The Journal of Physical Chemistry B

© Copyright 1999 by the American Chemical Society VOLUME 103, NUMBER 6, FEBRUARY 11, 1999

FEATURE ARTICLE

Stochastic Path Approach to Compute Atomically Detailed Trajectories: Application to the Folding of C Peptide

Ron Elber,* Jaroslaw Meller, and Roberto Olender

Department of Physical Chemistry, The Hebrew University, Givat Ram, Jerusalem 91904, Israel

Received: September 23, 1998

A novel method to compute long-time molecular dynamics trajectories is employed to study the folding kinetics of C peptide. The computational method makes it possible to use a time step larger by orders of magnitude compared to widely used molecular dynamics integrators. Rather than solving the trajectory in small time steps, the whole trajectory is optimized. The algorithm filters high-frequency modes that are modeled as Gaussian noise. The assumption of “Gaussian noise” is tested numerically in two cases and found to be adequate. In all, 31 trajectories of C peptide that folds into a helix in explicit solvent (TIP3P water molecules) are computed. The time step is 500 ps. The folding pathways and the early formation of structure are discussed. Comparisons to a 2-ns trajectory calculated with the usual molecular dynamics approach and to available experimental data are made.

I. Introduction

Molecular dynamics simulations contributed significantly to our understanding of condensed phases.[1] From the perspective of molecular biophysics, the simulations shed light on the kinetics and on the thermodynamics of numerous biophysical and biochemical processes. Examples are ligand diffusion,[2] ligand binding,[3] and folding of short peptides.[4] Nevertheless, a significant drawback of the current computational methodology is of time scales. The longest individual trajectories that are computed today are up to thousands of nanoseconds. This is far shorter than many biophysical processes such as ion transport through membranes (microseconds)[5] and folding (milliseconds to seconds).[6]

Here, we discuss an alternative method to compute molecular trajectories that makes it possible to study dynamics on extended time scales.[7] It produces approximate (but numerically stable) solutions to the differential equations of motion. The time step that we use in the alternative methodology can be of an (almost) arbitrary size and still provides a stable solution. High frequencies, $\omega \approx \pi/\Delta t$, are automatically filtered out as the time step, Δt, increases. The main features of the algorithm and a few simple examples were discussed elsewhere.[7] Here, we present the algorithm from a new theoretical viewpoint with emphasis on the noise in the difference equation. This eliminates some of the conceptual difficulties associated with stochastic differential equations.[8−10] Another theoretical question relevant to the computations of experimental measurements is what are the weights of the different trajectories. A statistical approach to compute the weights is outlined.

As an application of a significant biophysical interest, we study the folding kinetic of a short peptide. The C peptide is one of the systems that was studied most extensively in the past. It was investigated in the context of folding nucleation sites (for a recent review, see Chakrabar and Baldwin[26]). The C peptide is a fragment of the protein ribonuclease A, a fragment that spontaneously forms a partial helix. The results of individual trajectories are examined to explore the mechanism of helix formation. The sampling we have so far is insufficient to compute ensemble averages and rates that are left for future studies. Further investigations are desirable to obtain statistically

10.1021/jp983774z CCC: $18.00 © 1999 American Chemical Society
Published on Web 01/23/1999

converged data. Nevertheless, our present results elucidate several intriguing features of the folding mechanisms.

II. Method

In the widely used molecular dynamics approach, an initial value problem is solved. The coordinates and the velocities (and hence the energy) are specified at the beginning of the calculations and are propagated in time using small steps. In contrast, the technique outlined below requires the specification of the total time and the starting and ending coordinates. The value of the energy is not required. The new technique is useful to study the dynamics when the reactants and the products are known. The total time of the trajectory, which is input, determines the energy of the system.

Most of our studies are of room temperature processes. Therefore, the individual trajectories should be weighted according to their energy with the Boltzmann weight, $\exp[-E/k_BT)]$. E is the energy of the trajectory, k_B is the Boltzmann constant, and T is the temperature. If the Boltzmann weight is small (the energy is high), the resulting path will add little to the ensemble of room-temperature trajectories leading from reactants to products. The decrease of the trajectory weight with energy suggests a practical procedure to select reasonable times for the trajectory. We start with short times. If the total time is too short, the resulting trajectory will be close to a straight line that passes over high-energy domains (ballistic paths). The Boltzmann weight will be small. In brief, we desire solutions in which further lengthening of the trajectory time does not change appreciably the energy of the barrier.

We are usually interested in transitions between quasiequilibrium states and not coordinates. Therefore, some averages over initial and final conditions are required. The additional averaging leads to further variations in the energies of the trajectories and in the Boltzmann weights. Derivation of the weights of the trajectories will be discussed in the Theory section.

The proposed approach[7] complements existing numerical algorithms that are based on initial values and solve the differential equations of motion. It is potentially useful for cases in which the "reactants" (starting configuration) and "products" (ending configuration) are known but is not suitable for problems in which the end structures are not available. For example, investigating the dynamics of the R to T conformational transition in hemoglobin is accessible to the new approach (experimental crystallographic coordinates are available for the R and the T states of the protein). On the other hand, searching for the correct fold of a protein with an unknown structure (which can be done by leaving the end of the trajectory free to move) is not advantageous compared to an initial value formulation.

The Method section is divided into two parts. The first part discusses the theory of a single trajectory and the protocol to sample ensembles of trajectories. The second section deals with the numerical algorithms and estimates of the computational costs for plausible applications in general and for the system investigated in the present paper, the folding of C peptide, in particular.

II.1. Theory. In this section, the theory behind the algorithm is discussed and derived. We start with the theory of a single trajectory.

II.1.1. Definition of Errors and the Stochastic Difference Equation. The goal is to find a method that computes classical dynamics of N particles with initial positions given by a coordinate vector X. The particles interact via a nonseparable potential, $U(X)$. There are numerous ways of formulating classical mechanics that suggest useful numerical algorithms.[11] For example, one may seek stationary points of the classical action or solve the Hamilton−Jacobi equation. By far, the most common approach in condensed-phase simulations is to solve a second-order differential equation that is Newton's second law:

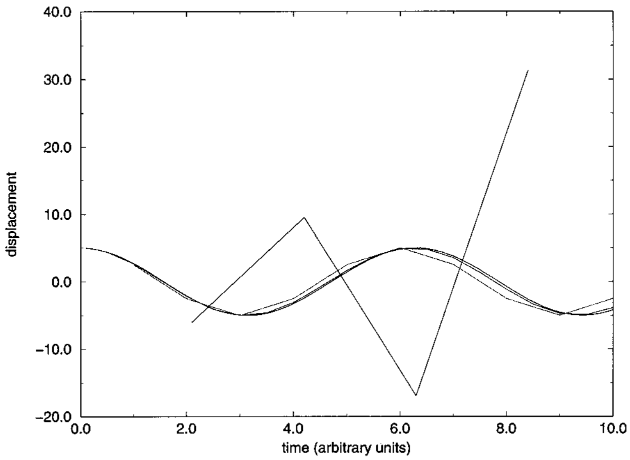

Figure 1. Numerical solutions of the equations of motions for the harmonic oscillator ($\ddot{X} + X = 0$, the period is 2π). The Verlet integrator provides accurate solutions for time steps of 0.1 and 0.5. Significant deviations from the exact solution are observed for a step of 1 (dashed line), but the solution is still stable. When the time step is 2.1, the Verlet integrator looses its stability (solid line).

$$\mathbf{M}\frac{d^2X}{dt^2} = -\frac{dU}{dX} \tag{1}$$

The mass matrix $\mathbf{M}$ is diagonal (for simplicity we restrict the discussion below to Cartesian coordinates only) and the derivative with respect to the vector X denotes a gradient. Equation 1 is the corner stone of the usual molecular dynamics approach and is solved with initial values.

A common algorithm to integrate eq 1 is the Verlet algorithm:[12]

$$\mathbf{X}_{i+1} = 2X_i - X_{i-1} - \mathbf{M}^{-1}\frac{dU}{dX_i}\Delta t^2 \tag{2}$$

The time step Δt must be small since the algorithm looses stability quite rapidly as the time step increases (Figure 1).

To start, the algorithm requires the coordinates and the velocities at some initial time, say $t = 0$;

$$X|_{t=0} = X(0);\ \frac{dX}{dt}\bigg|_{t=0} = V(0)$$

As an alternative to velocities, one may use coordinate set at somewhat later time, $X(\Delta t)$, as is done in eq 2. In principle, it is possible to write many alternative algorithms to solve the equations of motions. An algorithm of some interest to us is

$$X_{i+1} = 2X_i - X_{i-1} - \frac{\Delta t^2}{\mathbf{M}}\frac{dU}{dX_{i-1}} \tag{3}$$

Equation 3, in which the force is evaluated at the edge of the time interval, is not the natural choice to solve initial value problems. It is not symmetric in time, is less accurate than eq 2, and requires the same computational effort. However, it is not all negative. If we start at X_{i-1}, eq 3 does not require the calculations of future forces, an attractive feature of the above algorithm. More importantly, in the limit of small Δt, eq 3 is reduced to the exact formula. As discussed below, shifting the computations off the time center is convenient for the present derivation.

Consider a set of coordinates $\{X_0, X_1, ...\}$ separated in time by the step Δt. We are told that the exact solution of the differential equation $X(t)$ coincides with the above discrete set as follows: $\{X(t_0) = X_0, X(t_1) = X_1, \ldots; t_{i+1} - t_i = \Delta t\}$. That is, we have a representative discrete set of the exact trajectory. How can we verify it? The discrete set does not allow for exact evaluation of time derivatives. Typically, we obtain the discrete set from a numerical solution using integrators like Verlet (eq 2). Hence, all we have is the discrete set.

