier to obtain cheaper (approximate) solutions. Second, the solution of eq 12 is stable for almost an arbitrary time step, providing approximate trajectories with filtered high-frequency modes. It is therefore different from the classical action that quickly becomes unstable as the step size increases. Third (true also for the classical action but not for MD), the solution of eq 12 can be optimized efficiently in parallel even on clusters of PCs with a relatively slow network.[3] A *single* MD trajectory cannot be run efficiently on such a cluster.

A similar protocol also follows for eq 6b. It is possible to have a related statistical model and an action when integrating over the path length. We denote the alternative action by S_{SDEL}:

$$S_{\mathrm{SDEL}} = \Delta l \sum_i \bar{\epsilon}_i^{\,2} \qquad (13)$$

The subscript SDEL stands for "stochastic difference equation with respect to length", to differentiate it from the time approach. The errors $\bar{\epsilon}_i$ are defined in eq 6b.

In the past[1-3] we only worked with the action defined in eq 12. Here we consider using the "length" action. The obvious question is, "What does eq 13 add in comparison to MD or to eq 12?"

Note that in eq 12 the total time of the trajectory is fixed. In classical mechanics the "conjugate" variable to time is the energy, and in eq 13 we exchange the time by the energy. The time can be computed by integration along the *optimized* path:

$$t = \int \frac{\mathrm{d}l}{\sqrt{2(E-U)}} \cong \Delta l \sum_i \frac{1}{\sqrt{2(E-U(Y_i)}} \qquad (14)$$

It is advantageous to have the time as an output in S_{SDEL} instead of an input as in S_{SDET}. The energy, which is an equilibrium property, is easier to estimate from the starting conditions compared to the total time of the process.

The two actions differ in the way in which the points are distributed along the path. In S_{SDET} the points are equally distributed in time, while in S_{SDEL} they are equally distributed in space. Depending on the problem at hand, different distribution of points along the path might be beneficial. Consider for example (again) a transition between two wells. In the time representation, this is a rare transition that, once initiated, happens rapidly. The actual transition time may be significantly shorter than the time step Δt used in eq 12. As a result, it is difficult to probe rapid transitions within the S_{SDET} protocol. A typical trajectory between two minima separated by a significant barrier will have the system at $t - \Delta t$ located at one side and at t located on the other side. The rapid transition will not be sampled.

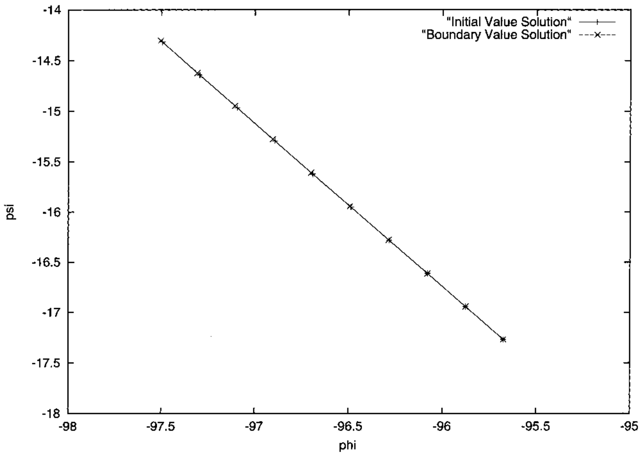

FIGURE 3. Final cycle of the path refinement procedure. The initial Δl is divided into 10 segments. The new segments are used to optimize the action, keeping the end point of the original interval fixed. Convergence is assumed when the results of the path optimization (+) agree with the solution of an initial value solver (|).

Rapid transition events are better studied with S_{SDEL}. We observe filtering of high-frequency components also in the optimization of S_{SDEL}. However, because the path is parametrized differently (with respect to length), large spatial motions are maintained, regardless of the speed at which they occur. As a result, rapid transitions (in time) with significant spatial changes (jumping between alternative minima) are well described in contrast to S_{SDET}.

As we show in the two computational examples (section VI), the spatial view of the transition that is obtained by optimization of S_{SDEL} is a sound approximation, even with only a few points. The features that are missed in the low-resolution trajectories are oscillations in quasi-stationary states that change little the spatial position of the system. However, the absolute time scale (eq 14) is an integration over the path. By filtering vibrations within wells, the path is made significantly shorter, and so is the output time. The S_{SDEL} calculation with a small number of points eliminates the waiting time in the well that in many cases makes a significant contribution to the total conversion time. Sampling a few intervals, and dividing these intervals further, can recover the correct time. The refinement stops when a solution of the differential action agrees with path optimization of the (small) refined interval (Figure 3).

