Weighted-Ensemble Brownian Dynamics Simulations for Protein Association Reactions

Gary A. Huber and Sangtae Kim
Department of Chemical Engineering, University of Wisconsin-Madison, Madison, Wisconsin 53706 USA

ABSTRACT A new method, weighted-ensemble Brownian dynamics, is proposed for the simulation of protein-association reactions and other events whose frequencies of outcomes are constricted by free energy barriers. The method features a weighted ensemble of trajectories in configuration space with energy levels dictating the proper correspondence between "particles" and probability. Instead of waiting a very long time for an unlikely event to occur, the probability packets are split, and small packets of probability are allowed to diffuse almost immediately into regions of configuration space that are less likely to be sampled. The method has been applied to the Northrup and Erickson (1992) model of docking-type diffusion-limited reactions and yields reaction rate constants in agreement with those obtained by direct Brownian simulation, but at a fraction of the CPU time (10^{-4} to 10^{-3}, depending on the model). Because the method is essentially a variant of standard Brownian dynamics algorithms, it is anticipated that weighted-ensemble Brownian dynamics, in conjunction with biophysical force models, can be applied to a large class of association reactions of interest to the biophysics community.

INTRODUCTION

Many biochemical reactions are characterized by events that occur infrequently on the characteristic time scale of the physical system. Among the most important are enzyme-substrate reactions and protein-protein association reactions. For a reaction event to occur, the two molecules must not only be near each other, but must also be properly oriented with respect to one another. Many of these reactions are diffusion-limited; once the molecules are properly juxtaposed, the reaction takes place very rapidly (Creighton, 1984; Northrup, 1994). Because the rate-limiting portion of the overall reaction consists of the diffusive motion of the two molecules, it is common practice to model each molecule as a rigid body and the solvent as a viscous Newtonian liquid with a dielectric constant (McCammon and Harvey, 1987; Luty et al., 1993b; Madura et al., 1994; Antosiewicz et al., 1995; Northrup et al., 1993, 1994; Kozack et al., 1995). Brownian dynamics simulations (Ermak and McCammon, 1978) can be performed on these systems without the detailed description of the atoms of the solutes or solvents. In the early 1980s an important method for extracting rate constants from such Brownian dynamics simulations was developed by Northrup et al. (1984). This method, along with other variations (Luty et al., 1992, 1993a), has been used to obtain estimates of the second-order rate constant for enzyme reactions. Still, these methods are limited by the infrequency of the reaction events; large numbers of trajectories must be generated in order to obtain enough reaction events. The recent paper by Kozack et al. (1995) represents the state-of-the-art in this regard, especially with respect to the utilization of high performance parallel supercomputers. In this paper, we develop a new simulation method, the weighted-ensemble (WE) method, which eliminates the problem of infrequency of events. The WE method, in conjunction with Brownian dynamics (i.e., WEB dynamics), is applied to a very simple energy barrier model and to a simple protein-protein association model (Northrup and Erickson, 1992), and is compared to other simulation methods.

The organization of this paper is as follows. The second section (Materials and Methods) on the difficulties associated with stochastic simulations of infrequent events provides various approaches to address these difficulties. The new method is described in the last subsection. In the next section (Results), two examples are considered to validate the new method: a simple one-dimensional (1-D) energy surface with a barrier, and the Northrup-Erickson (1992) model for protein association reactions. A quantitative estimate of the speedup (factor of 1600) exhibited by the new algorithm is provided. The paper concludes with a speculation on future directions for research, including the simulation of models that possess many more degrees of freedom (Dwyer and Bloomfield, 1993), with the help of scalable algorithms on parallel computers (efficient utilization of memory via advanced parallel programming models). An Appendix provides additional details for those wishing to apply or extend our method, as well as information on retrieval of auxiliary files (visualization with MPEG) by ftp over the Internet.

Received for publication 5 August 1995 and in final form 2 September 1995.

Address reprint requests to Dr. Sangtae Kim, Department of Chemical Engineering, University of Wisconsin, 1415 Engineering Drive, Madison, WI 53706. Tel.: 608-262-5921; Fax: 608-262-0832; E-Mail: kim@engr.wisc.edu

0006-3495/96/01/97/14 $2.00

MATERIALS AND METHODS

Theory

For the most general formulation of the problem, we consider diffusion in a d-dimensional configuration space with an arbitrary potential and (possibly varying) diffusion matrix. Suppose there is a $d - 1$ dimensional surface, or reactant surface, in this space upon which a particle, or member

of an ensemble, is created and placed according to some probability distribution. The particle begins its journey and is destroyed upon reaching a second $d-1$-dimensional surface, or product surface (Fig. 1). This raises the question: What is the mean lifetime or mean passage time of the particle? Of course, one way to answer this question is to use Brownian dynamics to simulate many trajectories all the way from the first to the second surface and to compute the average time required.

Alternatively, one may consider the following scenario. A given distribution throughout the space of many particles is present at the start. The particles all begin their journeys at the same time, and each time a particle reaches the product surface it is immediately reintroduced on the reactant surface according to the probability distribution mentioned in the previous paragraph. The question here is: What will be the flux of particles once steady state is reached? This question might be answered by maintaining many particles in the system at once, all in different stages of their journey, and counting the number that end their journey in a given time interval (Farkas, 1927). It has been shown that these two questions are mathematically equivalent; the answer to the first question is merely the inverse of the answer to the second question (Hanggi et al., 1990).

The main difficulty in answering the question by using simulations is that most interesting systems have free energy barriers, which divide the configuration space into reactant and product regions, between the two surfaces. The particles spend most of their time wandering around in the reactant region and only very rarely surmount the barrier. Thus, computing the mean time by simulating an entire particle lifetime can consume vast amounts of computer time. We refer to this method as the mean passage time (MPT) method (Northrup and McCammon, 1980). Attempting to answer the question using the second method, called the flux overpopulation method (Farkas, 1927), has its drawbacks as well; particles will arrive at the product surface too infrequently to obtain a meaningful number for the particle flux. However, this scheme can be modified to obtain meaningful results; this is the essence of the WE method.

There are two ways to imagine a large collection of particles in a configuration space. The first way is to see them as actual copies of the physical system in the real world; the second is to regard the collection of particles as an estimate of the probability distribution of states taken by one system. In other words, each particle represents a packet of probability (Gardiner, 1985). If all particles are considered to carry the same probability, then these two viewpoints are equivalent from a computational perspective; the only difference is one of philosophy. However, if each particle is allowed to carry a different probability, then adopting the second viewpoint allows additional flexibility.

For example, the probability distribution at the top of a barrier would be poorly sampled by an ensemble of particles with equal weights, because very few, if any, particles would be near the barrier top. On the other hand, if the particles are endowed with variable statistical weights, then many particles can be present at the barrier top, allowing it to be adequately sampled, but each particle will have a very small weight. If the second surface is at the barrier top, then the use of the flux overpopulation method with this weighted ensemble can yield a measurable flux, no matter how small it is.

Our next provision is to allow particles to be split, destroyed, and combined with one another. The remaining task is to devise a scheme to get those small particles to the barrier top without breaking any rules of probability theory.

The first step is to identify a reaction coordinate that describes the progress from the reactant surface to the product surface. Next, the configuration space is divided into several slabs, or bins, along the reaction coordinate, such that any point within a given slab has a reaction coordinate value within a given range associated with that slab (Fig. 2). If a value for the reaction coordinate is defined for all points in the configuration space, then the bins cover the entire space. Finally, the particles are distributed. There are several possible ways to do this; perhaps the easiest way is to give them all the same weight initially and place them according to the Boltzmann distribution by using the rejection method (Press et al., 1988). In other words, trial positions are generated uniformly in configuration space at random and accepted with probability $\exp(-V(\mathbf{x})/kT)$, where V is the potential energy. The sum of the particle weights must add up to unity so that the collection of particles represents a probability distribution.

After the ensemble is set up, the simulation proceeds. The particles are stepped forward one time step according to the Brownian dynamics algorithm (Ermak and McCammon, 1978):

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \frac{1}{kT}\mathbf{D}\cdot\frac{\partial V}{\partial \mathbf{x}}\Delta t + \sqrt{2}\,\sqrt{\mathbf{D}}\cdot\Delta\mathbf{W} + \frac{\partial}{\partial \mathbf{x}}\cdot\mathbf{D}\Delta t \tag{1}$$

The vector $\mathbf{x}$ is the position in configuration space, and $\partial/\partial\mathbf{x}$ is a vector of derivatives with respect to the different coordinates. The matrix $\mathbf{D}$ is the general diffusion matrix, which can depend on $\mathbf{x}$, and the potential of mean force is V. The time step is Δt, and $\Delta\mathbf{W}$ is a vector of independent random Gaussian numbers with a mean of ϕ and a variance of Δt.

During the time step, particles might move from one bin to another, and some might be absorbed at the product surface and re-emitted at the reactant surface. At the end of the time step, some bins have undergone a net gain of particles, whereas others have undergone a net loss, so we adjust the particle numbers in order to get back approximately the same

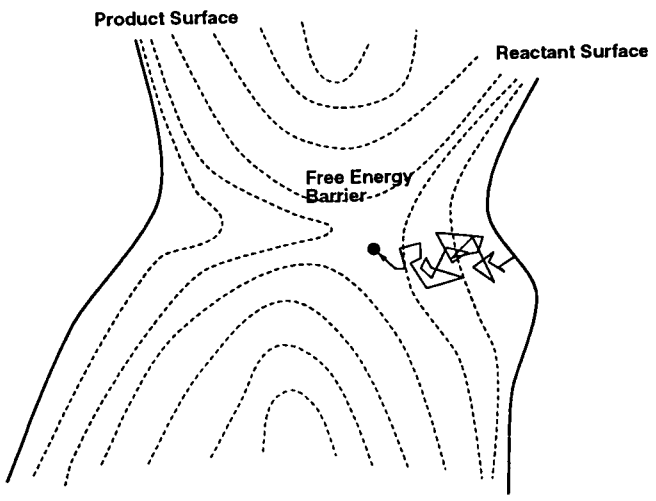

FIGURE 1 2-D configuration space. The dashed contour lines represent the potential energy; the system must diffuse through a narrow passage.