A suggestion for a test is to recalculate the value of the differential equation (the starting point for the computation). Ideally, it should be 0 (eq 1).[13] Unfortunately, the exact local value of the differential equation is not known since we only have a discrete set. There is no exact procedure to obtain the second derivatives with respect to time with a finite and discrete set of trajectory points. Nevertheless, the value of the time derivative can be approximated by a finite difference formula. Below, we use a simple expression, which is convenient computationally. As discussed later, the simple expression has unique properties, and it is not clear if more accurate differentiation improves the overall accuracy or stability.

$$\left.\frac{d^2X}{dt^2}\right|_{t=t_0} \approx \frac{X_1 + X_{-1} - 2X_0}{\Delta t^2} \qquad (4)$$

This estimate coincides with the exact expression only in the limit of an infinitesimally small time step, Δt. Hence, the estimate of the differential equation below (for finite Δt) is not necessarily 0 if the trajectory is exact.

$$\epsilon_i = \left[\mathbf{M}\frac{(X_{i-2} + X_i - 2X_{i-1})}{\Delta t^2} + \frac{dU}{dX_{i-2}}\right] \qquad (5)$$

Equation 5 is not that great as a test of accuracy, since it is an approximation. Nevertheless, eq 5 is a useful starting point for the derivations in the present manuscript. It suggests a different way to compute trajectories by minimizing the absolute value of the vector of "errors", ϵ_i. Since eq 5 is approximate, ϵ_i is not a true estimate of the errors of the solution to the differential equation. Nevertheless, for convenience, we omit the quotes in further discussions of the errors. The vector is of the same rank as the coordinates, and it provides a useful measure of the errors in the whole coordinate space.

Note also the asymmetric position of the coordinate difference in time and the force. The force $- dU/dX_{i-2}$ is computed at the edge of the interval, while computing it at the middle is in general more accurate. Even the time derivatives are computed off the center in eq 5. The off-centering is similar to eq 3 with comparable ups and downs. The off-centering has the considerable advantage of a convenient Jacobian as discussed below.

In particular, we are interested in the transformation from the error vector, ϵ_i, to the coordinate vector, X_i. For example, consider the probability to obtain an error vector between ϵ_i and $\epsilon_i + d\epsilon_i$, $P(\epsilon_i)\, d\epsilon_i$. If this distribution is known, we can ask the more "interesting" question from the molecular dynamics perspective, what is the probability of finding a coordinate vector between X_i and $X_i + dX_i$, $P(X_i)\, dX_i$. Equation 5 provides the connection; $P(X_i)\, dX_i = P[\epsilon_i(X_i)](d\epsilon_i/dX_i)\, dX_i$. The Jacobian term $(d\epsilon_i/dX_i)$ depends on the location of the force in the interval!

We have chosen the errors as in eq 5 since the resulting Jacobian is a constant ($d\epsilon_i/dX_i = \mathbf{M}/\Delta t^2$) and it does not depend on the coordinates. In fact, it can be show that even the Jacobian for the whole trajectory, from a sequence of ϵ_i to a sequence of coordinates, X_i, is a constant. The procedure to prove that the Jacobian is a constant is not trivial and is based on the expansion of the Jacobian determinant. The reader interested in a brief explanation is encouraged to use ref 9. Reference 10 includes a complete description.

$$\prod_j d\epsilon_j = \prod_j \left[\sum_i \frac{d\epsilon_j}{dX_i}\, dX_i\right] = \prod_i J_i\, dX_i = \text{const} \times \prod_i dX_i \qquad (6)$$

As we demonstrate later, simulations that are based on atomic models behave in accord with our ad hoc placement of the force at the edge of the interval. Related studies of the transformation between the errors and the coordinates were performed in the past in the context of stochastic differential equations.[8] An example is the Langevin equation, in which noise is present in the differential form.

$$\mathbf{M}\frac{d^2X}{dt^2} + \gamma\frac{dX}{dt} + \frac{dU}{dX} = R \qquad (7)$$

Our focus here is the analogy between R and ϵ. However, a few remarks on the Langevin formalism are warranted. There are two phenomenological terms that are added to the Newton's equation to give the Langevin model: (1) the friction $\gamma(dX/dt)$ and (2) the random force R. They are introduced in order to describe an external environment that is not modeled explicitly. There is no simple and exact connection between the above equation and Newtonian mechanics. Specifically, a constant friction cannot be derived from a Newtonian model. In general, the friction should be time and coordinate dependent. In practice, almost all Langevin simulations employed a constant value. Moreover, the choice of the friction strongly influences the dynamics of the system, and a trajectory so obtained is not a systematic approximation to a Newtonian atomic trajectory.

Finally, we note that in the long-time limit the mass term is sometimes ignored yielding another stochastic differential equation that describes Brownian dynamics.

$$\gamma\frac{dX}{dt} + \frac{dU}{dX} = R \qquad (8)$$

We return to explore the analogy between R, the random force, and ϵ, the errors in the numerical solution of the Newton equation. A similar argument to the ϵ case is used to find a connection between the probability distribution of R and X. The connection can be made in both the Langevin and the Brownian cases.[8] The treatment of stochastic differential equations is however trickier, since R is a stochastic variable with zero correlation in time (i.e., $\langle R(t)\, R(t')\rangle = C\delta(t - t')$, where C is a constant and δ is the Dirac's delta function). Because of the unpredictable future of R, even for a very small time step, it is not trivial to integrate the above equation. The result is an ambiguity in the solution, and different approaches to stochastic calculus (Ito and Stratonovich[8]) for the same differential equation.

We define a stochastic difference equation (SDE), which is essentially the same as eq 5.

$$\epsilon_i = \left[\mathbf{M}\frac{(X_{i-2} + X_i - 2X_{i-1})}{\Delta t^2} + \frac{dU}{dX_{i-2}}\right] \qquad (5)$$

However, we add the anstaz that the errors, ϵ_i, behave like uncorrelated (in time) Gaussian random numbers, making the above equation truly stochastic. This assumption is examined in the next section, using numerical examples. It is shown to be adequate for the two cases we studied.

Our choice of the errors, if the analogy between the difference equation and the differential equations is pushed further, is similar to the Ito calculus. If the force is computed at X_i (and therefore the Jacobian includes a derivative of the force), the corresponding calculus is of Stratonovich.[10] It is beyond the scope of the present paper to discuss stochastic calculus, and for more details we refer the interested reader to the literature.[8,10]

Our goal is to use discrete stochastic paths to model solutions of a deterministic differential equation. For comparison, it is useful to mention studies of stochastic paths, which are solutions of stochastic differential equations. Similar computational tools can be used in both cases.

Discussions on paths that are based on stochastic differential equations are widespread. They are coming back to Hibbs and Feynman[14] and to Onsager and Machlup.[15] In the chemical context, Berkowitz et al.[16] discussed reaction coordinates calculated as optimal paths of Brownian particles. Huo and Straub designed a computational procedure based on the above formalism.[17] Pratt proposed a path integral method to compute reaction coordinates following Brownian motions.[18] Olender and Elber[19] showed that the optimal trajectory of Brownian particles within the Ito calculus is the steepest descent path. Dellago et al. and Csajka and Chandler[20] discussed and applied algorithms similar to Pratt.

If we are after a specific type of a Fokker−Planck equation, then the choice of the stochastic differential calculus (and the Jacobian of the transformation) is unequivocal. Hence, the equation for the density is free from the ambiguity of the differential equation that was mentioned above. For example, if we wish to obtain a diffusion equation with a term proportional to the divergence of the force,[8] then the coordinate-dependent Jacobian (and the Stratonovich calculus) should have been kept.[10]

However, in the present manuscript, we do not start from a Fokker−Planck equation. Instead, we are modeling a deterministic differential equation with a stochastic difference equation. The choice of the stochastic calculus should be made based on comparison of accurate atomically detailed trajectories and the approximate trajectories that employ much larger time steps.

II.1.2. Theory of a Single Trajectory. The simplest derivation of the proposed approach to trajectory computations proceeds as follows. We consider a discrete guess of a complete trajectory, $\{X_0, X_1, ..., X_N\}$, where the sequential coordinates are separated by a time step Δt. The guess may be a poor approximation to the true trajectory. We attempt to improve it and to generate a better approximation for the complete traiectory by minimizing the sum of the squares of the errors. We call this sum S.

$$S = \sum_{i=1} {\epsilon_i}^2 = \sum_{i=1} \left[\frac{X_i + X_{i-2} - 2X_{i-1}}{\Delta t^2} + \frac{\mathrm{d}U}{\mathrm{d}X_{i-2}}\right]^2 \qquad (9)$$

The sum S is a function of the trajectory coordinates at all times, $\{X_i\}_{i=1}^N$. A global minimum of S is achieved when all the individual errors, ϵ_i, are exactly 0. The sets of coordinates, $\{X_i^{\mathrm{opt}}\}$, at the minimum of S are thus providing an optimal trajectory. Note that the end points and the total time are fixed during the optimization.

If the evaluation of the second derivative with respect to time is exact, then the global minimum of S would yield an exact trajectory. However, the approximate finite difference derivative means that our results are not necessarily exact even if S is exactly 0.

A few questions come to mind.

(1) What do we loose from the exact trajectory by using the finite difference approximation?

(2) How far is the exact solution from the optimal trajectory?

(3) Can we sample suggestive trajectories (not only the optimal trajectory) to form a set that is likely to include the exact path?