V. Computational Procedure

All the calculations described in this section were performed with the program MOIL.[17] MOIL is a suite of programs for molecular dynamics simulations in the condensed phase and is in the public domain. Execution files, source code, and documentation are available from http://www.tc.cornell.edu/CBIO/moil. One of the modules in the current release of MOIL is STO, which performs S_{SDET} calculations. A related code was created to make it possible to compute classical trajectories as a function of the spatial length of the trajectory (S_{SDEL}). Specifically, the functional of eq 13 was implemented into the code, and

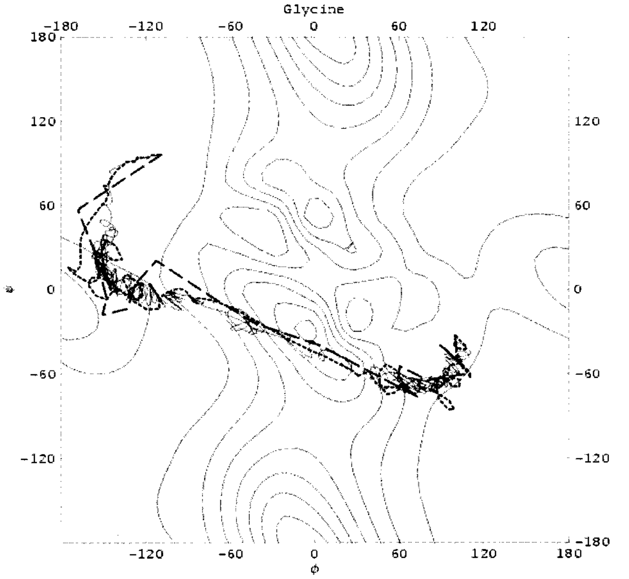

FIGURE 4. Conformation transition from C_7 axial to C_7 equatorial in a blocked glycine on a (ϕ,ψ) map. Three trajectories, computed with the optimization of S_{SDEL} with the same thermal energy, are shown. The dashed line is the low-resolution path with 20 points, the dotted line consists of 160 intermediates, and the thin continuous line has a total of 640 length slices. Note the recovery of significant oscillations within the wells once more length slices were added.

a penalty function on equal-length displacements was added ($\lambda\sum_i(\Delta l_{i,i+1} - \langle\Delta l\rangle)^2$) $\langle\Delta l\rangle = 1/(N+1)\sum_{i=0,...N}\Delta l_{i,i+1}$, λ is a constant (see also ref 18).

The force field in MOIL is a united atom model, which is a combination of AMBER[19] and OPLS.[20] Only polar hydrogens are treated explicitly. The 1−4 scaling factors for electrostatic and van der Waals interaction were 2 and 8, respectively. No distance cutoff of nonbonded interactions was used in the present study. Solvation effects were modeled within the surface-generalized Born model (SGB).[21] The SGB approximates the potential of mean force of the solvent. Effective solvation models were used for similar tasks in the past (e.g., ref 22).

In the computations described below, the initial and final configurations and the total energy were provided as input. In choosing the energy, we considered the depth of the minima that we wished to reach and added to it the average thermal energy for a system with N degrees of freedom. The total energy was therefore $E_{\mathrm{total}} = E_{\mathrm{minimum}} + NkT/2$.

Glycine Dipeptide. The initial guess for the action optimization was a straight-line interpolation between the two end points: the C_7 axial and C_7 equatorial states. All degrees of freedom were included in the calculations (including bond length and bond angles). We expect that bond displacements will be filtered out since they fluctuate rapidly and do not contribute significantly to monotonic changes in the spatial location of the path. The number of grid points varies from 20 to 160 and 640, and the results are displayed in Figure 4. The optimization of the S_{SDEL} action was done to convergence, i.e., until the gradient of the functional was less than 0.01 mass·kcal/mol·(Å)$^{-1}$. The typical number of optimization steps that were required to reach convergence was 2000. The other test of convergence is further divisions of sampled displacements Δl. In Figure 3 we show a final refinement for an interval in which the path optimization agreed with a solution of the differential equation.