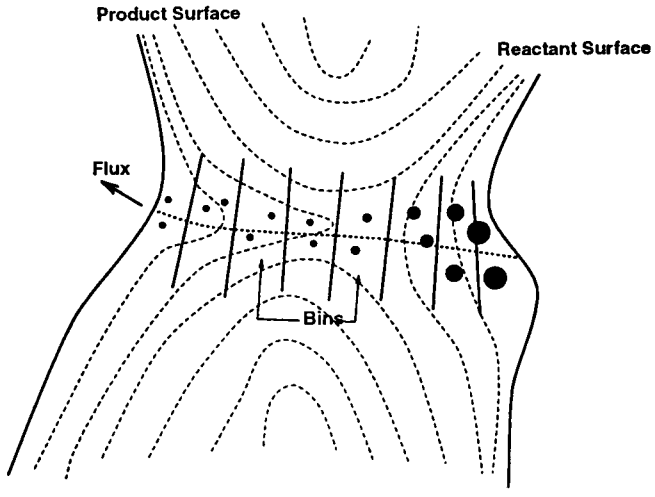

FIGURE 2 Reaction coordinate (*dotted line*) and bins for the WEB dynamics method. System copies are represented as dark circles, with statistical weight represented by size.

number of particles, n, in each bin. By doing this, we ensure that the barrier regions are sufficiently sampled and that we get measurable numbers for the probability flux. At the same time, it is desirable for the weights of particles that share a bin to have approximately the same value; this helps to reduce noise in the measured flux.

For each bin i, the total probability P_i is computed. The "ideal" particle weight is P_i/n. Each particle with weight greater than $2P_i/n$ is split into m particles, each with weight equal to $1/m$ of the parent particle. The number m is chosen so that the individual child particles have weight between P_i/n and $2P_i/n$. The child particles keep the same position as the parent, although they will go their separate ways in subsequent time steps. Probability is conserved and no new errors are introduced.

The next step in bin i is to combine some of the particles with small weights. To combine a given group of particles, the weights are added up, and the total weight is given to the resulting conglomerate particle. The conglomerate particle is then given the position of one of the original particles; the probability of a position being chosen is proportional to the statistical weight of the original particle at that position. For example, if two particles, particle a with weight 0.01 and particle b with weight 0.02, are combined, the resulting particle has a weight of 0.03. The probabilities of the particle being given the position of a or of b are 1/3 and 2/3, respectively. No statistical bias is introduced by combining particles in this manner; this is demonstrated in Appendix A.

The particles in bin i are sorted in order of statistical weight. Starting with the smallest particle, the particles are added in order of ascending weight to a conglomerate particle whose position is yet to be determined. As we move up the list, the particles are added until either 1) the next particle has a weight greater than one half the ideal weight; or 2) the combined weight is greater than the ideal weight. If the first event occurs, the current particle is still added to the conglomerate particle unless the combined weight would be >1.5 times the ideal weight; in that case, it is not included. After one of these termination events, the conglomerate particle is assigned a position. If the termination event is of the second kind, that is, the individual particles are still small, then the combination procedure is started again with the current particle. These combinations can be repeated until the individual particles become too large. The overall procedure for each bin accomplishes two goals: to keep the number of particles in each bin near a desired value n, and to keep the weights of the particles in a given bin roughly the same.

After the particle readjustments, the next time step is taken and the process is repeated. Because we can control the distribution of particles along the reaction coordinate, even at the barrier top, we can always measure some movement of probability over the barrier, even if the value is very small. An individual large particle in a well is still very unlikely to make it to the top, but a small portion of its probability is still being transported over the top. In other words, instead of waiting an extremely long time for one particle to make it over the top, we hardly wait at all and observe an extremely small fraction of a particle making the passage. If a large particle ventures into a less-probable bin, it is usually chopped up into smaller particles. Likewise, a small particle that falls down into a well is consumed by larger particles. Furthermore, if the distribution of particles should start out with equal weights and none in the upper bins, the situation is quickly adjusted. Throughout the simulation, the weight per unit time of particles that leave and are reintroduced is recorded; this is an estimate of the probability flux. Eventually, a steady state condition is reached in which the flux fluctuates around a fixed value; the inverse of this value is the MPT.

Treatment of varying time steps

During a Brownian dynamics simulation, it is often desirable to vary the size of the time step. Using large time steps where possible saves computer time. However, in regions where the generalized force $\partial V/\partial \mathbf{x}$ is varying strongly with position or where the product surface is near, small time steps are required. An important example is seen in simulations of enzyme-substrate systems, in which smaller time steps are used when the two molecules are closer together (Luty et al., 1993b; Madura et al., 1994; Antosiewicz et al., 1995; Northrup et al., 1993, 1994; Kozack et al., 1995). At first glance, this appears to present difficulties in carrying out a weighted-ensemble simulation, because the particles must be stepped forward in time together, and the overall time step size would need to be the smallest one in the system. Of course, the forces on the particles that are able to take larger steps could be updated less frequently, which would save computer time. Unfortunately, another difficulty presents itself. The regions very near the product end of the reaction coordinate are generally the ones requiring small time steps; whereas most of the remaining length of the reaction coordinate is populated by particles not requiring small steps. The time required to reach steady state scales as the square of the system's length scale, so it is these latter particles that determine the time required to reach steady state. If they could take their natural large steps, steady-state would be rapidly achieved. However, if they can only step along in time with the small time step particles, steady state will be achieved very slowly with respect to the time spent by the computer.

Fortunately, it is possible to step each particle forward by its own allowed time step size during each step of the simulation. The correct steady-state flux is achieved, steady-state is reached much more quickly, and all that is required is an easy renormalization of the particle weights at each time step. The justification for this is given in Appendix B.

Setting up the bins

Before a WEB dynamics simulation can be performed, it is necessary to choose the number of bins, where they are to be placed along the reaction coordinate, the average total number of particles and how they should be distributed among the bins, and how long the simulation should run. The main goal is to minimize the amount of computer time spent for a desired degree of precision in the final result. Implementation details such as the initial distribution of bins, dynamic adjustment of bins, and number of particles, are described in Appendix C for the benefit of readers interested in using WEB dynamics. The adaptive procedures described therein (analogous to adaptive mesh refinement in computational fluid dynamics) have a substantial impact on computational performance.

Measurement of algorithm efficiency

For most systems of interest, it is reasonable to assume that the bulk of the computer time would be spent stepping the particles forward in time; force evaluations, in particular, would dominate this step for realistic systems. Therefore, we quantify the computer time in terms of particle steps. For example, if we were to double the average number of particles but halve the simulation time, the number of particle steps taken would remain almost the same.

We also make the assumption that the square of the width of the 95% confidence interval on the quantity of interest, A, whether it is a mean time or a rate constant, is inversely proportional to the total number of particle steps taken, N_s (Press et al., 1988).

$$\eta\left(\frac{\Delta A}{A}\right)^2 = N_s^{-1} \tag{2}$$

The constant of proportionality, η, depends on the number and placement of both bins and particles; it is the measure of algorithm efficiency that is used. The quantity η^{-1} can be considered to be the total number of force evaluations or time steps that is required to obtain an order-of-magnitude estimate for A.

Statistical analysis

The most straightforward way to measure the flux is to pick a time interval that includes several time steps, measure the amount of probability that makes it across the reactant surface in that interval, and divide by the time interval to obtain an estimate of the flux during that time. If this is done repeatedly during the simulation, a record of the measured flux is accu-

mulated that can be analyzed to obtain a confidence interval of the true steady-state flux. This is a challenge because the noise in the measured flux is very non-Gaussian and is strongly correlated among successive measurements. Fortunately, it can be accomplished by using the bootstrap Monte Carlo method for data analysis (Press et al., 1988). Because we know nothing about the error structure of the flux measurements, the only estimate we have available of the distribution of measured fluxes is the set of flux measurements itself. If we have N measurements, we can generate a large number of artificial data sets (each also with N measurements) by drawing N data points at random with replacement from the original data set. We then compute the average flux in each data set, sort the values, and obtain confidence intervals for the actual flux. The assumption required to make this method valid is that the measurements extend over a time period that is several times greater than the largest time difference for which a significant autocorrelation exists.

Time required to reach steady state

For complex systems, it is difficult to obtain an *a priori* estimate of τ_r, the time required to reach steady state. Instead, we compute the autocorrelation function of the flux output and assume that τ_r is equal to the largest time difference for which a significant autocorrelation exists. This assumption is based on Onsager's regression hypothesis, which states that the relaxation of artificially imposed disturbances from steady state (such the particle distribution at the beginning of a WEB simulation) follow the same laws as spontaneous fluctuations from steady state (Chandler, 1987). The sample autocorrelations for a series of N flux values are given by

$$\rho_k = \frac{N}{N-k} \frac{\sum_{i=k+1}^{N} (J_i - \bar{J})(J_{i-k} - \bar{J})}{\sum_{i=1}^{N} (J_i - \bar{J})^2} \tag{3}$$

$$\bar{J} = \frac{1}{N} \sum_{i=1}^{N} J_i \tag{4}$$

To find out whether a measured ρ_k is statistically significant, we adopt the null hypothesis that there are no autocorrelations whatsoever and generate 1000 artificial data sets using the Monte Carlo bootstrap method as above. We then apply Eq. 4 with $k = 1$ to these data sets and sort the resulting values of ρ_1 by ascending order. If ρ_1 from the real data set is greater than the average of the 975th and 976th values from the artificial data sets or less than the average of the 25th and 26th values, then it is considered significant, because the chance of obtaining such a value purely by chance from an uncorrelated signal is less than 5%. If ρ_1 is found to be significant, then the same test is made for ρ_2 and so on, until the autocorrelations are no longer significant. The time difference represented by the value of k for the last significant ρ_k is our estimate of τ_r. Thus, to eliminate effects from the start-up, the first k values of the flux are eliminated from the computation of the median and confidence intervals mentioned above.

Interface with continuum description

In many cases, it might be possible to divide the configuration space into two regions. In the first region, it would be possible to represent the probability distribution in a region of the configuration space as a continuum rather than by discrete particles; instead of a simulation, an analytical or numerical (i.e., finite difference, finite element, etc.) solution to the governing Fokker-Planck diffusion equation would be used (Cartling, 1989). This could come about because the spatial dependence of the potential energy and diffusion matrix might be very simple in that region (Smoluchowski, 1917), or because the probability distribution function itself might have a simple structure, such as separability, allowing many coordinates to be lumped together (Grabert et al., 1980). The second region, on the other hand, would be complex enough that a simulation method, such as WEB dynamics, would be required.

If it were possible to carry out simultaneously a numerical solution in the first region with a simulation in the second, considerable savings in computer time would result versus a simulation in the entire configuration space. First of all, stepping a continuum-based numerical scheme forward in time would take much less computer time than maintaining many more copies of the system. Also, a large length scale could pose serious limitations because of the scaling of the relaxation time with the length scale. However, a continuum numerical solution could be utilized in such a manner that this problem would be mitigated. Finally, not as much memory would be required if fewer system copies could be used. An example of coupling WEB dynamics with a very simple continuum solution is given in the protein-protein association model discussed below.