The first question was addressed in ref 7, in which we proved that the minimization of S filters out the atomic motions with frequencies ω that are higher than $\pi/\Delta t$. The filtering prevents many of the stability problems that algorithms based on initial value propagation must face. However, we must keep in mind that the filtered trajectories are approximate. Some of the filtering resemble past approximations. For example, filtering the rapid vibrations of bonds was the target of much research and algorithms (such as SHAKE[21] and RATTLE[22]). Experience suggests that the underlying dynamics does not change profoundly when vibrations of bonds are eliminated. Many of the characteristics of the slow modes are recovered.

The optimization of S as a tool to filter the high-frequency modes does not require the identification of the fast modes to be removed. The automated filtering is in contrast to the filtering in SHAKE or RATTLE and is an advantage. We do not have to know in advance which motions are likely to cause numerical instabilities; the algorithm will find them by itself. However, the “blind” filtering might be a disadvantage; some fast modes (e.g., barrier crossing) may be of interest. Barrier crossing occurs quickly once it is activated. It is a rare but fast motion. The rapid transition from one side of the well to the other (Figure 8 in ref 7) corresponds to a high-frequency motion and is therefore likely to be eliminated. To “see” fast and rare processes, it is necessary to use an inhomogeneous time grid. A large time step, Δt, is used to sample the rare event and a smaller step, δt, is needed to resolve the fast transition.

In practice, such sharp transitions are easy to detect by searching for large structural changes over a single or a few time steps. However, the matching in time of partial solutions with different grids is not trivial and is left for future work. Throughout this manuscript, a uniform time step is used.

The observation that the large-time-step approximation is equivalent to the elimination of fast motions is of some conceptual interest. However, perhaps more relevant is a comparison of the optimal solution with the exact result (that includes the fast components). We attempt to answer this question with a numerical experiment. Let the set $\{X_i^{\mathrm{opt}}\}$ be the discrete trajectory that globally minimizes S ($S = 0$). As argued above, $\{X_i^{\mathrm{opt}}\}$ is not the exact solution since a finite difference formula is used to compute the time derivative, an approximation that results in the removal of rapid motions. A discrete representation of the exact solution, $\{X_i^{\mathrm{exact}}\}$, provides another value for S, which is in general larger than 0. We define a range of acceptable trajectories (with nonzero S). The allowed range of errors in S must be large enough to enable the collection of a significant ensemble of trajectories that include the exact result.

In a previous publication,[7] we made the conjecture that the errors, defined in formula 8, are distributed normally; that is, we assume that

$$P(\epsilon_i) \propto \exp\left[-\frac{{\epsilon_i}^2}{2\sigma^2}\right] \qquad (10)$$

(σ^2 is the (constant) variance of the errors). Below, we present numerical data from atomically detailed simulations. The simulations suggest that the use of a normal distribution to describe the errors is a reasonable approximation.

We consider two systems: valine dipeptide and solvated C peptide. The smaller system is more accessible to detailed analysis, while the analysis of C peptide is more relevant to the application described in the present manuscript.

Valine dipeptide is a small molecule (14 atoms, the CH_n groups are treated like point masses) that can model a short fragment of a protein chain. It was used for tests in our laboratory in the past.[7] The experience and the understanding we gathered on its properties encouraged us to use it (again) as a test. The MOIL molecular dynamics package was used in the simulation.[23] No cutoff distance for nonbonded interactions was employed, and the velocities were assigned from a room temperature (300 K) Boltzmann distribution. After the assignment, a constant energy trajectory was calculated. The total time was 12 ns, the time step was 2 fs, and coordinates of snapshots in time were save each 2 ps.

The set of coordinates as a function of time that we obtained is assumed to be exact. However, when the "errors", ϵ_i, are computed from "exact" trajectory with a large time step ($\Delta t =$ 2 ps), the error ϵ_i ($\epsilon_i = \mathbf{M}(X_{i-2} + X_i - 2X_{i-1})/\Delta t^2 + dU/dX_{i-2}$) is in general different from 0.

The lowest possible value for S is 0. We assume that such a solution exists; that is, a discrete trajectory $\{X_i\}_{i=1}^{N}$ can be found for which S is exactly 0. As a result, the calculated errors for the "exact" trajectory define the minimal width of an error distribution function. Such a distribution function will be centered near the optimal trajectory but still includes the exact trajectory as a part of the set.

In Figure 2a, we show the distribution of the norm of the individual ϵ_{ij} (the index i is for the time and j is for the different coordinates). J is the total number of degrees of freedom (for a single structure), and N is the total number of time steps. TheGaussian distribution that we fitted to the numerical distribution is

$$P(\epsilon_{ij}) \cong \sqrt{\frac{1}{2\pi JN\sigma^2}}\exp\left[-\frac{\sum_j(\epsilon_{ij} - \langle\epsilon\rangle)^2}{2JN\sigma^2}\right] \qquad (11)$$

The parameters of the Gaussian fit are $\langle\epsilon\rangle = -2.3 \times 10^{-14}$ and $2\sigma^2 = 144.579$. The units of ϵ are of force, kcal/(mol Å). The figure suggests that the analytical fit captures successfully most of the features of the numerical distribution. The only significant differences are that the tails of the numerical distribution are longer than those of the Gaussian. As a result, the fitted Gaussian may underestimate the errors. If we prefer to overestimate the errors (instead of underestimating them), it is possible to use another fit (Figure 2b) that overestimates the errors (parameters are $\langle\epsilon\rangle = -2.3 \times 10^{-14}$ and $2\sigma^2 = 216.8685$).

The Gaussian distribution is convenient for further developments of the theory, and we therefore stay with it. As is discussed below, it is possible to work with arbitrary distribution of the ϵ_i provided that the process is Markovian. The exact functional form of the error distribution is not known for the general case. Nevertheless, the Gaussian suggests a simple and a convenient approximation to the error density. The suggestion is consistent with the examples that we have tested so far.

Another intriguing and important question regarding the properties of the ϵ_{ij} is the correlation of the errors. We computed the double average to obtain $C_k = \langle\langle\epsilon_{i,j}\epsilon_{i+k,j}\rangle\rangle$. The average is performed over two indices, i the time index and j the coordinate index. In Figure 3, it is clearly shown that the decay of the correlation is extremely rapid. The relaxation occurs during the first picosecond. Therefore, our estimates provide a lower bound to the relaxation rate. The lack of correlation between different errors in time is an essential ingredient for the approximate theory we outline below, since it makes it possible to describe the process as Markovian.

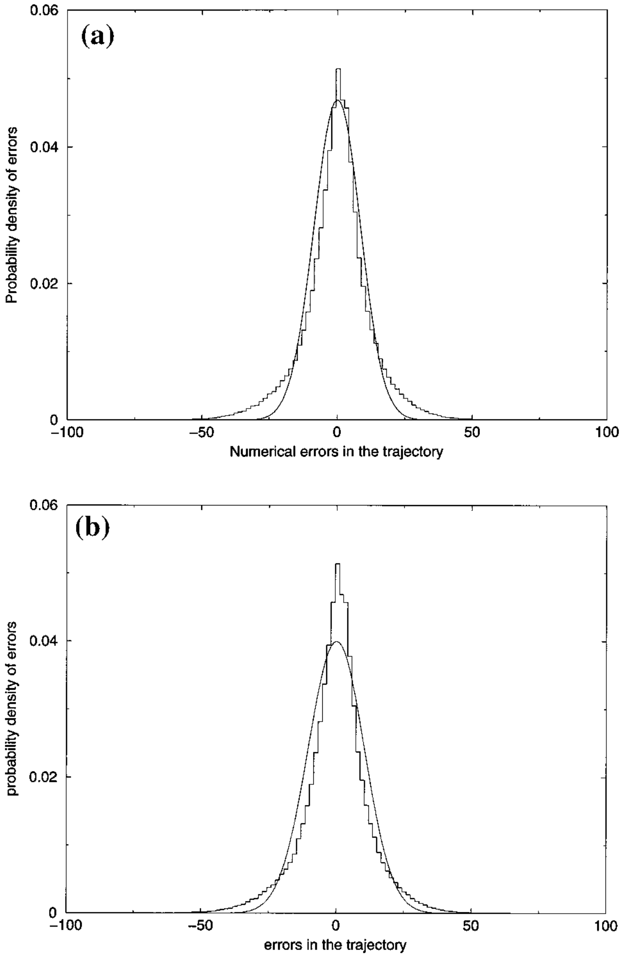

Figure 2. Distribution of the errors, ϵ_{ij}, as extracted numerically from an "exact" trajectory of valine dipeptide. Also shown are Gaussian fits to the numerical histograms: (a) Gaussian fit that underestimates the errors; (b) Gaussian fit that overestimates the errors. See text for more details.

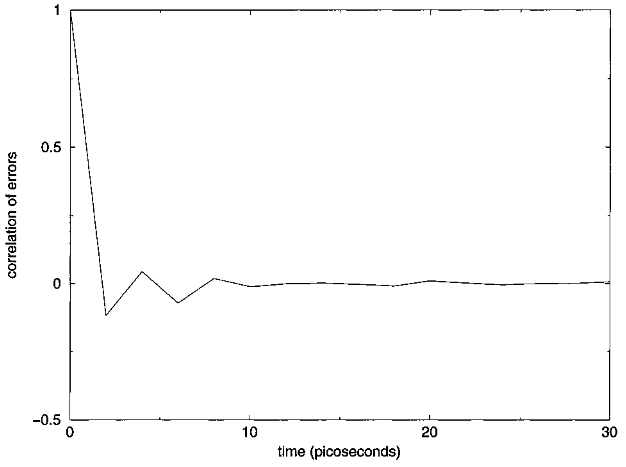

Figure 3. Error correlation function $\langle\epsilon(t),\epsilon(0)\rangle$ calculated numerically from straightforward molecular dynamics trajectory of valine dipeptide.