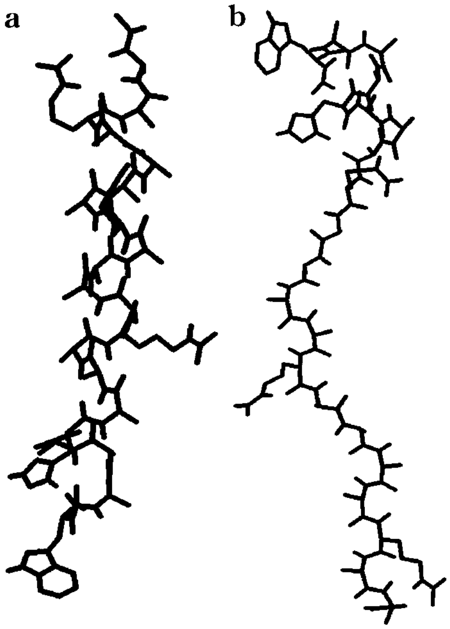

FIGURE 5. Stick models of the alanine-rich peptide: (a) the minimized helical conformation; (b) one of the starting unfolded structures. See text for more details

Helical Peptide. We considered a peptide with significant tendency to form a helix,[23] for which the folding time has been measured experimentally (it is about 220 ns). The alanine-rich peptide is AcWAAAH$^+$ (AAAR$^+$A)$_3$ANH$_2$ (Figure 5). The transition from an unfolded conformation to an α helix is considered. The unfolded conformations were generated by one nanosecond trajectory at a temperature of 600 K. Structures were picked at intervals of 200 ps and were minimized using the conjugate gradient algorithm. Convergence was assumed when the norm of the force vector was less than 0.01 kcal/mol Å. The folded state was constructed as an ideal α helix followed by the same minimization protocol. The number of grid points that was used to describe the trajectory and approximate the action integral was either 11 or 101, and the initial guess for the trajectory was a SPW minimum energy path between the two structures.[24] A refinement procedure of two intervals suggests the total time to be 4 ns. The optimization was stopped after 50 000 simulated annealing steps (about 6 h on a PC). The complete optimization run includes 100 heating and cooling cycles of 500 steps each.

VI. Results

In Figure 4 we show trajectories of glycine dipeptide, displayed on a (ϕ,ψ) map, computed at thermal energy with different levels of path resolution (starting at the low resolution of 20 grid points and refining the path to include 640 points). The blocked glycine is a highly flexible

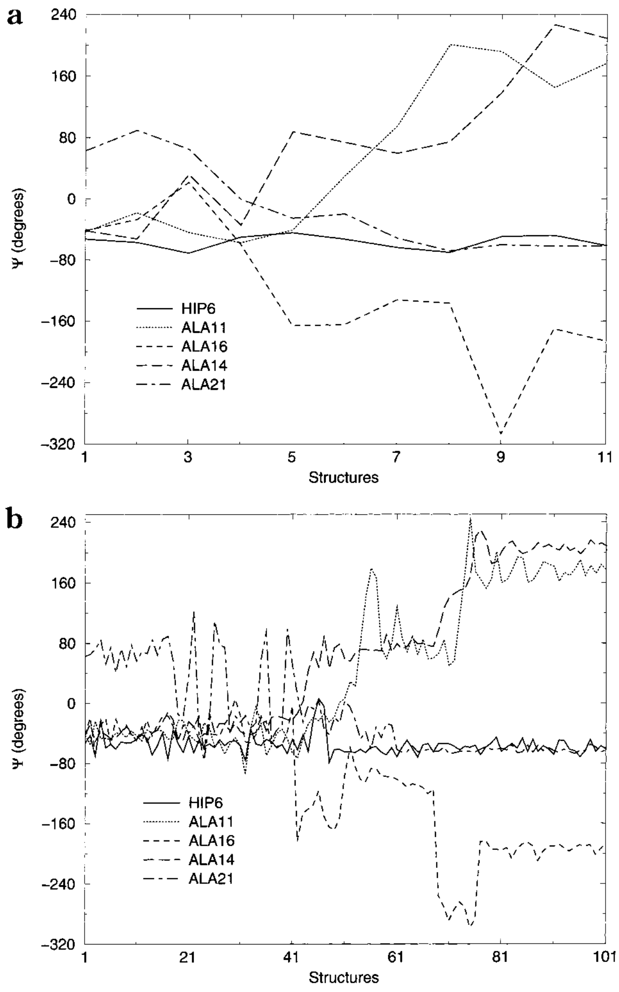

FIGURE 6. Following ψ dihedrals as a function of the trajectory spatial length. Note the sequential nature of the transition and the small (overall) differences between the (a) low-resolution path (11 points) and (b) the high-resolution path (101 points).