Application to bimolecular systems

Formulating the general problem in terms of a probability distribution function is appropriate for studying systems composed of one molecule. However, for studying reactions involving more than one molecule, this is not appropriate; the problem must be formulated in terms of molecular distribution functions (McQuarrie, 1976). We consider in this paper a diffusion-limited, irreversible bimolecular reaction

$$A + B \rightarrow C \tag{5}$$

We assume that both A and B are very dilute, such that individual molecules of the same type do not compete for molecules of the other type. We further assume that both types are uniformly distributed throughout space, and that the distribution of molecular orientations and internal coordinates are independent of the position in space. Following the notation of Lee and Karplus (1986), we consider N_A molecules of A and N_B molecules of B. The positions of an A molecule and a B molecule in space are $\mathbf{r}_A$ and $\mathbf{r}_B$, respectively; the rotational coordinates are $\mathbf{\Omega}_A$ and $\mathbf{\Omega}_B$ and any relevant intramolecular coordinates are $\mathbf{q}_A$ and $\mathbf{q}_B$. We define $P_{Ai(Bj)}(\mathbf{r}_A, \mathbf{\Omega}_A, \mathbf{q}_A, t|\mathbf{r}_B, \mathbf{\Omega}_B, \mathbf{q}_B)$ to be the conditional probability density of the ith molecule of A having the coordinates $\mathbf{r}_A$, $\mathbf{\Omega}_A$, and $\mathbf{q}_A$ at time t, given that the jth molecule of B has coordinates $\mathbf{r}_B$, $\mathbf{\Omega}_B$, and $\mathbf{q}_B$ at that same time. We also define a conditional concentration of A:

$$\Phi_{A(Bj)} = \sum_{i=1}^{N_A} P_{Ai(Bj)} \tag{6}$$

Because of the uniformity assumption, we can make the following identification:

$$\mathbf{\Phi}_{A(Bj)}(\mathbf{r}_A, \mathbf{\Omega}_A, \mathbf{q}_A, t|\mathbf{r}_B, \mathbf{\Omega}_B, \mathbf{q}_B) = \Phi_{A(B)}(\mathcal{T}_B(\mathbf{r}_A - \mathbf{r}_B), \mathbf{\Omega}_B \circ \mathbf{\Omega}_A, \mathbf{q}_A, \mathbf{q}_B, t) \tag{7}$$

where $\mathcal{T}_B$ denotes the transformation achieved by rotating the vector using rotation $\mathbf{\Omega}_B$ and $\circ$ denotes the result of two successive rotations; $\Phi_{A(B)}$ is a more general conditional concentration. Finally, the A–B particle distribution function (PDF) is defined as

$$\rho_{AB}(\mathbf{r}, \mathbf{\Omega}, \mathbf{q}_A, \mathbf{q}_B) = \frac{\Phi_{A(B)}}{[A]} \tag{8}$$

where $[A]$ is the bulk concentration of A; $\mathbf{r}$ and $\mathbf{\Omega}$ are the relative position and orientation of an A molecule with respect to a B molecule. The PDF obeys the same diffusion equation as does the probability density of particles in the configuration space representing one A molecule and one B molecule. A product surface can still be defined as before, representing the relative placement of the molecules and the arrangements of the internal coordinates required for reaction. However, ρ_{AB} does not integrate to unity, and no reactant surface can be defined; rather, the fact that $\rho_{AB} \rightarrow 1$ as $|\mathbf{r}| \rightarrow \infty$ is used as a boundary condition. WEB dynamics can be applied with

no changes in the general method; the particles are now considered to be packets of concentration rather than probability. The second-order rate constant can be obtained by measuring the flux across the product surface (Eq. 42). As described in the example below, the simulation region is the region of configuration space in which the molecules are close, while the continuum region is the region in which the molecules are widely separated.

Energy barrier

The first system to which the WEB dynamics method is applied is a 1-D system with a force and diffusivity that are constant over the simulation domain. The system starts at $z = 0$ and ends its journey when it reaches $z = 1$; it is not allowed to go past $z = 0$ in the negative direction. The units are chosen so that the diffusivity and kT are equal to 1. The force, F, is equal to -15.76 and pushes the system toward $z = 0$. The value -15.76 is chosen to make a connection with the model of protein-protein association described below. The quantity of interest is the mean time, T_{mp}, for the system to move from 0 to 1, and the efficiency of WEB dynamics for computing this quantity is compared to that of the MPT method.

Before the WEB dynamics simulations, bins were generated by starting 100 particles at $z = 0$ and following the procedure described in Appendix C. The time step size Δt used in this stage and in the subsequent simulations was chosen to be 0.0001 time units for $z < 0.9$ and 0.00001 time units for $z > 0.9$ (Perhaps a larger time step could have been used, but it would not have changed the comparison between the two methods.) The bin generation algorithm resulted in 122 bins; repeated application of this procedure using different sequences of random numbers produced similar bin numbers. Simulations were run with 2,4,8,16, and 32 particles per bin. In addition, simulations were run for smaller numbers of bins (61, 30, 15, and 7), each with 2 particles per bin; the larger bins were generated from the 122-bin configuration by removing alternate bin partitions. Each simulation was given a "budget" of 160,000,000 particle moves. During each simulation, the flux was accumulated and printed every 0.1 time units. Significant autocorrelations were found for time differences of less than 0.3 time units, so the first three flux measurements were discarded. From the remaining flux data, 1000 artificial data sets and their average fluxes were generated using the bootstrap method and were sorted into ascending numerical order. The median was computed as the average of the 500th and 501th flux values; the lower confidence interval bound was computed as the average of the 25th and 26th values, and the upper confidence interval bound was computed as the average of the 975th and 976th values.

Protein-protein association model

The protein-protein association model is the simple spherical model previously studied by Northrup and Erickson (1992). Both protein molecules are represented by hard spheres with a radius of 18Å. For a protein-protein complex to be formed, the two spheres must not only be adjacent to each other, but must be mutually oriented properly along all three axes. To emulate the orientation requirements, each sphere has a set of four contact points mounted in a 17 × 17Å square arrangement tangent to the surface (Fig. 3). Each point on one molecule has a partner on the other molecule, and a reaction occurs when three of the four points are simultaneously within 2 Å of their partner. Northrup and Erickson also add a "docking potential" to take into account intermolecular forces, which serve to accelerate the reaction, but we omit it to demonstrate WEB dynamics' usefulness for studying very infrequent events. The molecules diffuse in water at 25°C and cannot interpenetrate, but no other forces are included.

The connection with the 1-D example is established using probability arguments. Northrup and Erickson have computed the probability of the two adjacent molecules being in a reactive configuration, given random orientations; the probability is 2×10^{-7}, which corresponds to an entropy decrease of $-kT \ln(2 \times 10^{-7}) = 15.42kT$. We also compute the entropy change associated with bringing the molecules together to a center-to-center separation of 38Å from a separation of 45Å, which is the termination

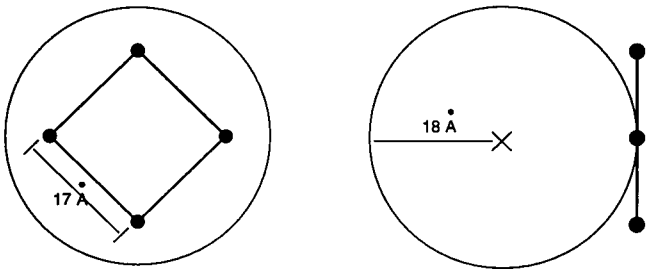

FIGURE 3 Spherical protein model, front and side views. Sites are represented by black dots.

radius of a hypothetical simulation described below under protein–protein association model. This adds another entropy decrease of $kT \ln(45/38)^2 = 0.34kT$. (This entropy decrease differs from the result because of Finkelstein and Janin (1989), because the molecules are approaching from a fixed distance in the simulation described below.) Together they add up to the free-energy increase of $15.76kT$ studied in the 1-D example.

Short-cut simulation method

In the original study, the association kinetics were studied using the Northrup-Allison-McCammon (NAM) (Northrup et al., 1984) method in conjunction with Brownian dynamics. In the NAM method, the two molecules are started with random orientations at a center-to-center distance b. Then, they are stepped forward in time using Brownian dynamics until they either react or move a larger distance q apart. Many such trajectories are generated, and the second-order rate constant is derived from the probability β of reaction versus attaining the larger separation. For this case, with no forces, the rate constant is given by

$$k = 8\pi \mathcal{D}_t b \beta_\infty \tag{9}$$

$$\beta_\infty = \frac{\beta}{1 - (1 - \beta)b/q} \tag{10}$$

where $\mathcal{D}_t = 1.36 \times 10^{-6}$ cm$^2 \cdot$ s^{-1} is the translational diffusivity of an individual sphere. Ordinarily, the starting distance b must be large enough that the forces between the molecules are centrosymmetric; in this case, however, b only needs to be large enough to avoid the reaction condition. The termination distance q must be large enough so that the steady-state PDF at that distance is independent of orientation. The quantity β_∞ is the probability that the reaction would never occur if the trajectory were not terminated at separation q; the expression for β_∞ from Eq. 10 is a way of correcting β to remove the effect of the termination radius. For this particular case, many trajectories must be run to obtain narrow confidence intervals on the rate constant, because so few of them result in reaction. Thus, Northrup and Erickson were able to report only one significant figure for this rate constant due to computer time limitations. Because we want a more precise and independent verification of the WEB dynamics, we have also modeled this system using a method similar to the NAM method. However, we take advantage of the absence of forces and the isotropy of the spheres, and use a shortcut method to generate trajectories.

To carry out the shortcut NAM simulation, we imagine the origin of the coordinate system to be located at the center of one of the molecules. The coordinate axes are given a fixed orientation in space, though, so that the center molecule can rotate around the origin. Thus, from the simulation's point of view, the other molecule appears to move with a diffusion coefficient of $2\mathcal{D}_t$, while the center of the stationary molecule appears to be fixed.

For most configurations, it is possible to combine the effect of many Brownian dynamics steps into one large step by using the procedure of Zheng and Chiew (ZC) (1988). A sphere is constructed, which is centered on the mobile molecule; we call it an R sphere. A point on the surface is

selected at random, and the center of the mobile molecule is moved to that point. The only physical limitation on the radius R of the R sphere is that no part of the stationary molecule may lie within it; other limitations are imposed by the algorithm described below. Physically, this step represents the escape of the mobile molecule from the R sphere. Because there are no forces and the molecules are isotropic, this is a legitimate method for generating trajectories, and it is much less time-consuming than generating detailed Brownian trajectories.