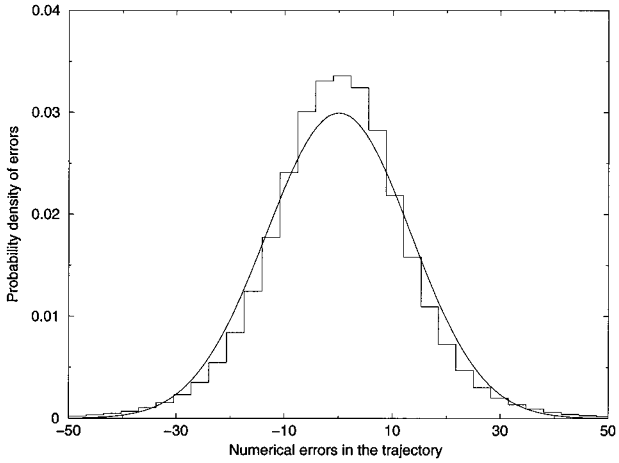

Figure 4. Distribution of the errors, $\epsilon_i^{\text{peptide}}$, as extracted numerically from an "exact" trajectory of C peptide computed with the usual molecular dynamics protocol. Also shown is a Gaussian fit to the histograms. See text for more details.

The investigation of C peptide was performed in a similar way to the study of valine dipeptide. A straightforward molecular dynamics trajectory of C peptide was computed. The peptide was solvated in a box of water molecules that includes 1376 TIP3 water molecules. Periodic boundary conditions were enforced, and the center of mass of the C peptide was constrained to the center of the box. The time step was 1 fs and the total time of the trajectory was 1 ns. Coordinate sets were saved each 2 ps. The coordinates that were saved each 2 ps were also used in the error estimates. The cutoff distances for electrostatic and van der Waals forces were 12 and 10 Å, respectively.

Exploring the numerical errors for C peptide is more related to the present investigation. We therefore attempted to mimic some of the simulation features of the stochastic path. The computations to be described below employed a time scale separation for the solvent (water) and the solute (the C peptide). Explicit time dependence was computed only for the C peptide and not for the water molecules. Therefore, we estimate the errors for the slower motions only, the motions of the C-peptide. That is,

$$\epsilon_i^{\text{peptide}} = \mathbf{M}_{\text{peptide}} \frac{X_{i-2}^{\text{peptide}} + X_{i-1}^{\text{peptide}} - 2X_i^{\text{peptide}}}{\Delta t^2} + \frac{\mathrm{d}U(X_{i-2}^{\text{peptide}}, X_{i-2}^{\text{water}})}{\mathrm{d}X_{i-2}^{\text{peptide}}} \qquad (12)$$

The distribution of errors for $\epsilon_i{}^{\text{peptide}}$ and the Gaussian fit (parameters are $\langle\epsilon\rangle = 7.8 \times 10^{-5}$ and $2\sigma^2 = 354.722$) are shown in Figure 4.

Note that the distribution is more "Gaussian-like" compared to the distribution that was obtained from the valine dipeptide simulations. This observation is encouraging since we are primarily interested in larger systems. Nevertheless, there are similar deviations from the Gaussian behavior in the small and large molecular systems perhaps indicating a conceptual difference.

The fit in Figure 4 overestimates the errors; we do not show a fit that underestimate the errors (like in the valine dipeptide). Note also that the values of the variances (for valine and C peptide) are not significantly different. This observation suggests that empirical modeling of the variance might be successful.

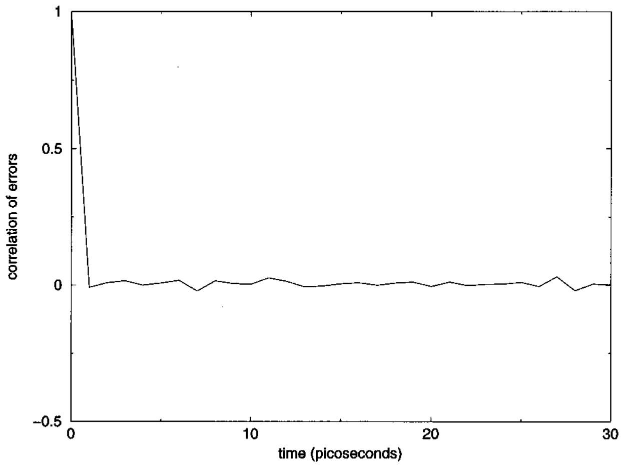

Figure 5. Error correlation function $\langle\epsilon(t),\epsilon(0)\rangle$ calculated numerically from straightforward molecular dynamics trajectory of C peptide. See text for more details.

The last characteristic that remained to be examined was the correlation of the errors. Similarly to the case of valine dipeptide, the error correlation function decays extremely rapidly to 0 (Figure 5).

How is it possible to understand the numerical results? Consider the ensemble average $C_k = \langle\langle\epsilon_{i,j}\epsilon_{i+k,j}\rangle\rangle$, which is computed using an exact trajectory. Each of the components ϵ_l moves at a potentially different rate (some modes are fast and some modes are slow), and therefore, the partial average, $\langle\epsilon_l(t)\epsilon_l(t + \Delta t)\rangle$, is expected to be "uncorrelated" with the average of a different coordinate $\langle\epsilon_n(t)\epsilon_n(t + \Delta t)\rangle$. The lack of spatial memory assumes short-range correlation. This is true for the majority of the protein motions excluding the slowest degrees of freedom.

Once the idea of the minimization of errors and the definition of the target function S (eq 8) are established, we are ready (in principle) to design and apply numerical algorithms to minimize S as a function of all the X_i variables. However, we can further use our knowledge on the distribution of the errors ϵ_i to write a statistical summation of paths. Let the probability of observing an error ϵ between ϵ_i and $\epsilon_i + \mathrm{d}\epsilon_i$ at t_i be $P(\epsilon_i)\,\mathrm{d}\epsilon$. The probability density of observing a time sequence of errors is $\hat{P}(\epsilon_1, ..., \epsilon_N) = \prod_{i=1,...,N}P(\epsilon_i)$. The last formula is equivalent to the statement that the stochastic variables $\{\epsilon_i\}_{i=1}^N$ are uncorrelated in time. However, the coordinates of the trajectory are more useful quantities compared to the errors. We therefore convert the formula to coordinate space (and the coordinates are correlated in time), $\hat{P}(\epsilon_1, ..., \epsilon_N) = \prod_{i=1,...,N}P[\epsilon_i(X_{i-2}, \mathrm{X}_{i-1}, X_i)]$. The optimal trajectory can now be obtained from the optimization of the target function

$$\hat{S} = -\log[\hat{P}(\epsilon_1, ..., \epsilon_N)] = \sum_{i=1,...,N} -\log[P(\epsilon_i)] \qquad (13)$$

This formula is reduced to eq 8 in the special case in which $P(\epsilon_i)$, is a Gaussian. For simplicity and lack of further knowledge, we are using a Gaussian formula. However, if better or alternative expressions are given for the error distribution for a single step, the alternative form (based on formula 13), can be employed as well. From the above discussion, it is also clear that a crucial element of the derivation of $\hat{S}$ and S is the lack of correlation between errors at different times.

Another comment concerning the theory of a single trajectory deals with degeneracy and quasiequilibrium. In many cases, the trajectories are highly degenerate. This phenomenon is easier to understand using an example. Consider a small molecule solvated in water. Our prime interest is in the trajectory of the solute, while the solvent molecules provide electrostatic shielding, hydrogen bonding, van der Waals interaction, etc. for the motion of the embedded molecule.

In the study of the solute, we are less interested in the individual trajectories of the water molecules. In fact, it is possible that similar trajectories of the solute will be found, while the individual trajectories of "labeled" water molecules will be quite different. For example, the solute does not care if a water molecule number *55* is hydrogen bonded to it or number 1500 is taking its place. As long as hydrogen bonding is provided the solute is happy. Obviously, the trajectories of molecules 55 and 1500 will be quite different in both cases. In that sense, the trajectories are degenerate.

In principle, we could have calculated all of the quasidegenerate trajectories, but the allowed range of errors is large and many similar trajectories (in the path of the solute) will be accepted. The large number of trajectories is a significant computational and analysis burden. It would be nice to sum up trajectories similar in the solute pathway into one compact computation. For that purpose, we suggest the use of an approximation that is based on time scale separation.

We can differentiate between the time scale that is required for the solute to "react" and the time scale in which the solvent, the water molecule, responds to the changes in the conformation of the solute. Here, we consider the case in which the response is faster than the conformational transition of the solute. The assumption of slow solute seems appropriate for the system at hand (folding of C peptide).

If the response of the water molecules is much faster than the typical rate of the peptide motions, we argue that after each time step, Δt, the water rapidly maintained an equilibrium configuration (assumed canonical) near the "fixed" state of the peptide. Hence, the weight of a water configuration, X_{w}, with a given peptide configuration, X_{p}, follows from a Boltzmann distribution, $\exp[- U(X_{\mathrm{w}},X_{\mathrm{p}})/(k_{\mathrm{B}}T)]$, where X_{p} is considered a vector of parameters. The individual weight of an error in the peptide coordinate is therefore modified from $P(\epsilon_i)$ to

$$P(\epsilon_{i\mathrm{p}}) = \frac{\int \mathrm{d}X_{i-2,\mathrm{W}} \exp\left[-\frac{U(X_{i-2,\mathrm{W}},X_{i-2,\mathrm{p}})}{k_{\mathrm{B}}T}\right] P(\epsilon_i)}{\int \mathrm{d}X_{i-2,\mathrm{W}} \exp\left[-\frac{U(X_{i-2,\mathrm{W}},X_{i-2,\mathrm{p}})}{k_{\mathrm{B}}T}\right]} \tag{14}$$

The above formula means that the error in the peptide trajectory is averaged over all possible configurations of the solvent using thermal weights for water coordinates. Besides the ease of interpretation and the smaller number of plausible trajectories that are obtained, the computations are more efficient. To optimize S, we need to compute its derivatives with respect to coordinates and therefore the derivatives of ϵ_i.