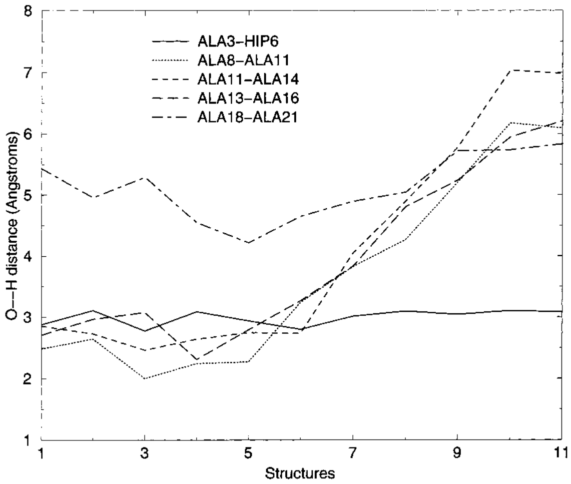

FIGURE 7. Hydrogen bond formation in low-resolution trajectories (parametrized by trajectory length, using a total of 11 points). An average over five folding trajectories is used. Note the significant similarity between the trajectories. Note also the concerted formation of hydrogen bonds, supporting a single barrier and the single-exponential behavior observed experimentally.[23]

molecule, and “rare” transitions between the C_7 axial and C_7 equatorial are not so rare. The “barrier” is low, and some oscillations are observed even at the location of the traditional barrier. A visual inspection suggests that the spatial domain visited by the trajectories remains essentially the same while the path resolution is increased. At the least, the low-resolution path (a few points, relatively large Δl) makes it possible to focus on the portions of configuration space that are relevant for the conformational transition at hand. A refinement procedure (Figure 3) suggests that at the limit of small Δl we recover correct classical dynamics. The parametrization according to the trajectory length ensures computational efficiency (we are making the most from each point in space) by emphasizing the transition periods and not the waiting time in minima. Waiting times are clearly observed when the number of length slices increases.

In Figure 5 we show a stick model for the folded and unfolded configurations of the alanine-rich peptide. In Figure 6 we show a transition path between an unfolded configuration of the alanine peptide and a helix. The transition path was computed with 11 and 101 grid points. Only the dihedral angles with most significant changes are shown (ψ dihedrals). Note that the significant oscillations that are observed for the ψ dihedral of alanine 21 are smoothed out in the lower resolution path. Clearly the length of the path will be significantly longer once the oscillations are included. Nevertheless, the position of the average path is reproduced quite well in the low-resolution trajectory. Note also that the process occurs in two bursts. Around a third of the way along the path there is a concerted transition of alanine 11, 14, and 16, and around two-thirds of the way alanines 11 and 16 undergo another transition. A refinement procedure led to a total trajectory time of 4 ns. More statistics is required to obtain the kinetic time scale.

Hydrogen bond changes averaged over five trajectories are shown in Figure 7. The simultaneous formation of hydrogen bonds suggests a single significant energy barrier, in accord with the single-exponential behavior that is observed experimentally.[23] This sequence of events is clearly seen in both paths, despite an order of magnitude difference in resolution.

VII. Concluding Remarks

In this Account we propose and demonstrate an alternative way of computing molecular trajectories. Instead of parametrizing the trajectory as a function of time, we parametrize it as a function of length. Instead of solving the Newton equations, we optimize an action, S_{SDEL}, based on a statistical model of the errors. For activated processes the numerical protocol leads to the elimination of the “incubation time” at low resolution. The waiting time at different minima is of little interest to those who focus on spatial changes of a molecular process, or in other words on “molecules in action” and not on “molecules in suspension”. The examples of a conformational transition in a blocked peptide and helix formation demonstrate the filtering effect and the conservation of structural features along the path. The concerted formation of hydrogen bonds during the process of helix formation is in accord with the single-exponential kinetics.

This research was supported by NIH and NSF grants to R.E. A.G. is supported by an NSF postdoctoral fellowship.

References

(1) Elber, R.; Meller, J.; Olender, R. Stochastic path approach to compute atomically detailed trajectories: Application to the folding of C peptide. *J. Phys. Chem.* **1999**, *103*, 899−911.

(2) Olender, R.; Elber, R. Calculation of classical trajectories with a very large time step: Formalism and numerical examples. *J. Chem. Phys.* **1996**, *105*, 9299−9315.

(3) Zaloj, V.; Elber, R. Parallel Computations of Molecular Dynamics Trajectories Using Stochastic Path Approach. *Comput. Phys. Commun.* **2000**, *128*, 118−127.