During the shortcut NAM algorithm, it is necessary to keep track of the physical time spent during ZC moves. During each move, the algorithm computes the physical time t_R required to move from the center to the surface of the sphere. If we define a dimensionless time $\tau = 2t_R\mathcal{D}_t/R^2$, then the probability distribution of τ is given by (Zheng & Chiew, 1988)

$$P(\tau) = 2\sum_{n=1}^{\infty}(-1)^{n-1}n^2\pi^2\exp(-n^2\pi^2\tau) \qquad (11)$$

The quantity τ is drawn from this distribution, which is computed and stored beforehand, and then t_R is computed.

Unlike the problem considered by Zheng and Chiew, the molecules also rotate while the mobile molecule is translating, so their changes in orientations during their "flying leaps" must be computed as well. It is not always necessary to compute the orientation changes after each ZC move; the orientations can be updated less frequently, as described below. For each ZC move between successive orientation updates, the escape times t_R are computed and summed to get the total time t_u between updates. When the update is needed, the angles of the rotations undergone by the molecules are drawn from a pre-computed distribution according to t_u. Any rotation can be defined in terms of an axis and angle of rotation. For this problem, it is most convenient to work in terms of the quantity $\alpha = \sin(\phi/2)$, where ϕ is the angle of rotation about the given axis (Saletan and Cromer, 1971). (This is equivalent to using Cayley-Klein parameters (Goldstein, 1980) to represent the rotation). Because of the isotropy of the molecules and lack of torques, all rotation axes are equally likely, so only α needs to be determined. It is drawn from a distribution $P(\alpha|t_u)$, which is computed beforehand by running an auxilliary simulation (see Appendix D). This distribution is peaked at $\alpha = 0$ for small values of t_u and spreads out for larger values. The axis of rotation is then selected at random, and the molecule is rotated. This step is carried out separately for the mobile molecule and the stationary molecule.

When the molecules are close to reacting, however, it is no longer possible to use the ZC algorithm; the molecules must be moved using Brownian dynamics. The mobile particle is translated according to the formula

$$\Delta x_i = \sqrt{4\mathcal{D}_t}\,\Delta W_i \qquad (12)$$

where x_i is the change in position along the i-axis ($i = x, y, z$). Both particles are rotated according to the formula

$$\Delta\phi_i = \sqrt{2\mathcal{D}_r}\,\Delta W_i \qquad (13)$$

where $\Delta\phi_i$ is a small rotation about the i-axis ($i = x, y, z$) and $\mathcal{D}_r = 3.16 \times 10^7$ sec^{-1} is the rotational diffusivity. The small rotations are accumulated using rotation matrices (Nambi et al., 1991). The time step size is varied according to how close the molecules are to reacting. The maximum time step size is 2.38 ps (this is equal to $0.0001 \times (18\text{Å})^2/\mathcal{D}_t$) and the minimum time step size is 0.0238 ps. In a configuration, there are four distances between the paired sites; we define ζ_i to be the i^{th}-smallest such distance. A trial size for the time step, in units of picoseconds, is computed as

$$\Delta t(\text{ps}) = 0.0136(\zeta_1(\text{Å}) - 2.0)^2 \qquad (14)$$

unless ζ_1 is less than 2 Å, in which case Δt is given its minimum value. If this time step size goes over the maximum value or under the minimum value, then it is given the maximum or minimum value, respectively. The reaction criterion, of course, is that ζ_3 be less than 2 Å.

The radius b is chosen to have the smallest possible value, 38Å, and the outer radius q is given the value 360Å. The trajectory of the mobile molecule is started at a random spot on the b surface centered at the stationary molecule, and both molecules are given random orientations at the outset. This random orientation is computed first by randomly choosing an axis of rotation, all axes being equally likely. Next, the rotation angle ϕ is chosen from the probability distribution

$$\frac{1}{2\pi}(1 - \cos(\phi)) \qquad (15)$$

with ϕ ranging from $-\pi$ to π, using the rejection method (Press et al., 1988). This is the proper procedure for generating an equilibrium distribution of orientations in the absence of forces (Hamermesh, 1989). We also define two other radii near the b-radius: b_1 which is slightly larger than b, and b_2, which is slightly larger than b_1. For this example we use $b_1 = 38.72$Å and $b_2 = 39.44$Å.

For the shortcut NAM method, we consider three different simulation regimes. When the center of the mobile molecule lies outside the b_2 surface, the ZC algorithm is used. The R sphere is constructed to be tangent to the b surface but not containing the stationary molecule. This move is repeated until the mobile molecule either lands inside the b_1 surface or outside the q surface. If the mobile molecule lands outside the q surface, then the trajectory is terminated with a probability $1 - b/r$, where r is the distance between the molecule centers. This is the probability that the mobile particle will never return to the b surface (Northrup et al., 1984). Otherwise, it is restarted on the b surface, again with random orientations for both molecules (Fig. 4). This restarting of the trajectory is very similar in spirit to the method of Luty et al. (1992). Because the mobile molecule does not go inside the b-surface at this stage, no reactions can occur.

If the mobile molecule lands inside the b_1 surface, then the rotations are updated and ζ_3 is computed. If ζ_3 is less than a threshold value ζ_t (see Appendix D), then the ZC moves are continued, except that the R sphere is chosen to be tangent either to the sphere of excluded volume (radius of 36Å) or to the b_2 sphere, whichever one is closer. If the mobile molecule lands outside the b_1 radius, then the system is again in the first simulation regime and the ZC moves are performed according to the previous paragraph. Allowing the molecules to collide creates problems with the ZC method because the steps become smaller and smaller as they approach. To overcome this, when the molecules are very close, the R sphere is chosen to be tangent to the b_2 surface, and the center of the mobile molecule is placed at random on the resulting approximate hemispherical surface, computing the escape time as usual. This helps to "kick" the molecules away from each other so that they do not become "stuck." As with the first regime, we assume that no reaction can take place while the system is in this second regime. Because ζ_3 is so large, the chance of a reaction occurring before the mobile molecule moves beyond the b_1 surface is very small.

If the mobile molecule lands inside the b_1 surface as before, but the value of ζ_3 is now less than the threshold value ζ_t then the molecules are moved using Brownian dynamics. The reaction criterion is tested at each step. This is continued until either a reaction occurs or the mobile molecule escapes beyond the b_1 surface.

The probability of reaction versus escape is computed for many trajectories. For this variation on the NAM method, the probability is β_∞ in Eq. 9. From Eq. 9 the second-order rate constant k is computed, and the width of the 95% confidence interval is computed using the properties of the binomial distribution from

$$\Delta k = 32\pi\mathcal{D}_t b\sqrt{\frac{\beta_\infty}{N_t}} \qquad (16)$$

where N_t is the number of trajectories generated.

Weighted-ensemble method

To carry out the WEB dynamics simulation on the protein-protein association model, it is necessary to define two regions of configuration space: an

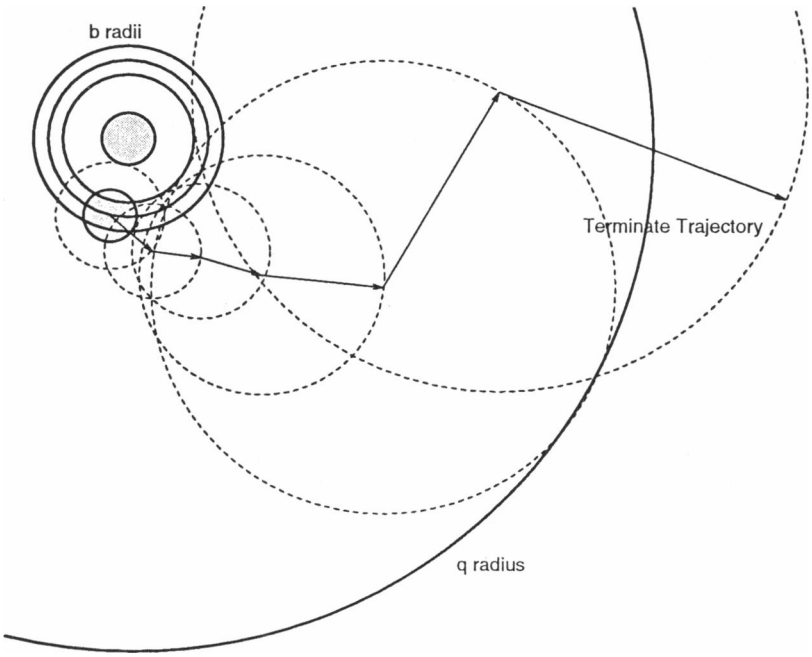

FIGURE 4 ZC procedure. Center of one molecule is started on *b* surface; dotted circles represent spheres from which molecule escapes, and arrows show overall trajectory to termination.

inner region in which simulations are performed, and an outer region in which a continuum solution is carried out. The inner region is defined by configurations in which the centers of the squares of sites on both molecules, or site centers, are separated by a distance of <9Å. The outer region is the rest of the configuration space. In addition, a subregion of the inner region, the interface region, defined by configurations with the site centers separated by a distance between 8.1 and 9Å, is included for matching the simulation and the continuum solution. The same rule (see Eq. 14) is used for choosing the size of the time step, and the molecules are moved according to the same Brownian dynamics algorithm. The continuum solution adopted here is very simple; we assume that the steady-state PDF in the interface and outer regions is identical to the equilibrium PDF in the absence of reaction. This is a good approximation, because the equilibrium PDF is barely perturbed by the very infrequent reaction events.

Before setting up bins and performing the simulation, it is necessary to determine the equilibrium population $\mathcal{N}$ of the inner region, defined by

$$\mathcal{N} = \int_V \rho_{AB}(\mathbf{r}, \mathbf{\Omega}) d\mathbf{r}\, d\mathbf{\Omega} \tag{17}$$

where the integration extends over the inner region. We call this quantity a "population," for want of a better word, because it is not really a probability or indicative of the physical concentration in the test tube. If the two molecules could interpenetrate, the population would be numerically equal to the volume of a 9Å-radius sphere, which describes the allowable distances between the site centers. However, the molecules cannot interpenetrate, so the actual accessible volume of configuration space and the population are much smaller. The fraction of accessible volume is computed by generating many random configurations with the site centers <9Å apart and counting those in which the molecules do not collide. The configurations are generated by placing the site center of one molecule at the origin, placing the site center of the other molecule randomly inside a 9Å sphere centered at the origin, and giving random orientations about the site centers to both molecules. A collision is determined by the center-to-center distance; a distance <36Å constitutes a collision. The population is then computed as the product of the sphere volume and the fraction of noncolliding configurations. The equilibrium population of the interface region is determined by counting the acceptable configurations that land in the interface region. We generated 100,000 configurations; the fraction of acceptable configurations was 0.1002, and the fraction of configurations that were both acceptable and within the interface region was 0.0339.

To generate the bins, the systems are started on the very edge of the inner region. The site center of one molecule is placed exactly 9Å from the site center of the other molecule; both molecules are given random orientations; and the acceptable configurations, with no collisions, are kept. The reaction coordinate used throughout the WEB dynamics procedure is $\zeta = \zeta_3$ as defined above. The bin-building procedure (see Appendix C) was performed with 100 system copies; the resulting number of bins was 44.