Computing the Hessian matrix (the derivative of the force) is expensive for macromolecules since there are many elements to compute and manipulate. In the least compressed form, the number of elements in the matrix is proportional to the square of the number of particles. The Boltzmann weight that is used for the water molecules eliminates the need to calculate elements of the second derivative for water−water interactions. This reduces dramatically the computational cost.

The calculations of the folding of C peptide to be described in the application part employ the above time scale separation and an assumed Gaussian form for the error density of peptide trajectories. There, we define the following functional for which we seek a minimum.

$$\hat{S} = \sum \frac{1}{2\sigma^2}\left[\mathbf{M}\frac{X_{i-2,\mathrm{p}} + X_{i,\mathrm{p}} - 2X_{i-1,\mathrm{p}}}{\Delta \mathrm{t}^2} + \frac{\mathrm{d}U}{X_{i-2,\mathrm{p}}}\right]^2 + \frac{U(X_{i-2,\mathrm{W}};X_{i-2,\mathrm{p}})}{k_{\mathrm{B}}T} \tag{15}$$

The functional $\hat{S}$ can be also the target of direct global optimization, which is the approach we took in the study of C peptide. This concludes the discussion on formulas used to calculate individual trajectories. Below, we consider the computations of an ensemble.

II.1.3. Theory of Ensemble of Trajectories. Sampling trajectories for thermodynamic and kinetic averages requires knowing in advance the weight of individual trajectories and perhaps also the weights of the configurations. For example, in constant-energy molecular dynamics trajectories, all the configurations have exactly the same weights. Unfortunately, computing the weights of the trajectories is not always so simple. In our case, trajectories with large values of $\{\epsilon_i\}_{i=1}^N$ must have smaller weights than trajectories in which the "errors" are 0. Some theoretical consideration must therefore be given to the weights of the trajectories calculated with the SDE. The weight were already mentioned and briefly discussed in refs 7 and 9. Here, we provide a complete formalism with a few new twists.

It is convenient to consider the probability for coordinates instead of errors (we use the Gaussian form for the probability density). The explicit form for the error distribution density makes the expression below more concrete. We further consider a canonical ensemble. The canonical distribution implies that the weight of the starting configuration, X_1, is determined by the Boltzmann factor $\exp[-E(X_1)/(k_{\mathrm{B}}T)]$ where $E(X_1)$ is total-energy. The weight density of a trajectory is

$$W(X_1, ..., X_N) = \exp\left[-\frac{E(X_1)}{k_{\mathrm{B}}T}\right] \frac{\exp\left[-\frac{S(X_1, ..., X_N)}{k_{\mathrm{B}}T_{\mathrm{eff}}}\right]}{\int (\prod_{i=2,N-1} \mathrm{d}X_i) \exp\left[-\frac{S(X_1, ..., X_N)}{k_{\mathrm{B}}T_{\mathrm{eff}}}\right]} \tag{16a}$$

For brevity, we define:

$$\exp\left[-\frac{S}{k_{\mathrm{B}}T_{\mathrm{eff}}}\right] \equiv \exp\left[\frac{-1}{2NJ\sigma^2}\sum_i \left[\mathbf{M}\frac{X_{i-2} + X_i - 2X_{i-1}}{\Delta t^2} + \frac{\mathrm{d}U}{\mathrm{d}X_{i-2}}\right]^2 \Delta t\right]$$

$$k_{\mathrm{B}}T_{\mathrm{eff}} = \frac{2NJ\sigma^2}{\Delta t} \tag{16b}$$

There is a nasty normalization factor that is hard to compute,

$$N(X_1, X_N, t) = \int (\prod_{i=2,N-1}) \exp\left[-\frac{S(X_1, ..., X_N)}{k_{\mathrm{B}}T_{\mathrm{eff}}}\right]$$

This is a result of Lagrangian mechanics that we are trying to approximate. In Lagrangian mechanics, only one trajectory starts and ends at the same coordinate with a predetermined total trajectory time. The integral over all paths guarantees the appropriate normalization. We need to compute this integral (which depends on X_1, X_N, and t) to know the weight of each trajectory.

This is an expression that is inconvenient for use in Monte Carlo procedures or molecular dynamics simulation. In both cases, we needed an effective energy, E_{eff}, so we could compute the Boltzmann factor or solve equations of motions. Attempting to write eq 16a as a Boltzmann factor with an effective energy, we have

$$W(X_1, ..., X_N) = \exp\left[-\frac{E(X_1)}{k_BT} - \frac{S(X_1, ..., X_N)}{k_BT_{\mathrm{eff}}} - \log[N(X_1,X_N,t)]\right] \quad (17)$$

Hence, in order to compute the weight of a single trajectory, we need to compute first the normalization function! While a feasible proposition for small molecular systems and systems with a few degrees of freedom, it is a discouraging result as a starting point for computations in large molecular systems.

An alternative approach to obtain the weight of the trajectory is to make the following argument. Up to now, we consider the starting and the ending position as exact. It is not symmetric and probably impractical to insist on high accuracy at the end points while admitting considerable errors in the trajectory in between. We might let the end points be loose, allowing for errors at the end points in accord with the weighting function we have. The errors at the starting point may lead to energy variations. It is therefore useful to weight each trajectory by a Boltzmann factor of its initial energy; at the limit of small time step, the error will vanish and we shall obtain a sum of trajectories weighed by their corresponding energies.

$$\tilde{W} = \frac{\exp\left[-\frac{E(X_1)}{k_BT} - \frac{S(X_1,...,X_N)}{k_BT_{\mathrm{eff}}}\right]}{\int(\prod_{j=1,...,N} \mathrm{d}X_j) \exp\left[-\frac{E(X_1)}{k_BT} - \frac{S(X_1,...,X_N)}{k_BT_{\mathrm{eff}}}\right]} \quad (18)$$

which suggests that straightforward Monte Carlo or molecular dynamics run can now be employed using as an effective energy $E(X_1) + (T/T_{\mathrm{eff}})S(X_1,..., X_N)$. Relatively narrow basins (reflecting the expected errors) near X_1 and X_N must be used in the computations.

We note that the trajectory we want to approximate is a constant-energy trajectory. The energy is independent of time and can therefore be calculated at an arbitrary point along the trajectory. It is not necessary to compute the energy only at the initial coordinate; we could compute it anywhere! In particular, we write

$$E(X_1) = \frac{1}{N}\sum_{i=1,...,N} E(X_i)$$

Writing the energy as an average over the trajectory does not change the result if the trajectory was solved exactly. However, in our case, it is not. Therefore, differences are expected between the results of the two formulations. The average is more symmetric and leads to a computationally more convenient formula as follows:

$$\tilde{W} = \frac{\exp\left[-\sum_{i=1,...,N}\frac{E(X_i)}{Nk_BT} - \frac{S}{k_BT_{\mathrm{eff}}}\right]}{\int(\prod_{j=1,...,N} \mathrm{d}X_j) \exp\left[-\sum_{i=1,...,N}\frac{E(X_i)}{Nk_BT} - \frac{S}{k_BT_{\mathrm{eff}}}\right]} \quad (19)$$

This concludes the section on computations of weights of trajectories, which makes it possible to compute averages over trajectory properties. All the usual quantities are accessible to statistical mechanics methods, such as the free energy perturbation, rate calculations, and differences in rates of similar species. As an example, consider the following reaction: A → B. To compute the number of trajectories that started at A and made it to B after time t, we consider the average of a product of two box functions, $H_A(X_1)$ and $H_B(X_N)$. The first box function is equal to 1 if X_1 is in the A domain and 0 elsewhere. Similarly, the second box function is 1 when X_N is in the B domain and 0 elsewhere. If the trajectories are already sampled with the correct weight built in, for example, using molecular dynamics with the effective energy as a potential, the fraction of "successful" trajectories is easily calculated. It is

$$f = \frac{\sum_{k=1,...,K} H_A(X_1)H_B(X_N)}{K}$$

where K is the total number of sampled trajectories.

II.2. Numerical Aspects. All the computations described in the next paragraph were performed using the molecular dynamics code MOIL.[23] The graphic analysis was performed with the visualization program MOIL-View.[24] Both programs are available free of charge on the Internet and are released with source codes. MOIL is a package of separate FORTRAN programs to perform molecular dynamics and modeling studies. One of these programs is "sto", which was used on the TERRA parallel computer to perform the present calculation (see below).

In previous publications,[7] only small molecular systems were investigated using the new methodology. In the next section, we shall present folding trajectories of C peptide, a nontrivial calculation that is also of considerable biological interest.

It is useful to have an estimate of the computational resources that are required in the study of such a large system. These resources are indeed quite extensive. Let the number of particles in the physical system be L. The computational effort is dominated by the calculation of energy. Let the number of operations that are required for a single energy evaluation be a function of L, say $O(L)$. In normal circumstances, this function grows faster than L and slower than L^2. The exact rate depends on the model at hand.

In straightforward molecular dynamics, we compute N steps as described in eq 2. The computational effort to generate a trajectory of total time $N\Delta t$ is therefore $NO(L)$. The protocol described in the present manuscript is based on a minimization algorithm of an effective energy. Once the neighborhood of the optimum trajectory was located, sampling of trajectories near the optimal path may follow. Hence, the estimate we provide below is for maximum effort. It might be considerably easier to generate another trajectory after the optimum was located.

