Brownian Simulation of Many-Particle Binding to a Reversible Receptor Array

Arieh L. Edelstein and Noam Agmon

Department of Physical Chemistry and the Fritz Haber Research Center, The Hebrew University, Jerusalem 91904, Israel

Received February 21, 1996; revised June 24, 1996

The principles and practice of a many-body Brownian dynamics algorithm of reversible binding to a static three-dimensional receptor array are presented. The mixed boundary conditions at the array are compared with single-particle direct propagation using a novel operator discretization. The many-body aspects are checked against previous one-dimensional Brownian simulations. The long-time behavior agrees with expected analytic solutions. © 1997 Academic Press

I. INTRODUCTION

Signalling between cells in a living organism is vital for the growth, differentiation, and functioning of an ensemble of cells as a single organism. An important mechanism for conveying information between cells involves the diffusion of small ligands and their binding to cell-membrane-bound receptors [1]. Known examples of such ligands are hormones and neurotransmitters.

The theoretical study of this problem yielded a simple analytical approximation for the steady-state, irreversible binding rate coefficient to receptors distributed on the surface of a (spherical, planar) membrane [1–3]. This is an interesting solution to a problem which, unfortunately, falls short of describing even the simplest case of receptor action. First, the ligands diffuse in the liquid phase in between membranes of adjacent cells and not in an infinite volume. Such geometric restrictions could have important consequences for biological reactions. Second, a detailed time evolution is required, not only steady-state properties. Finally, the binding *must* be reversible if the transduced signal is ever to terminate. The study of reversible diffusion influenced reactions [4, 5] is more complicated than that of irreversible reactions due to the convoluted nature of the dissociation–recombination events and to correlations between particles induced by the ability of a site to bind just a single particle at a time.

The basic model of receptor action we suggest to study is a simplified representation of neurotransmitter release into the synaptic cleft [6] and its reversible binding to a receptor-array on the postsynaptic membrane (Fig. 1). Electrophysiological methods have produced quantitative recordings from ionic channels in the postsynaptic membrane due to receptor binding in both neuromuscular junctions [7] and the central nervous system [8, 9]. The process begins by exocytosis of neurotransmitters from a vesicle(s) in the presynaptic membrane, their diffusion in the synaptic gap, reversible binding to receptors (each composed of two binding sites), and the opening of ionic channels in response to the binding event.

In the model under consideration, N transmitter molecules (noninteracting point particles) are initially at one point on the inner face of the "ceiling" of the simulation box (Fig. 1). They diffuse (with a diffusion coefficient D) between the ceiling and the "floor," on which a static array of binding sites is located. The particles bind reversibly to the sites subject to the restriction that any given site may bind only one particle. We wish to find the probability that a given site is in the bound state by time t after particle release.

It is evident that the question posed has no simple analytic solution, and one needs recourse to simulation techniques. In the present work we will describe an efficient numerical algorithm providing an accurate approximation to the *many-body* diffusion equation with the mixed boundary conditions which depict *reversible* binding to sites on the "floor" of the simulation box. We shall concentrate on the (nontrivial) numerical aspects of the problem by extending our earlier one-dimensional many-body Brownian dynamics (BD) algorithm for reversible binding [10–13]. To date, such a problem has been addressed either by chemical–kinetic rate equations [14, 15], by solving bulk (mean-field) diffusion equations [16, 17], or by Monte-Carlo simulations [18–20], apparently less accurate than the BD algorithm described below.

Two types of numerical methodologies may be considered, corresponding to the duality between Fick's and Langevin's equations [21]. The first is a direct propagation (DP) of the partial differential equations governing the time-evolution of the particle spatial distribution [22]. Unless the particles are independent, this joint probability density is a function of $3N$ coordinates [11, 23], which makes its time-propagation prohibitive except for a very

0021-9991/97 $25.00

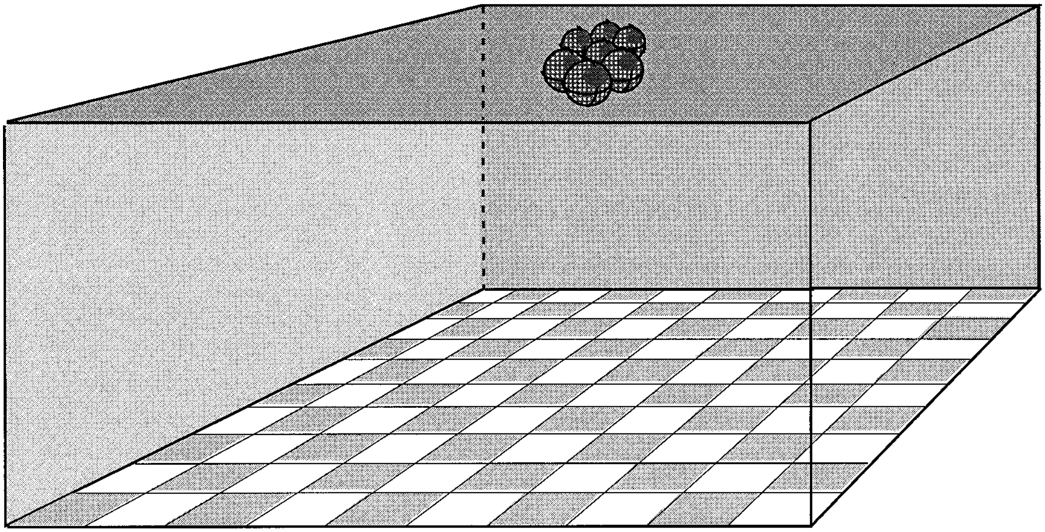

FIG. 1. The receptor-array geometry.

small number of particles (perhaps just one). Since the probability density is represented on a grid, the method is also limited to rather simple geometries which do not require too many grid points.