(4) Geissler, P. L.; Dellago, C.; Chandler, D.; Hutter, J.; Parrinello, M.; Autoionization in liquid water. *Science* **2001**, *291*, 2121−2124.

(5) See, for instance: Berendsen, H. J. C.; Hayward, S. Collective protein dynamics in relation to function. *Curr. Opin. Struct. Biol.* **2000**, *10*, 165−169.

(6) For a recent discussion of the R-to-T transition, see: Gibson, Q. H. Kinetics of oxygen binding to hemoglobin A. *Biochemistry* **1999**, *38*, 5191−5199.

(7) Hladky, S. B.; Haydon, D. A. Ion transfer across lipid membranes in the presence of gramicidin A: I. Studies of the unit conductance channel. *Biochim. Biophys. Acta* **1972**, *27*, 4294−312.

(8) Verlet, L. Computer experiments on classical fluids I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.* **1967**, *15*, 98−110.

(9) Duan, Y.; Kollman, P. A. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science* **1998**, *282*, 740−744.

(10) See, for instance: Schlick, T.; Beard, D. A.; Huang, J.; Strahs, D. A.; Qian, X. L. Computational Challenges in Simulating Large DNA Over Long Times. *Comput. Sci. Eng.* **2000**, *2*, 38−51. Figueirido, F.; Levy, R. M.; Zhou, R. H.; Berne, B. J. Large-scale simulation of macromolecules in solution: Combining the periodic fast multipole method with multiple time step integrators. *J. Chem. Phys.* **1997**, *106*, 9835−9849.

(11) See, for instance: Sagui, C.; Darden, T, Multigrid methods for classical molecular dynamics simulations of biomolecules. *J. Chem. Phys.* **2001**, *114*, 6578−6591.

(12) Ryckaert, J. P.; Ciccotti, G.; Berendsen, H. J. C. Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular Dynamics of n-alkanes. *J. Comput. Phys.* **1977**, *23*, 327−341.

(13) Ulitsky, A.; Elber, R. Application of the Locally Enhanced Sampling (LES) and a mean field with a binary collision correction (cLES) to the simulation of Ar diffusion and NO recombination in myoglobin. *J. Phys. Chem.* **1994**, *98*, 1034−1043.

(14) Schlick, T.; Peskin, C. S. Comment on "Backward Euler and other methods for simulating molecular fluids". *J. Chem. Phys.* **1995**, *103*, 9888−9889.

(15) Landau, L. M.; Lifshitz, E. M. *Classical Mechanics*; Butterworth-Heinenann: Oxford, 2000; Chapter 1.

(16) Landau, L. M.; Lifshitz, E. M. *Classical Mechanics*; Butterworth-Heinenann: Oxford, 2000; pp 140−143.

(17) Elber, R.; Roitberg, A.; Simmerling, C.; Goldstein, R.; Li, R.; Verkhivker, G.; Keasar, C.; Zhang, J.; Ulitsky, A. MOIL: A program for simulations of macromolecules. *Comput. Phys. Commun.* **1995**, *91*, 159−189.

(18) Elber, R.; Karplus, M. A method for determining reaction paths in large molecules: Application to myoglobin. *Chem. Phys. Lett.* **1987**, *139*, 375−380.

(19) Weiner, S. J.; Kollman, P. A.; Case, D. A.; Singh, U. C.; Ghio, C.; Alagona, G.; Profeta, S.; Weiner, P. A new for field for molecular mechanics simulation of nucleic acids and proteins. *J. Am. Chem. Soc.* **1984**, *106*, 765−784.

(20) Jorgensen, W. L.; Tirado-Rives, J. The OPLS potential function for proteins: Energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.* **1988**, *110*, 1666−1671.

(21) Ghosh, A.; Rapp, C. S.; Friesner, R. A. Generalized born model based on a surface integral formulation. *J. Phys. Chem B* **1998**, *102*, 10983−10990.

(22) Ferrara, P.; Caflisch, A. Folding simulations of a three-stranded antiparallel β-sheet peptide. *Proc. Natl. Acad. Sci. U.S.A.* **2000**, *97*, 10780−10785.

(23) Thompson, P. A.; Munoz, V.; Jas, G. S.; Henry, E. R.; Eaton, W. A.; Hofrichter, J. The Helix-Coil Kinetics of a Heteropeptide. *J. Phys. Chem.* **2000**, *104*, 378−389.

(24) Czerminski, R.; Elber, R. Self avoiding walk between two fixed points as a tool to calculate reaction paths in large molecular systems. *Int. J. Quantum Chem.* **1990**, *24*, 167−186.

AR010021D