It is difficult to estimate how many steps are required to reach an acceptable optimal trajectory. Clearly, the number of steps depends on the system at hand, the optimization protocol (e.g., the use of multigrid techniques can profoundly enhance the speed of the optimization[7,25]), and the requirements of accuracy.

If the goal of the computations is trajectory sampling (rather than an evaluation of a single trajectory), then a very accurate minimization may be unnecessary.

Let the number of minimization steps that we use be M; in typical calculations, it is of order of several thousands. The computational effort for a single optimization step of the trajectory weight is proportional to (a) the number of intermediate structures and (b) the computational resources required for a single energy evaluation. The derivatives of S require the computations of the second derivatives of the potential, which are more expensive than first derivative (force) calculations. In practice (using cutoff distances for nonbonded interactions), the matrix of second derivatives is sparse and the computational efforts are only several times (say, n times) more expensive than the efforts associates with the evaluation of the usual energy. As a result, the computational effort is proportional to $MN'nO(L)$, where N' is the number of grid points in the SDE computation. The ratio of the computational efforts of the usual MD approach and the SDE method is therefore

$$\frac{NO(L)}{N'MnO(L)} = \frac{N}{N'Mn}$$

The SDE approach is considerably less efficient than MD when the same size of a time step its used (i.e., when $N = N'$). Only when the time step used in SDE is about nM times the step used in MD does the SDE method becomes competitive. This factor is expected to be (at least) in the thousands. The usual MD employs steps of femtoseconds, and therefore, SDE is computationally justified only when the step used in SDE is (at least) in the picosecond time domain. The time step that was used in the SDE study for folding of C peptide is of 500 ps.

In the computations, we took advantage of extensive parallelism. Since each structure is coupled (via the time derivative) to only four nearest structures, the communication overhead is minimal. The parallelization protocol is described below.

We divide the sum that makes the functional S between the processors; that is, for P processors and N structures, we have

$$S = \sum \epsilon_i^{\,2} = \sum_{p=1,\ldots,P} S_p = \sum_{p=1,\ldots,P} \Big[\sum_{i=1,\ldots,N_p} \epsilon_i^{\,2} \Big]$$

$$N = \sum_{p=1,\ldots,P} N_p \qquad (20)$$

Each of the processors has N_p "active" structures, which are allowed to change. Since the computations of ϵ_i require three structres (to estimate the time derivative), each of the processors need $N_p + 4$ structures, of which only N_p are modified in an optimization step. Processor p requires two structures from processor $p - 1$ (that only $p - 1$ is allowed to change) and two structures from processor $p + 1$ (Figure 6).

In common parallel environments that employ MPI or PVM, it is necessary to make $2P$ data transfers between the processors. As a result, the communication effort is proportional to the number of processors. This communication overhead is sound since the computation time is dominated by the calculations of the derivatives of the action. However, a better performance is still possible.

The TERRA 2000 superworkstation makes it possible to use an extension to the MPI operation. In this extension, the processors communicate simultaneously with the nearby processors. For example, all calls $p \rightarrow p - 1$, $p = 1, ..., P$ is a single operation. Note that the machine is arranged in a ring and $1 \rightarrow 0$ is the same as $1 \rightarrow 16$. Hence, the communication is done in parallel as well, making the effort independent on the number of processors. The TERRA was used in the computations of the folding of C peptide, which will be described in the next section. The speed up on a 16 node machine was essentially 16. The computations described in the next section require about 1 month. A single new trajectory was computed in a day.

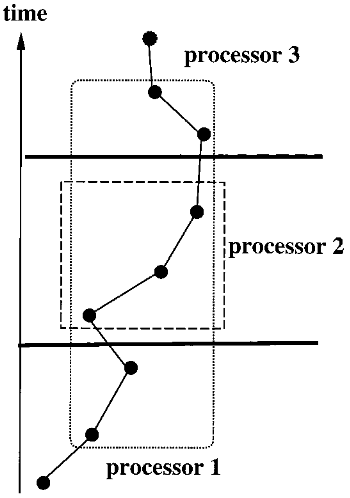

Figure 6. Schematic description of the communication setup in the parallel computations of the action of the trajectory. Processor 2 is discussed in more detail. The dashed box enclosed the (three) structures that processor 2 is allowed to change. The dotted line enclosed four additional structures (two from processor 1 and two from processor 3), which are required to compute the time derivative. For accurate computations of the derivatives, the structures are communicated after each modification of the coordinate sets. However, in principle, the communication can be performed less frequently.

III. Folding of C Peptide

The C peptide is a short fragment of ribonuclease A that was the target of numerous investigations, primarily led by the pioneering work of Baldwin and co-workers.[26] Ribonuclease A is a robust folder and was used by Anfinsen[27] in his pioneering study on determinants of protein structures. The conformation space available to the unfolded protein chain is very large. The large space poses an intriguing question: what is the mechanism in which the structure is formed?

One possible attempt to identify plausible folding pathways is to search for nucleation sites. We loosely define nucleation sites as fragments of the protein chain that have a significant tendency to form a structure. The structure they fold into is also their final configuration in the (complete) folded protein. The tendency of the nucleation sites to structure is assumed stronger than the tendency of other potential fragments. Only after the nuclei are roughly folded, the process continues to the final structure or to form larger "nuclei". Minimal models of nucleation are covered in the excellent paper by Guo et al.[31]

The best way of probing nuclei experimentally is via short-time spectroscopy, watching early events in the folding of the complete protein chain. Direct observations of folding kinetics are beautiful (but difficult) experiments and were attempted in the last few years for a few systems.[28] Another approach is to consider fragments of the protein chain and to examine if these fragments show significant tendency to structure. One of the first and the most comprehensively studied protein fragment is the C peptide.

In the crystal structure, the C peptide (embedded in ribonuclease A) is a partial helix.[29] NMR studies of the C peptide

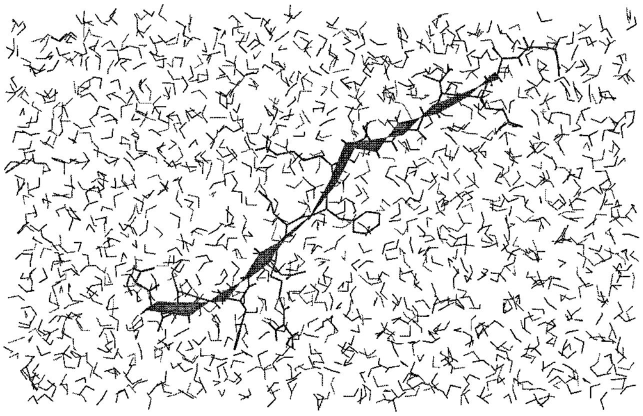

Figure 7. Initial unfolded configuration of C peptide (an extended chain) in the box of water that was used in the simulation. Periodic boundary conditions are employed.

suggest that its structure (without the rest of the protein) is a partial helix as well. The similarity of the structures of C peptide, with and without the rest of the protein chain, supports the picture of a nucleation site.

To the best of our knowledge, there are no direct experimental measurements of the kinetics of C peptide folding. The information available is on structure and stability of the original C peptide and variants. Nevertheless, extrapolating from experimental results on formation of helixes in other peptides,[28] it is likely that the folding rate of the C peptide is in the submicrosecond time scale. Recent theoretical arguments also suggest time scales of tens of nanoseconds for helix formation.[32]

We did not use a "native" form of the C peptide but employed instead a chemically modified variant with an enhanced helix affinity. In accord with the observation of Baldwin et al., we used a chemical modification of the N terminal to succinate acid that was shown (experimentally) to enhance the probability of helix formation.[30] Hence, the sequence of the C peptide variant that we used is Suc-Ala-Glu-Thr-Ala-Ala-Ala-Lys-Phe-Leu-Arg-Glu-His-Met-Asp-Ser. The modified peptide carries a total negative charge of 3. We therefore added to the system three sodium counter ions. The peptide and ions were embedded in a box of 1376 water molecules (box size of 54 × 33 × 28 Å^3). The cutoff distances for nonbonded interactions were 9 Å for van der Waals forces and 10 Å for electrostatic interactions. Periodic boundary conditions were used during room-temperature simulations.

Here, we compute the kinetics of C peptide folding into a complete helix. The details of the computations are as follows. We simulate the computations of an extended chain configuration of the C peptide, using trajectories of 16.5 ns only. While this time is probably too short, it provides some hints into the mechanism of the process. Even if the time scale is of order of 100 ns, it is possible that some room-temperature trajectories will fold at the 10 ns time scale. Obviously, in our simulations the C peptide "succeeds" to fold during this short time (because we forced it to). The time step was 500 ps, and 34 structures were used on the 16 processors of the Terra computer. Hence, there are two "active" structures on each processor. The first and last structures are fixed and were added to the first and last processor.

The starting unfolded conformation was taken as an extended chain equilibrated in a box of water (Figure 7). The final conformation was built as an ideal α-helix. The ideal helix was minimized and equilibrated in the same box of water. The original path was obtained from a straight-line interpolation of the extreme configurations using only 11 structures. The linear interpolation was followed by minimization and independent short equilibration runs of the individual structures. To construct the complete path (with 34 structures), we used duplicates of the same coordinate sets. Hence, if we indexed the final coordinates according to the starting path, it is (1,1,1,2,2,2, ..., 11,11,11), where the first and last structures are fixed (to make exactly 34 configurations, the middle structure was multiplied four times). This choice provides an additional test for the deviation of the optimized path from the initial guess. After the optimization, we do expect the coordinates to be distributed more uniformly along the trajectory or at least with no relation to the initial setup.