Because the continuum solution is so simple, the procedure for matching the simulation and continuum regions is straightforward. At the end of each time step during the simulation, the systems that are in the interface region are deleted. Then, the same number of systems, plus one, are created in the interface region and distributed at random, by generating configurations and keeping the acceptable ones as described above. If there are no systems in the interface region at the end of the time step, then one system is generated. The value of the equilibrium population of the interface region is evenly divided among the weights of the new systems. We need to pay attention to the interface region only; the outer region is ignored in the simulation, and any move taking the system into the outer region is rejected. Assuming a constant value for the continuum solution greatly reduces the time required to reach steady-state as well as reducing the required number of system copies.

The WEB dynamics simulation was performed with 2, 4, 8, 16, and 32 systems per bin, for an average of 88, 176, 352, 704, and 1408 system copies, respectively. The systems were evenly distributed with equal weights inside the inner region, and their weights added up to the equilibrium population of the inner region. The simulation for the first two cases was run for a total of 1,000,000 time steps/system, or 2,380 ns based on the largest time step possible. The simulation for the next two cases was run for half that time, or 500,000 time steps/system, and the last case was run for 250,000 time steps/system. For all simulations, the flux was accumulated and output every 1000 time steps. The bootstrap Monte Carlo method was used to analyze the flux data; we used the same procedure as was used for the 1-D model. For all cases, the autocorrelation was significant only to $k = 2$, (corresponding to $\tau_r = 4.76$/ps), so the first two flux values were excluded from the analysis.

RESULTS

Energy barrier

The computed MPTs and their 95% confidence intervals for the 1-D case are shown in Fig. 5. It can be seen that the confidence intervals become more narrow as the number of system copies increases. The analytical solution is shown as a straight line for comparison. The mean times from the WEB dynamics method tend to be a bit higher than the analytical solution, probably because of the finite size of the time step near the barrier top (Lamm and Schulten, 1982).

For this simple system, it is possible to calculate an analytical solution not only for T, but also for the efficiency of the MPT method. Thus, it is not necessary to actually run a MPT simulation for the purposes of this study. To obtain the solution for $T(0)$, we note that the function $T(z)$, which is the mean time to go from z to 1, is the solution of the differential equation and boundary conditions (Gardiner, 1985)

$$\frac{d^2T}{dz^2} + F\frac{dT}{dz} = -1 \tag{18}$$

$$T(1) = 0 \tag{19}$$

$$\frac{dT}{dz}(0) = 0 \tag{20}$$

The solution to Eq. 20 is

$$T(z) = \frac{e^{-F}}{F^2} + \frac{1}{F} - \frac{e^{-Fz}}{F^2} - \frac{z}{F} \tag{21}$$

For such a large negative value of F, this gives approximately

$$T_{mp} \equiv T(0) = \frac{e^{-F}}{F^2} = 28,100 \tag{22}$$

To evaluate the efficiency, it is necessary to compute the second moment, $T_2(z)$, of passage times from z to 1. It is the solution of the equation and boundary conditions (Gardiner, 1985)

$$\frac{\partial^2 T_2}{\partial z^2} + F\frac{\partial T_2}{\partial z} = -2T(z) \tag{23}$$

$$T_2(1) = 0 \tag{24}$$

$$\frac{\partial T_2}{\partial z}(0) = 0 \tag{25}$$

For large negative F, solving this gives approximately

$$T_2(0) = \frac{2e^{-2F}}{F^4} \tag{26}$$

The variance about the mean is given by

$$V_T = \sqrt{T_2(0) - T^2(0)} = T_{mp} \tag{27}$$

If the MPT method is carried out and a large number, N_t, of trajectories are generated from beginning to end, then the width of the 95% confidence interval is

$$\Delta T = \frac{4V_T}{\sqrt{N_t}} \tag{28}$$

If the average time step size is Δt, then the total number of time steps is $NT_{mp}/\Delta t$; using Eq. 2 gives an efficiency of

$$\eta = \frac{\Delta t}{16T_{mp}} \tag{29}$$

It can be seen that if larger time steps can be taken, greater efficiency results. For this problem, assuming that almost all of the time steps taken have the larger value of 0.0001 time units, the efficiency of the MPT method would be 2.22×10^{-10}.

The efficiencies of the WEB dynamics method using the different numbers of system copies are shown in Fig. 6, and the efficiency of the MPT method is shown at the spot on the far left where the number of system copies would be 1. The efficiency increases with the number of systems, but levels off and actually starts to decrease at larger numbers. The efficiency of the WEB dynamics method for large numbers is about 14,000 times greater than that of the MPT

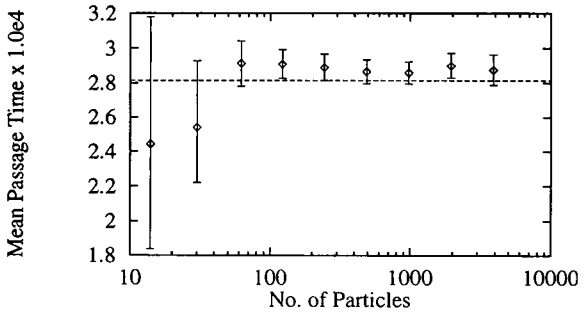

FIGURE 5 MPT computed by WEB dynamics using different numbers of systems; same number of time steps are taken for each data point. Dotted line is analytical solution.

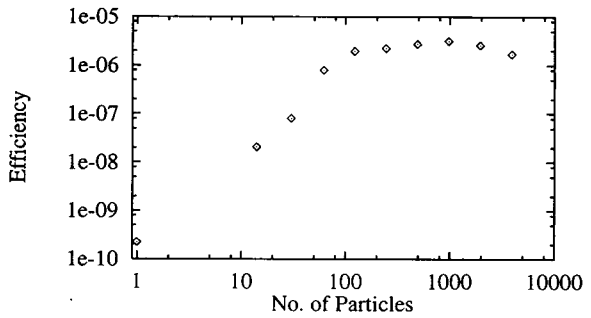

FIGURE 6 Efficiency of WEB dynamics for different numbers of systems (1-D system). Leftmost data point is efficiency of MPT approach.

method. In other words, to obtain comparable results, it would be necessary to use 14,000 times more computer time with the MPT method than would be required for WEB dynamics. The leveling-off effect is not well understood, but probably arises from the trade-off between the reduction in noise and the greater time required to move more system copies.

Protein-protein association model

The shortcut NAM method was run with 10,000,000 trajectories, of which 353 led to reaction. This gave a reaction rate of $2.72 \times 10^5 \pm 0.29 \times 10^5\ M^{-1}\ s^{-1}$. The results from the WEB dynamics simulations are given in Fig. 7. The results from the WEB dynamics are consistent with the result from the shortcut NAM method given the uncertainties involved, although the medians of the WEB results tend to be somewhat lower. This could be due to the approximations in the shortcut NAM method. All of our results are over a factor of 2 greater than the results reported by Northrup and Erickson (1992). Because they report only one significant figure and do not report the size of the time step used, it is difficult to compare our results with theirs. The reaction rate obtained for any of these methods is quite sensitive to the size of the time step; if a time step is too large, then reactive regions of the configuration space can be skipped inadvertently, resulting in a lower measured rate. We used the same time step sizes for both methods to ensure consistency. We should point out that the purpose of the Northrup-Erickson study was to obtain order-of-magnitude estimates for protein association reactions, not to obtain accurate rate constants for this particular model.

We do not compute the efficiency of the short-cut NAM method, because it is not general enough to be useful for real problems. It is possible to compute the efficiency if the simulation is run instead as an ordinary NAM method, with all Brownian dynamics steps taken. However, in the shortcut method, we used a much larger termination radius than a user of the ordinary NAM method would have used. For the shortcut method, a large termination radius is used because it barely increases the computer time required for the simulation but results in higher reaction probabilities.

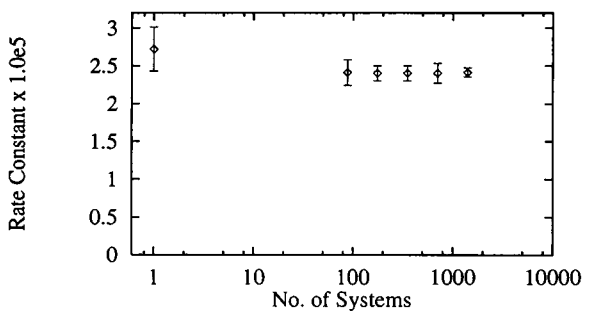

FIGURE 7 Rate-constants ($M^{-1}\ s^{-1}$) from WEB dynamics simulation on protein-protein association model. Results from the shortcut NAM method are shown at the far left, corresponding to one system.

For the ordinary NAM method, such a large radius would increase the simulation time enormously and result in a very small efficiency. Now, in the WEB dynamics runs, we assume that the reactant PDF is not disturbed by the reaction once the site centers are 9Å apart; this corresponds approximately to the condition that the molecule centers be 45Å apart. Therefore, we compare the efficiency of the WEB with the efficiency that the NAM method would have if the termination radius q were set at 45Å instead of 360Å.

We first find the value of β in Eq. 10 that would result. Since b and q are given and β_∞ has been computed already, Eq. 10 can be solved for β, resulting in a value of 3.49×10^{-6}. The next step is to find the average time of a trajectory. Because a reaction event is so unlikely, this is essentially an MPT problem in spherical coordinates, with a starting position of b, an absorbing boundary at q, and an impermeable boundary at $\alpha = 36$Å, the radius of excluded volume. The mean time for escape, $T_{mp} \equiv T(b)$ is computed by solving the equation (Agmon, 1984)

$$\frac{1}{r^2}\frac{d}{dr}\left[r^2\frac{dT}{dr}\right] = -2\mathcal{D}_t^{-1} \tag{30}$$

$$T(q) = 0 \tag{31}$$

$$\frac{dT}{dr}(a) = 0 \tag{32}$$

This gives

$$T_{mp} = T(b) = \frac{1}{12}\mathcal{D}_t^{-1}[q^2 - b^2] + \mathcal{D}_t^{-1}\frac{a^3}{6}\left[\frac{1}{q} - \frac{1}{b}\right] = 1.21\ \text{ns} \tag{33}$$

Because it is β that is directly computed by the ordinary NAM method, the confidence interval width is computed by

$$\Delta k = \frac{32\pi\mathcal{D}_t b}{1 - b/q}\sqrt{\frac{\beta}{N_t}} \tag{34}$$

Furthermore, we assume that almost all of the time steps taken are with the maximum size 2.38 ps described above. The mean escape time is used to compute the number of time steps, and the expressions from Eqs. 9, 10, and 34 are plugged into Eq. 2 to give the following expression for the efficiency:

$$\eta = \left[\left(\frac{4\sqrt{\beta/N_t}}{\beta}\right)^2\left(\frac{N_t T_{mp}}{\Delta t}\right)\right]^{-1} = \frac{\Delta t\beta}{16T_{mp}} = 6.69 \times 10^{-10} \tag{35}$$

The efficiencies of the WEB dynamics method and the hypothetical NAM method are shown in Fig. 8. For this particular problem, the best case of the WEB dynamics method is ~1600 times more efficient than the NAM method.