We generated suggestive trajectories of folding by simulated annealing of the functional in formula 15. The "temperature" for the annealing protocol has little to do with the physical temperature (300 K) and is reflecting more the estimate of the errors. In the present investigation, we did not have information on the errors for a step of 500 ps. Moreover, at present, we were interested in plausible optimized trajectories and we therefore annealed the action many times. Hence, we find the set of optimal trajectories. Ideally, the action for all these optimal trajectories should have been 0. In practice, we were not able to reduce S exactly to 0.

The annealing temperature to optimize S was set to 30 K following some trial and errors searching for good optimization conditions. The system with S for a potential energy was cooled linearly to a "temperature" of 0. The cooling cycles consist of 1000 minimization steps. In about 30 cycles, the value of the peptide action (i.e., not including the water potential energy) was reduced by 5 orders of magnitude to about 10. At that point, we resort to cycles of simulated annealing runs of 1000 steps. At the end of each simulated annealing cycle, we obtain a new trajectory. A total of 31 trajectories of 16.5 ns each were used in the analysis described below.

In Figure 8, we show a typical schematic path of a trajectory from an extended chain to a folded structure. The path suggests that the structure is initiated at the edge of the chain. The start of the structure at the edge of the peptide chain repeats in all the annealed trajectories that we obtained. However, the trajectories that we examined are still correlated since the annealing cycles are not very long. As a result, we cannot eliminate the possibility that other nucleation sites were not sampled during our limited exploration of trajectory space.

An interesting quantity to look at is the end-to-end distance EED (the distance between the first and last C_α). The EED is a possible measure for a "reaction coordinate" from the extended chain (large EED) to the helix (smaller EED). In Figure 9, we show a projection of different folding trajectories onto the EED. The projections of a significant number of trajectories suggest a "waiting state" in which the EED does not change appreciably for a considerable length of time (a few nanoseconds). Hence, using the EED, it is possible to identify an "intermediate" along this reaction coordinate that lives for a substantial length of time. In Figure 10, we show one of the structures from the ensemble that makes this intermediate. The conformation is that of a partial helix. This is in accord with the X-ray structure and the NMR data for the equilibrium structure, which is a partial helix as well.

It must be emphasized that the state of a partial helix is not a single structure. Instead, it is a collection of many conformations in which only a part of the helix is formed correctly. While we found more structures in which the helix is forming from the edge, there are numerous exceptions to this rule, and it is

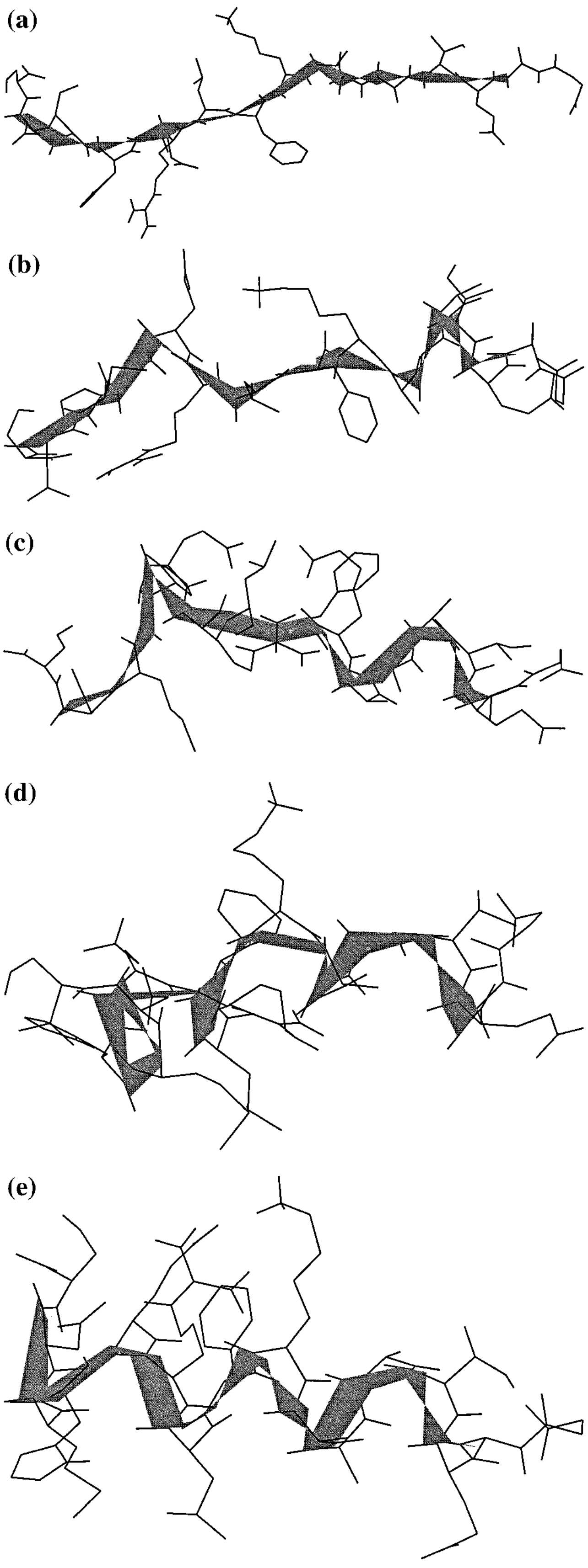

Figure 8. Set of backbone structures following a typical folding trajectory.

also possible to find (infrequent) structures in which a partial helix is observed at the middle of the chain.

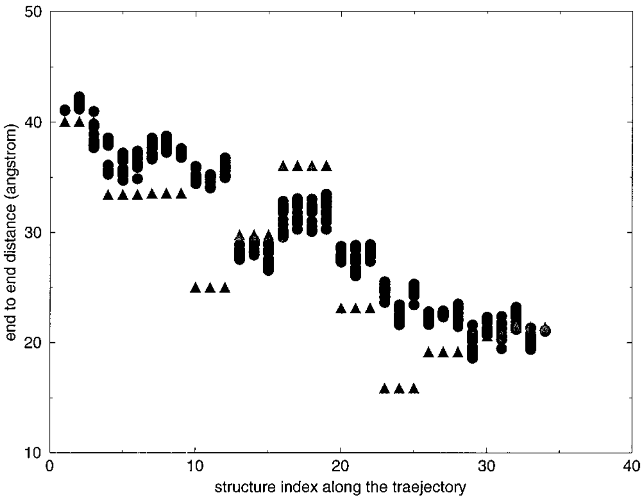

Figure 9. Summary of the end-to-end distance computations for the 31 trajectories. The distances are recorded as a function of the structure index. The total trajectory length was 16.5 ns. The triangles correspond to the distances of the initial path (before the action optimization) and are provided to demonstrate the significant changes that occur in the trajectory during the simulated annealing. Approximately from structure 18 to structure 30, the configurations can be defined as partial loose helixes. The structures are not the same, but they maintain a significant fraction of and resemblance to a helix.

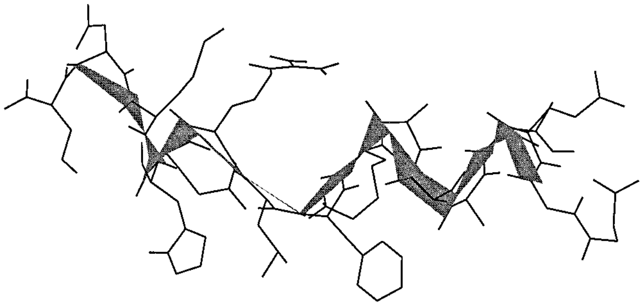

Figure 10. A snapshot of a partial helix from the thermodynamic intermediate state of a "partial helix". We emphasize that the intermediate cannot be identified as a single structure but instead consists of a large number of conformations that maintain some degree of similarity to a helical conformation.

The ability to identify a relevant intermediate is perhaps another intriguing property of the SDE approach. The two end points (the extended chain and the "ideal" helix) were probably not stable states. Energetically and entropically, the extended chain conformation is not favorable. The ideal helix might be favorable energetically, but entropy is against it. According to experimental data,[29] the partial helix is the most stable state. The SDE approach forces the system to start and end in different configurations. However, if the thermodynamically most stable set of configurations does not include one of the end coordinate sets, the "chain" of structures (the trajectory) will seek a better state to be in and will aggregate in a thermodynamically more favorable domain of the energy surface (Figure 11).

To probe the suggestion that the partial helix is indeed the most stable conformation of the C peptide, we also performed a straightforward molecular dynamics trajectory of C peptide. The trajectory was performed in an identical box of water molecules and was of 2 ns. The final structure of the molecular dynamics simulation is shown in Figure 12. It is clearly seen that the picture of a stable partial helix was obtained in the molecular dynamics simulations as well. This supports the notion that the "intermediate" that we observe in the simulation of C peptide is the most stable thermodynamic state in accord with available experimental data.[29]

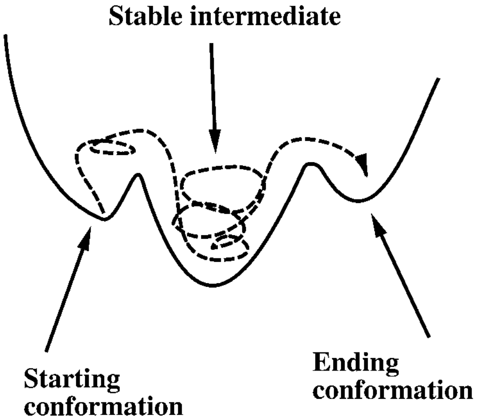

Figure 11. Schematic drawing of aggregation of the trajectory in a thermodynamically stable state, which in our case is the partial helix.