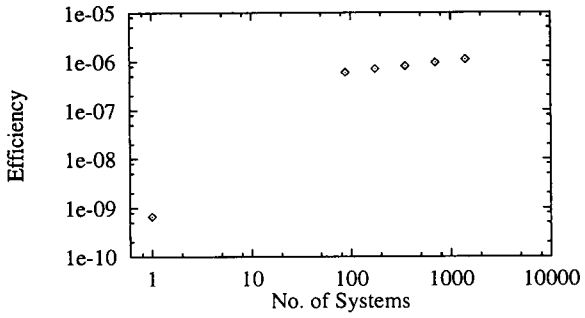

FIGURE 8 Efficiencies of the WEB dynamics method and the NAM method. The efficiency of the NAM method is shown at the far left, corresponding to one system.

DISCUSSION

The WE method, used in conjunction with Brownian dynamics, is an efficient simulation method for computing rate constants of infrequent events. It has been applied to a simple 1-D model of an energy barrier and was found to be 14,000 times more efficient than the MPT method. It has also been applied successfully to a simple model of protein-protein association and was found to be 1600 times more efficient than the NAM method. Indeed, it appears that the efficiency plateau (Fig. 8) for the bimolecular model has not been reached. Also, very high efficiencies relative to the NAM method are obtained even for more modest numbers of system copies.

These two problems have free-energy barriers of approximately the same height. The barrier of the 1-D problem is an energy barrier, whereas the barrier of the protein-protein association model is an entropy barrier in several dimensions. However, the efficiencies are of the same order of magnitude, so it does not appear that the increase in dimensionality by the bimolecular problem over the 1-D problem seriously hinders the efficiency of the WEB dynamics method.

Of course, the large increase in efficiency for this model is due to the very small chance of a reactive encounter; for reactions that proceed at a faster rate (e.g., if we had included the docking potential of Northrup and Erickson), the difference in efficiency probably would not be as great. On the other hand, the utility of WEB dynamics would be even greater for reactions with repulsive energy barriers.

Future work will include a more sophisticated numerical treatment of the continuum region, which will allow the simulation region to be made even smaller and will result in higher efficiencies. We will be able to explore the role of hydrodynamic interactions, especially in the cleft regions as suggested by Brune and Kim (1994), despite the cost incurred by solving boundary integral equations (Kim and Karrila, 1991) at each Brownian time step. The general nature of the WE method will also allow for the efficient linkage of Brownian dynamics simulations of intermolecular encounters with more detailed molecular dynamics simulations required for the reaction event itself (Luty et al., 1993a).

We plan to disseminate the WEB dynamics method via the Internet, by installing an example WEB dynamics (C) program on our ftpserver ftp.engr.wisc.edu (in the directory pub/stkim/microhydro/WEBD) as soon as a "user's guide" is completed. A computer-aided visualization of the WEB dynamics method has already been installed on the same ftpserver, as described in Appendix E.

Computer memory is the limiting resource in the WEB dynamics algorithm, especially when we go to problems with many degrees of freedom, e.g., multiple reacting species and more elaborate models of proteins that possess internal degrees of freedom. In the current computational landscape, parallel computers offer one pathway to large amounts of memory (on the order of gigabytes on current machines and terabytes on machines of the next generation), albeit distributed over many processing nodes. This raises an intriguing avenue for future work. WEB dynamics map naturally onto the "message passing" programming model for parallel computing. We first assign bins to different processors (typically, we will have more bins than processors, so each processor will handle multiple bins). We then perform Brownian dynamics simulations within each processor; particles crossing boundaries between bins residing on different processors activate interprocessor messages. The challenge is to apply WEB dynamics to a large simulation arising from realistic protein models. In the computer architecture community, there is great interest in designing cache-coherent, shared-memory computers in which the physical memory is actually distributed over many processors, e.g., (Hill et al., 1993, Lenoski et al., 1992). Shared-memory programming models offer a more supportive environment for parallel algorithm development. We believe that the WEB dynamics algorithm provides an ideal test for such advanced programming models, and we plan to validate this hypothesis by comparing message passing and shared memory versions of the algorithm, along the lines of Traenkle et al., (1995).

APPENDIX A

Removal of particles

Suppose we have a collection of N particles at positions $\mathbf{x}_i$ and weights p_i where $i = 1$ to N. If we consider $t = 0$ to be the present, then this sums up our present knowledge of the system; namely, that the probability of it actually being at $\mathbf{x}_1$ is p_1, the probability of it being at $\mathbf{x}_2$ is p_2, and so on. The actual state of the system, unknown to us, is denoted by i^*.

Next, we suppose that there exists an agent who knows what i^* is and to whom we can direct questions. We decide to ask the following question about states 1 and 2: Which of these two states can we definitely eliminate from consideration? If $i^* = 2$, then the agent must answer 1, and if $i^* = 1$, then the answer must be 2. If neither state is the actual state of the system, then the agent is free to give either answer. We assume, in this case, that the agent tells us to eliminate state 1 with probability $p_2/(p_1 + p_2)$ or to eliminate state 2 with probability $p_1/(p_1 + p_2)$. So, from the agent's viewpoint there are four cases: 1) state 1 is eliminated because the system actually resides in state 2; this occurs with probability p_2; 2) state 2 is eliminated because the system actually resides in state 1; this occurs with probability p_1; 3) the system resides in neither state 1 or 2, and state 1 is chosen to be eliminated; this occurs with probability $(1 - p_1 - p_2)p_2/(p_1$

$+ p_2$); 4) same as in 3), but state 2 is chosen to be eliminated; this occurs with probability $(1 - p_1 - p_2)p_1/(p_1 + p_2)$. It can be seen that the overall probability of state 1 being eliminated is equal to $p_2/(p_1 + p_2)$, and the probability of 2 being eliminated is $p_1/(p_1 + p_2)$.

We now define several probabilities:

- $p(i^* = i)$—*a priori* probability that the system actually resides in state i; equal to p_i
- $p(\dagger j)$—*a priori* probability that system j will be eliminated
- $p(i^* = i|\dagger j)$—probability that the system actually resides in state i, given that system j is the one eliminated
- $p(\dagger j|i^* = i)$—probability that state j will be eliminated, given that the system actually resides in state i. We already know that $p(\dagger 1|i^* = 1) = 0$, $p(\dagger 1|i^* = 2) = 1$, and $p(|1|i^* = k) = p_2/(p_1 + p_2)$, where $k > 2$.

Finally, suppose that we have made our inquiry and the agent tells us to eliminate state 1. Now that we have additional information, we must reassign weights to the states, or particles, because our knowledge of the probabilities has changed. This updating of the information is done by means of Bayes' theorem:

$$p(i^* = i|\dagger 1) = p(\dagger 1|i^* = i)p(i^* = i)/p(\dagger 1) \quad (36)$$

Plugging in the probabilities on the right-hand side, we see that

$$p(i^* = 1|\dagger 1) = 0 \quad (37)$$

$$p(i^* = 2|\dagger 1) = p_1 + p_2 \quad (38)$$

$$p(i^* = k|\dagger 1) = p_k \quad (k > 2) \quad (39)$$

This is exactly analogous to the procedure above; we decide to eliminate particle 1 and transfer its weight to particle 2, while keeping the weights of the other particles unchanged. (If the agent had told us to eliminate state 2, we could have performed the same analysis, with a symmetrical result.) If we so desire, we can now ask the same question about states 2 and 3; it can be seen that an inquiry about 1 and 2 followed by an inquiry about the survivor (1 or 2) and 3 is equivalent to one inquiry about 1, 2, and 3 together. By inductive reasoning, we can see that any number of particles can be combined in this fashion without violating any rules of probability. So, when we eliminate particles, we do not destroy any information. Instead, we force the algorithm to make a more firm commitment about the actual state of the system.

APPENDIX B

Use of variable time steps

To justify using different time steps for different systems, we note that the distribution $P(\mathbf{x}, t)$ of a collection of particles that move according to Eq. 1, all using the same Δt, is given by (Gardiner, 1985)

$$\frac{\partial P}{\partial t} = -\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{J} \quad (40)$$

$$\mathbf{J} = -\mathbf{D} \cdot \left[\frac{1}{kT}\frac{\partial V}{\partial \mathbf{x}} P + \frac{\partial P}{\partial \mathbf{x}}\right] \quad (41)$$

The vector $\mathbf{J}$ is the flux of probability. Let us also denote by $S(\mathbf{x})$ the probability distribution, defined on the reactant surface, of the placement of newly generated particles. The reaction rate constant is given formally by

$$k = \int_{SP} \mathbf{J}_{ss} \cdot \mathbf{n}\, dS \quad (42)$$

This is a surface integral over the product surface. The steady-state flux, $\mathbf{J}_{ss}$, is obtained formally by setting the time derivative in Eq. 41 to 0 and solving the equation, subject to two boundary conditions and the normalization condition: that $P = 0$ on the product surface, that $\mathbf{J}(\mathbf{x}) = kS(\mathbf{x})\mathbf{n}$ on the reactant surface, and that $\int P(\mathbf{x})d\mathbf{x} = 1$. On both surfaces, the unit surface normals are denoted by $\mathbf{n}$.

Now, suppose we have a collection of particles, each of which moves according to its own time step. Let Δt_{max} be the largest allowed time step in the ensemble, and let $\alpha(\mathbf{x})\Delta t_{max}$ be the time step size as a function of position. The scale factor α is between 0 and 1. The particles are stepped forward in time according to the formula

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{1}{kT}\mathbf{D} \cdot \frac{\partial V}{\partial \mathbf{x}} \alpha \Delta t_{max} + \sqrt{2\alpha}\, \sqrt{\mathbf{D}} \cdot \Delta \mathbf{W}_{max} + \frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{D}\alpha \Delta t_{max} \quad (43)$$

where $\Delta \mathbf{W}_{max}$ is evaluated using the maximum time step. The steady-state diffusion equation for the resulting probability distribution $\hat{P}(\mathbf{x}, t)$ is

$$\frac{\partial \hat{P}}{\partial t} = -\frac{\partial}{\partial \mathbf{x}} \cdot \hat{\mathbf{J}} = 0 \quad (44)$$

$$\hat{\mathbf{J}} = \frac{1}{kT}\left[\mathbf{D} \cdot \frac{\partial V}{\partial \mathbf{x}}\right]\alpha\hat{P} + \left[\frac{\partial}{\partial \mathbf{x}} \cdot \mathbf{D}\right]\alpha\hat{P} + \frac{\partial}{\partial \mathbf{x}} \cdot [\mathbf{D}\alpha\hat{P}] \quad (45)$$

where the third and fourth terms in Eq. 45 are divergences of tensors. The same boundary conditions apply for $\hat{\mathbf{J}}$, but we have not yet specified a normalization condition. The next question to be answered is that of the relation of $\hat{P}$ to P in Eq. 41. Intuitively, we would expect particles to "pile up" in regions where the time step is smaller. Therefore, we guess that

$$\hat{P} = P/\alpha \quad (46)$$

Plugging this into Eq. 45 and comparing it to Eq. 41 shows that $\hat{\mathbf{J}} = \mathbf{J}$, and Eq. 45, along with both boundary conditions, is satisfied. Furthermore, the value of k is the same for both cases, so the measured reactive flux should be correct. The normalization condition is slightly different:

$$\int \alpha\hat{P}\, d\mathbf{x} = 1 \quad (47)$$

At each step of such a simulation, the particle weights should be rescaled by the same factor to maintain this normalization condition. All subsequent WEB dynamics simulations follow this procedure of moving each particle according to its own time step.

APPENDIX C

Bin placement

Perhaps the most crucial implementation decision is the total number of system copies to be used. Using more system copies should increase the degree of sampling and reduce noise in the flux, but would take more time. In addition, for systems with many degrees of freedom, memory limitations would be an issue. The next decision is that of bin placement. The bins should be small enough so that they effectively bridge regions of differing probability densities; where the density is decreasing most rapidly, we would expect to construct the most bins. These two decisions are coupled by the desired number of systems per bin. If there are fewer systems per bin, this allows more bins and hence better bridging of different regions, but then the weights of the particles might fluctuate more. This dilemma is resolved by an adaptive bin scheme described below. The final decision is the length of the overall simulation, which depends on the desired precision, and the desired start-up time to ensure attainment of steady state.

We have settled upon the following procedure for bin placement. The first bin is constructed at the reactant end of the reaction coordinate, with one boundary at the reactant surface. A large number, N, of equally weighted systems (perhaps the number to be used in the WEB dynamics simulation itself) are placed on the boundary. They are all stepped forward in time by one time step, and sorted in order of reaction coordinate value. Then, we compute the value of the reaction coordinate, which evenly divides the set of systems, $0.5 * (\zeta_{N/2} + \zeta_{N/2+1})$, where ζ_i is the reaction coordinate of the ith systems; this becomes the upper boundary of the first bin and the lower boundary of the second bin. (We use "upper" and "lower" to mean toward and away from the free energy barrier, respectively.) The $N/2$ systems that are still below the new boundary are deleted, and the other $N/2$ particles are each split into two particles, getting back N systems. This set of steps is repeated, with the new dividing value becoming the upper boundary of the second bin, and so on (Fig. 9).

Eventually the collection of systems approaches the product surface, and the procedure is terminated when more than half of the systems cross the product surface during a time step. The product surface becomes the upper boundary of the final bin. While the systems are climbing a steep slope of a barrier, they can slide back downhill and the dividing value can be actually lower than the previously computed boundary. During the procedure, this is temporarily ignored, because the systems go uphill in the long run. After the bin boundaries are constructed, however, the boundary values are sorted by value to avoid "inside-out" bins. This results in a denser collection of bins where the systems struggle uphill. The overall procedure spaces the bins in such a way that a system has, very roughly, a 50% chance of moving uphill to the next bin during a time step. This ensures a steady flow of probability up to the product surface. We note that bins cluster where the probability decrease along the reaction coordinate is the greatest. Given the time step size, the number of bins created is a fixed number; but if fewer bins are desired, then bins can be merged by eliminating their common boundaries.

We have adopted the idea of allowing as many bins as possible, with a small number of particles in each bin. However, where this results in the probability being too "clumpy," the bins are temporarily merged to smooth out the particle weights. For many systems of interest, it can be expected that the steady-state probability profile, in real life, is monotonically decreasing as one approaches the produce surface. This does not hold at each time step in the WEB dynamics simulation; indeed, several bins might be empty at any one time. These empty bins, or bins that contain much less probability than they would on average, should be merged with other, occupied bins during that particular time step. Before the particles are rearranged, we start at the top bin (nearest the product surface) and call this the center bin of the first merged bin. Then, the probability in the next-highest bin is computed; if its contained probability is greater than that in the top bin, then this bin becomes the center of the next merged bin; the top bin forms a bin by itself. Otherwise, if the next-highest bin has a smaller probability, due to a fluctuation, then we keep going to lower bins until we reach one containing a greater probability than the top one; this one becomes the center bin of the second merged bin. The bins lying in between that were passed over due to their small populations are then evenly divided between the first and second merged bins; in the case of an odd number of bins, the extra one goes to the lower merged bin. This process is repeated, starting at the center of the second merged bin; the center of the third merged bin is the first bin to contain a greater probability than the center of the second merged bin. If it should happen that the bin probabilities are monotonically decreasing uphill at a particular time step, then each bin is the center of a merged bin comprised only of its center bin. The ideal number of systems contained by a merged bin is simply the sum of the ideal system numbers of its component bins. The merged bins are then treated like normal bins in carrying out the system rearrangements described above. This procedure has the advantage in smoothing out fluctuations in probability, while allowing probability to flow uphill easily in sections where monotonicity is temporarily achieved (Fig. 10).

APPENDIX D

Auxilliary simulations for shortcut NAM method

Creating rotation distribution

The auxillary simulation was run by starting 10,000 spheres in the same orientation and rotating them simultaneously according to the Brownian dynamics algorithm in Eq. 13, using the maximum time step of 2.38 ps described above. At various times the values of α were computed for each sphere and sorted in ascending order. The lower bound of α was taken to be 0, and the upper bound was taken to be the 99,900th value (the top 1% of the values are treated as outliers). This interval was divided into 100 evenly spaced subintervals, and the values of α were distributed into these intervals to build up a binned representation of the probability distribution at that particular value of t_u. This distribution of α was generated and stored

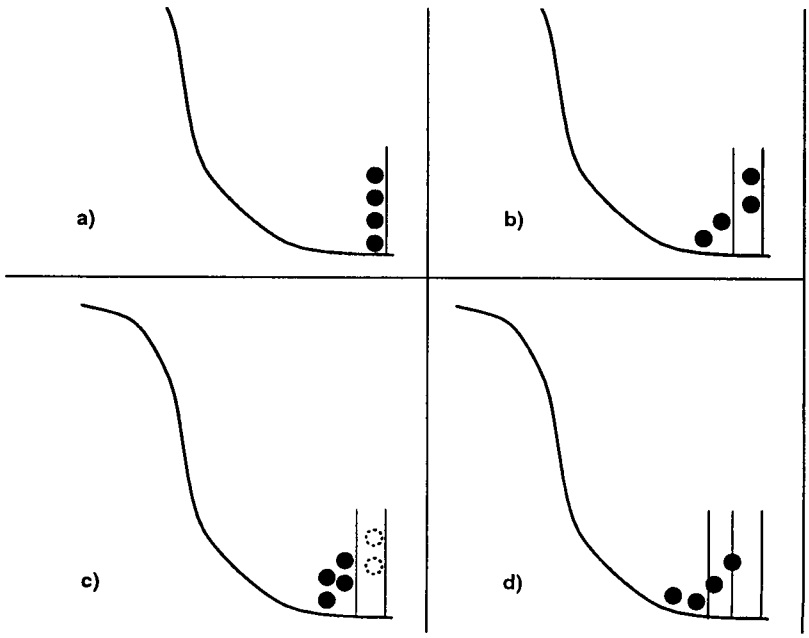

FIGURE 9 Steps in building a bin: *a*) systems start out on reactant surface; *b*) systems are stepped forward; *c*) systems are partitioned: surviving systems are duplicated; *d*) procedure is repeated.

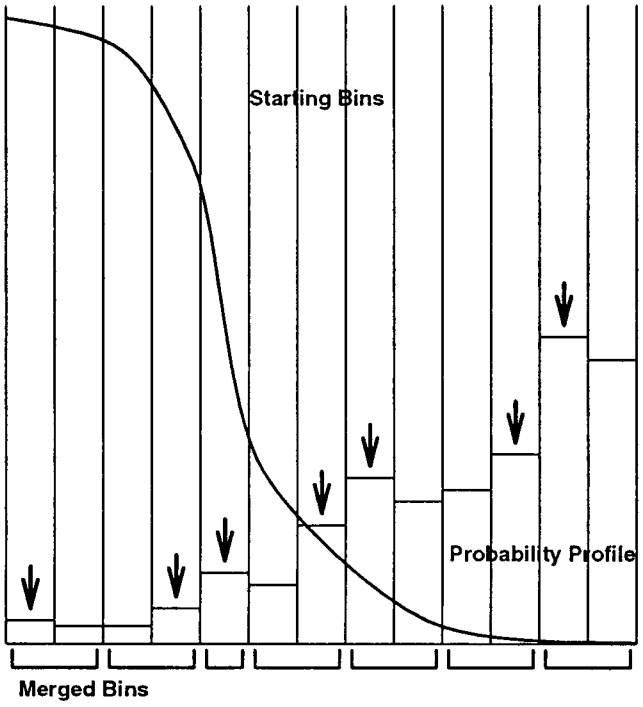

FIGURE 10 Merged bins are indicated by brackets at bottom, center bins are indicated by arrows.

at each time step for the first 11 time steps. After that, the next sampling time was $0.1t_u$ time units into the future, where t_u is the time of the present sample. This caused the sampling times to be more widely spaced as time increases, which was appropriate, because the rms value of α approximately follows an exponential decay to its final value (Huber, 1995). The process was continued until the time reached 47.6 ns, after which it was assumed that the rotations were completely randomized. The number of samples obtained was 98. These 98 times, each with 100 values of probability for the intervals, make up the distribution $P(\alpha \mid t_{out})$ used in the shortcut simulation.

Computing threshold value for reaction coordinate

When taking large steps with the ZC algorithm, it is important that the simulation not "parachute" directly into a reactive region of configuration space. If the product surface were directly accessible to the ZC moves, then many reaction events would be missed because the ZC move does not take into account the local exploration necessary to actually find the product surface. This is not a problem when the molecules are separated by a distance greater than b, because the ZC moves land the molecules too far away for reaction. Once the molecules are closer than b_1 and collisions are allowed, the possibility for missing reaction events exists. However, when ζ_3 is large enough, the possibility of a reaction occurring before again achieving a separation greater than b_1 is very small. As long as ζ_3 is large enough, it is safe to take ZC steps.