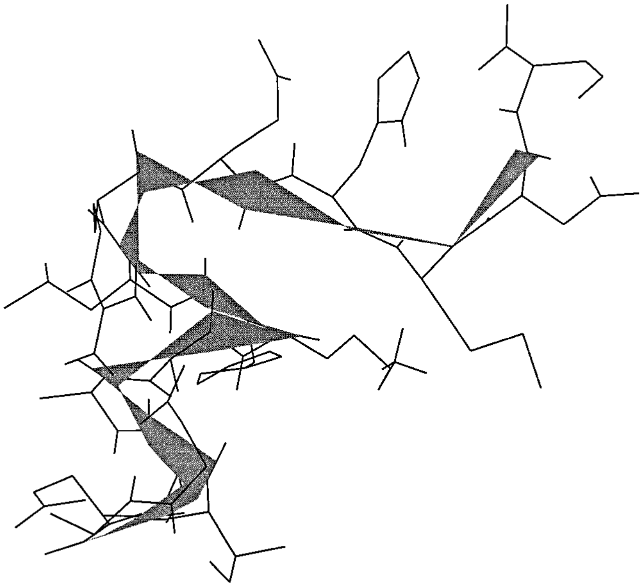

Figure 12. Stable structure of C peptide as suggested by the molecular dynamics simulations.

Another feature that we can extract from the simulation is the relative flexibility of the side chains and the backbone. We examine snap shots in time of different trajectories (Figure 13). While the backbone structures vary approximately in the same rate in all the trajectories (suggesting relatively narrow and stiff pathway), the side chain positions are significantly less correlated between different trajectories.

The observation of more rapid and broader fluctuations of the side chains compared to backbone motions is similar to investigations of other peptides[4] using different tools. Studies of the peptide SYPFDV clearly demonstrated that the side chains fluctuate rapidly and sample larger space than the backbone. The studies further showed that the most stable configuration of the side chain does not correspond to a deep free energy minimum.

IV. Final Remarks

In this paper, we present a detailed practical scheme and the theory of how it is possible to compute long-time trajectories. While elements of the new computational methods were presented elsewhere, we provided here a more complete picture. We discussed the origin of the "noise" in the solution and how it can be computed from "first principles". We further outlined how an ensemble of trajectories can be computed and how time scale separation and adiabatic approximation might be built into the same approach. Finally, we studied a problem of biophysical interest, the folding of C peptide, and we demonstrated (in accord with experiment and straightforward molecular dynamics simulations) that the most stable structure of the C peptide is of a partial helix. The identification of the most stable structure was done without assuming it to be at the end points, suggesting another test and a useful aspect of the stochastic difference equation. Further work on yet larger systems and at yet longer times is possible and is being carried out in the authors' laboratory.

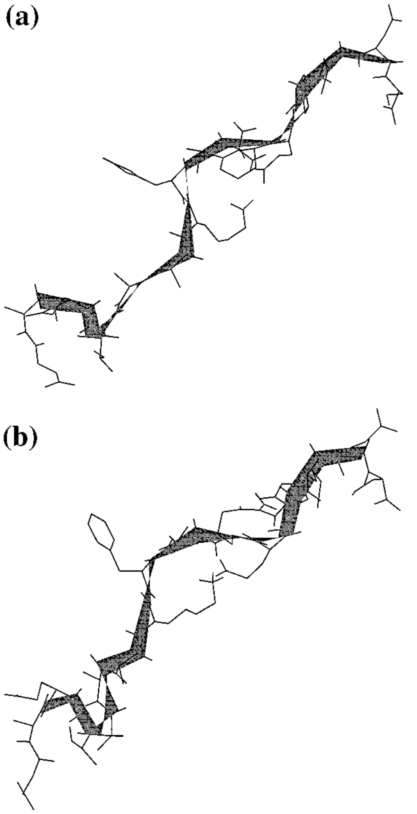

Figure 13. Structures from different trajectories ((a) and (b)) suggesting that the backbone is about the same (snapshots at the same trajectory time) but the side chains are widely different. The structures were picked at the early beginning of the folding process in which the chain is extended. Nevertheless, a small nucleation site at the edge of the chain is already recognized, and the side chains adopt widely different conformations.

Acknowledgment. This research was supported by a grant from the Israel Academy of Sciences and the National Institutes of Health to R.E. The Fritz Haber research center is supported by the Minerva fund. The generous support of the Terra Computer Company is gratefully acknowledged.

References and Notes

(1) Frenkel, D.; Smit, B. *Understanding Molecular Simulation: From Algorithms to Applications*; Academic Press: San Diego, 1996.

(2) Gibson, Q. H.; Regan, R.; Elber, R.; Olson, J. S.; Carver, T. E. Distal Pocket Residues Affect Picosecond Recombination in Myoglobin: An experimental and Molecular Dynamics Study of Position 29 Mutants. *J. Biol. Chem.* **1992**, *267*, 22022.

(3) Li, H.; Elber, R.; Straub, J. E. Molecular Dynamics Simulations of NO Recombination to Myoglobin mutants. *J. Biol. Chem.* **1993**, *268*, 17908.

(4) Mohanty, D.; Elber, R.; Thirumalai, D.; Beglov, D.; Roux, R. Kinetics of Peptide Folding: Computer Simulations of SYPFDV and peptide variants in water. *J. Mol. Biol.* **1997**, *272*, 423−442.

(5) Hille, B. *Ionic Channels of Excitable Membranes*; Sinauer Associates: Suderland, 1992.

(6) Creighton, T. E. *Protein Folding*; W. H. Freeman: New York, 1992.

(7) 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.

(8) Gardiner, C.W. *Handbook of stochastic methods for physics, chemistry and natural sciences*; Springer-Verlag: Berlin, 1990.

(9) Elber, R.; Roux, B.; Olender, R. Application of a Stochastic Path Integral to the Computations of an Optimal Path and Ensembles of Trajectories. *Proceeding. Of the Berlin Conference on Algorithms for Molecular Dynamics*, in press.

(10) Kleinert, H. *Path Integrals in Quantum Mechanics, Statistics, and Polymer Physics*; World Scientific: Singapore, 1995.

(11) Landau, L. D.; Lifshitz, E. M. *Mechanics*; Pregamon Press: Oxford, 1959.

(12) Verlet, L. *Phys. Rev.* **1967**, *159*, 98.

(13) David Shalloway is testing the quality of molecular dynamics trajectories with ϵ_i (private communication). The Gauss variation principle can be found in the following: Lanczos, C. *The Variational Principles of Mechanics*; University of Toronto Press: Toronto, 1970.

(14) Feynman, R. P.; Hibbs, A. R. *Quantum Mechanics and Path Integrals*; McGraw Hill: New York, 1965; Chapter 12.

(15) Onsager, L.; Machlup, S. *Phys. Rev.* **1953**, *91*,1505; **1953**, *91*,-1512.

(16) Berkowitz, M.; Morgan, J. D.; McCammon, J. A.; Northrup, S. H. Diffusion-controlled reactions: A variational formula for optimum reaction coordinate. *J. Chem. Phys.* **1983**, *79*, 5563−5565.

(17) Huo, S.; Straub, I. E. *J. Chem. Phys.* **1997**, *107*, 5000.

(18) Pratt, L. R. *J. Chem. Phys.* **1986**, *85*, 5045.

(19) Olender, R.; Elber, R. Yet another look at the Steepest Descent Path. *J. Mol. Struct. (THEOCHEM)* **1997**, *398/399*, 63−72.

(20) Dellago, C.; Bolhuis, P. G.; Csajka, F. S.; Chandler, D. Transition Path Sampling and the Calculation of Rate Constants. *J. Chem. Phys.* **1998**, *108*, 1964−1977. Csajka, F. S.; Chandler, D. Transition Pathways in many-body system: Application to hydrogen-bond breaking in water. *J. Chem. Phys.* **1998**, *109*, 1125−1133.

(21) 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.

(22) Andersen, H. C. Rattle: a velocity version of the SHAKE algorithm for molecular dynamics calculations. *J. Comput. Phys.* **1983**, *52*, 24−34.

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

(24) Simmerling, C.; Elber, R.; Zhang, J. MOIL-View - a Program for Visualization of Structure and Dynamics of Biomolecules and STO - A Program for Computation of Stochastic Paths. In *The Proceeding of the Jerusalem Symposium on Theoretical Biochemistry*; Pullman, A., Ed.; Kulwer Academic Publishers: Netherlands, 1995; pp 241−265.

(25) Briggs, W. L. *Multigrid tutorial*; SIAM press, 1987.

(26) Chakrabar, A.; Baldwin, R. L. Stability of α-helices. *Adv. Protein Chem.* **1995**, *46*, 141−176.

(27) Anfinsen, C. B. *Science* **1973**, *181*, 223.

(28) Thompson, P. A.; Eaton, W. A.; Hoffichter, J. *Biochemistry* **1997**, *36*, 9200.

(29) Osterhout, J. J.; Baldwin, R. L.; York, E. J.; Stewart, J. M.; Dyson, J. H.; Wright, P. E. Proton NMR studies of the solution conformations of an analog of the C-peptide of Ribonuclease A. *Biochemistry* **1998**, *28*, 7059−7064.

(30) Mitchinson, C.; Baldwin, R. L. *Proteins* **1986**, *1*, 23.

(31) Guo, Z.; Thirumalai, D. *Biopolymers* **1994**, *36*, 83−102.

(32) Klimov, D. K.; Betancourt, M. R.; Thirumalai, D. *Folding Des.* **1998**, *3*, 381−496.