An auxilliary Brownian dynamics simulation was run to find ζ_t. The molecules were initially touching and in perfect alignment, so that all ζ_i were equal to 0. They were then stepped forward in time, using the usual time step rule, until a separation of b_2 was achieved. The value of p_3 was recorded. This was done 1000 times, after which the resulting values of ζ_3 were sorted into ascending order. The 990th value of ζ_3 was chosen to be ζ_t. Thus, only about 1% of the reactions that occur are caused by systems starting with both $\zeta_3 > \zeta_t$ and separation $< b_1$, and failing to pass beyond a separation of b_1 before reacting.

APPENDIX E

Visualization of WEB dynamics

The "big picture" for the WEB dynamics methodology described in this paper can perhaps be seen most readily with computer-aided visualization of the stochastic processes. To this end, six video sequences were compressed in the MPEG format and placed on the Internet (ftpserver ftp.engr.wisc.edu, directory pub/stkim/microhydro/mpeg). The files, named clip1.mpg through clip6.mpg, illustrate the main points behind WEB dynamics. These files may be viewed on most PCs or workstations with the aid of an mpeg-player program. These are readily available as public domain utilities on the Internet.

A short description of the sequences are as follows.

clip1.mpg A point in configuration space, with a 2-D energy surface. Narrowing of the valley walls near the saddle point represents an entropic barrier. Visualization of the stochastic process. Very low probability of crossing the energy barrier, therefore, many trajectories must be run to obtain accurate statistics to estimate the reaction rate constant.

clip2.mpg The embarassingly parallel algorithm: parallel computing helps, but not a whole lot. With N processors, we can follow many systems. Note: the trajectories (as represented by the particles) do not interact with each other. We see a few at the higher energy levels, but still, very few crossings. None during this window of observation. The problem gets worse as the barrier height increases.

clip3.mpg The WE method. The subspace orthogonal to the reaction pathway is subdivided into slabs (bins). After a finite number of steps, the particles in each bin are subdivided or merged. This renormalization leads to an equal number of particles in each bin; but now, particles carry different weights of probability. The visualization shows this with a red (high) to white (low) color scheme on a logarithmic scale spanning 6 decades. Good statistics on crossing of the energy barrier, but note that each white particle now represents a very minute quantity of probability.

clip4.mpg Stochastic simulation over the entire configuration space is inefficient. Where available, use a numerical method of solution on the associated Fokker-Planck equation for the probability density. The weighted ensemble Brownian dynamics simulation can be merged with the FPE solution, as shown in this sequence. Particles "emerge" from under the "blanket" with the latter representing the probability density function.

clip5.mpg Simulation proceeding as seen from real space. The Northrup-Erickson (1992) model. Patches (little spheres) must align for reaction to occur. No reactions during window of observation.

clip6.mpg The same, but with WE method and merge with numerical solution of FPE. One reactant reduced in size for visualization purposes. Length of tail indicates misalignment and required axis of rotation to induce alignment. Color scheme represents probability scale as before. Yellow (yolk-like) explosions represent reactions. Many reactions observed, but must scale by lower weights.

This work has been supported by Office of Naval Research Grant N00014-92-J-1564 and National Science Foundation Grant CTS-9218668. G.A. Huber is the recipient of a National Science Foundation Fellowship, University of Wisconsin WARF Fellowship, and an Amoco Foundation Fellowship.

REFERENCES

Agmon, N. 1984. Residence times in diffusion processes. *J. Chem. Phys.* 81:3644–3647.

Antosiewicz, J., M. K. Gilson, I. H. Lee, and J. A. McCammon. 1995. Acetylcholinesterase: diffusional encounter rate constants for dumbbell models of ligand. *Biophys. J.* 68:62–68.

Brune, D. A., and S. Kim. 1994. Hydrodynamics steering effects in protein association. *Proc. Natl. Acad. Sci. USA.* 91:2930–2934.

Cartling, B. 1989. From short-time molecular dynamics to long-time stochastic dynamics of proteins. *J. Chem. Phys.* 91:427–438.

Chandler, D. 1987. Introduction to Modern Statistical Mechanics. Oxford University Press, Oxford.

Creighton, T. E. 1984. Proteins: Structures and Molecular Properties. Freeman, New York.

Dwyer, J. D., and V. A. Bloomfield. 1993. Brownian dynamics simulations of probe and self diffusion in concentrated protein and DNA solutions. *Biophys. J.* 65:1810–1816.

Ermak, D. L., and J. A. McCammon. 1978. Brownian dynamics with hydrodynamic interactions. *J. Chem. Phys.* 69:1352–1360.

Farkas, L. 1927. Keimbildungsgeschwindigkeit in ubersattigen dampfen. *Z. Phys. Chem.* 125:236–242.

Finkelstein, A. V., and J. Janin. 1989. The price of lost freedom: entropy of bimolecular complex formation. *Protein Eng.* 3:1–3.

Gardiner, C. W. 1985. Handbook of Stochastic Methods, 2nd ed. Springer, Berlin.

Goldstein, H. 1980. Classical Mechanics, 2nd ed. Addison-Wesley Publishing Co., Reading.

Grabert, H., P. Hänggi, and P. Talkner. 1980. Microdynamics and nonlinear stochastic processes of gross variables. *J. of Stat. Phys.* 22:537–552.

Hamermesh, M. 1989. Group Theory and its Application to Physical Problems. Dover Publications, Inc., New York.

Hänggi, P., P. Talkner, and M. Borkovec. 1990. Reaction-rate theory: fifty years after Kramers. *Rev. of Mod. Phys.* 62:251–342.

Hill, M. D., J. R. Larus, S. K. Reinhardt, and D. A. Wood. 1993. Cooperative shared memory: software and hardware for scalable multiprocessors. *Trans. Comput. Sys.* 11:300–318.

Huber, G. A. 1995. Ph.D. Dissertation. University of Wisconsin-Madison, Madison, WI.

Kim, S., and S. J. Karrila. 1991. Microhydrodynamics: Principles and Selected Applications. Butterworth-Heinemann, Boston.

Kozack, R. E., M. J. d'Mello, and S. Subramaniam. 1995. Computer modeling of electrostatic steering and orientational effects in antibody-antigen association. *Biophys. J.* 68:807–814.

Lamm, G., and K. Schulten. 1982. Extended Brownian dynamics. II. Reactive, nonlinear diffusion. *J. Chem. Phys.* 78:2713–2734.

Lee, S., and M. Karplus. 1986. Kinetics of diffusion-limited bimolecular reactions in solution. I. General formalism and relaxation kinetics of fast reversible reactions. *J. Chem. Phys.* 86:1883–1903.

Lenoski, D., J. Laudon, K. Gharachorloo, W.-D. Weber, A. Gupta, J. Hennessy, M. Horowitz, and M. Lam. 1992. The Stanford DASH multiprocessor. *Computer.* 25:63–79.

Luty, B. A., J. A. McCammon, and H. X. Zhou. 1992. Diffusive reaction rate from Brownian dynamics simulations: replacing the outer cutoff surface by an analytical treatment. *J. Chem. Phys.* 97:5682–5686.

Luty, B. A., S. El Amrani, and J. A. McCammon. 1993a. Simulation of the bimolecular reaction between superoxide and superoxide dismutase: synthesis of the encounter and reaction steps. *J. Am. Chem. Soc.* 115: 11874–11877.

Luty, B. A., R. C. Wade, J. D. Madura, M. E. Davis, J. M. Briggs, and J. A. McCammon. 1993b. Brownian dynamics simulations of diffusional encounters between triose phosphate isomerase and glyceraldehyde phosphate: Electrostatic steering of glyceraldehyde phosphate. *J. Phys. Chem.* 97:233–237.

Madura, J. D., M. E. Davis, M. K. Gilson, R. C. Wade, B. A. Luty, and J. A. McCammon. 1994. Biological application of electrostatic calculations and Brownian dynamics simulations. *In* Reviews in Computational Chemistry. (K. B. Lipkowitz and D. B. Boyd, editors.) VCH Publishers, Inc., New York. 229–267.

McCammon, J. A., and S. C. Harvey. 1987. Dynamics of Proteins and Nucleic Acids. Cambridge University, Cambridge.

McQuarrie, D. A. 1976. Statistical Mechanics. Harper and Row, New York.

Nambi, P., A. Wierzbicki, and S. A. Allison. 1991. Intermolecular interactions between bovine pancreatic trypsin inhibitor molecules probed by Brownian dynamics simulations. *J. Phys. Chem.* 95:9595–9600.

Northrup, S. H. 1994. Hydrodynamic motions of large molecules. *Curr. Opin. Struct. Biol.* 4:269–274.

Northrup, S. H., and H. P. Erickson. 1992. Kinetics of protein-protein association explained by Brownian dynamics computer simulation. *Proc. Nat. Acad. Sci. USA.* 89:3338–3342.

Northrup, S. H., and J. A. McCammon. 1980. Efficient trajectory simulation methods for diffusional barrier crossing processes. *J. Chem. Phys.* 72:4569–4578.

Northrup, S. H., S. A. Allison, and J. A. McCammon. 1984. Brownian dynamics simulation of diffusion-influenced bimolecular reactions. *J. Chem. Phys.* 80:1517–1524.

Northrup, S. H., J. O. Boles, and J. C. L. Reynolds. 1988. Brownian dynamics of Cytochrome c and Cytochrome c peroxidase association. *Science.* 241:67–70.

Northrup, S. H., K. A. Thomasson, C. M. Miller, P. D. Barker, J. G. Eltis. Guillemette, Inglis, S. C., and Mauk, A. G. 1993. Effects of charges amino acid mutations on the bimolecular kinetics of reduction of yeast iso-1-ferricytochrome c by bovine ferrocytochrome b_5. *Biochemistry.* 32:6613–6623.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1988. Numerical Recipes in C. Cambridge University Press, Cambridge.

Traenkle F., M. D. Hill, and S. Kim. 1995. Solving microstructure electrostatics on a proposed parallel computer. *Comput. Chem. Eng.* 19: 743–757.

E. J., and A. H. Cromer. 1971. *Theoretical Mechanics.* John Wiley and Sons, Inc., New York.

von Smoluchowski, M. 1917. Versuch einer mathematischen theorie der koagulationskinetik kolloider losungen. *Z. Phys. Chem.* 92:129–168.

Zheng, L. H., and Y. C. Chiew. 1988. Computer simulation of diffusion-controlled reactions in dispersions of spherical sinks. *J. Chem. Phys.* 90:322–327.