We present a three-dimensional DP routine as a check on the simulation procedure described below. It is applied to a single receptor with mixed (reversible-reflective) boundary conditions on the "floor" of the diffusion box. Such boundary conditions are notoriously difficult to handle analytically [24, 25] but are nevertheless important for problems other than receptor binding, such as chemical reactions at microelectrodes [25, 26]. Time propagation is achieved, as before, using Chebyshev polynomials [22]. This allows reaching long times with high accuracy.

When many particles and complex geometries are involved, there is no alternative to simulation techniques. In these methods, each particle is moved every time-step to generate a "stochastic trajectory" [21]. Averaging over many trajectories generates the probability density function as obtained (when feasible) by a single DP run, but with statistical noise, the penalty paid for saving huge amounts of computer memory required for representing $3N$-dimensional functions on discrete grids. This penalty is worth paying, since it makes the solution feasible.

There are two types of stochastic simulation algorithms. The simple-minded Monte-Carlo algorithm implements a lattice random-walk. For free diffusion, each particle is moved with equal probabilities in the six canonical directions. At a reversible site, the particle may enter or exit with probabilities related to the binding and unbinding rate coefficients [18, 27]. A drawback is that such a procedure requires sticking with the small elementary time-step throughout the simulation. Biological processes occur on timescales varying from femtoseconds to days. For example, an elementary particle hop along its diffusive trajectory may take 1 nanosecond, while one is monitoring processes occurring, as in the neurotransmitter case, in the millisecond to second range. A trajectory of over 10^{10} steps, while quite common in nature, can be computationally expensive. In addition, the computation quality may deteriorate by the accumulation of small errors over that many steps. Finally, part of the computational effort might be in vain. For example, one knows that the solution for free diffusion is a gaussian which widens *ad infinitum* [21]. Any computational effort that goes into the calculation of the free diffusion part of the solution teaches us nothing new.

Brownian dynamics (BD) is a method that circumvents these problems and allows using large time-steps [28, 29]. The philosophy behind BD is to apply random numbers from a locally valid analytical solution. Thus a free diffusion trajectory is generated by gaussian rather than uniform random numbers [30]. In this simple case the time-step may be arbitrarily large without loss in accuracy. The situation is more complicated if one needs to take into account long-range interactions or complicated (e.g., reactive) boundary conditions.

For describing the kinetics near a binding site, one might take a random step from the solution of the semi-infinite one-dimensional binding problem, followed by a gaussian step in each parallel direction. The approximation is then in the assumption that the parallel and perpendicular motions are independent. This evidently restricts the time-step near the site, but it is nevertheless larger than the elementary Monte-Carlo step. An excellent discussion of these principles is given by Lamm and Schulten [29]. Their application allowed Northrup, McCammon, and collaborators [31–33] to simulate several problems of biological interest, including the cell-bound receptor problem discussed above [33].

Their BD simulation of steady-state rate coefficients has been generalized to *time-dependent* rate-coefficients [34].

The above work concentrated on irreversible reactions, with two consequences. First, in the absence of any restriction on the number of bound particles, there are no correlations between particles and it suffices to solve for single-particle diffusion [31–34]. Despite the introduction of Ref. [29], DP is the method of choice for single-particle diffusion even in three dimensions, as we demonstrate below. Thus for the irreversible problems BD was actually inessential. In contrast, reversible binding to saturable sites is a coupled many-body problem even if the particles are otherwise non-interacting. In such cases BD becomes an indispensable tool.

The approximation behind our reversible many-body BD application [10–13] involves a separation of time scales between the fast Brownian motion and the slower onset of particle–particle correlations, induced by the reversible binding sites. Thus a particle is randomly chosen and moved for a finite time interval, Δt, assuming that during this time the remaining $N - 1$ particles are immobile. This procedure is repeated N times and the clock advanced one Δt unit. After a few Δt units such a simulation converges to the result obtained with a smaller Δt [11].

The second difference between reversible and irreversible binding is that the latter allows the usage of the "rejection method" for obtaining random numbers from a given distribution (Ref. [30, Section 3]), namely, from the solution of the diffusion equation near a binding site. The rejection method employs a "comparison function" which is everywhere *above* the given distribution. A first uniform random number generates a random number from the comparison function while a second one accepts or rejects this choice based on the ratio of the two functions. For irreversible binding, the reflective solution (zero recombination), which is everywhere above the reactive solution, is a viable comparison function [29, 31–33]. In contrast, the reversible solution may be either smaller or larger than the reflective solution since a trap can release a particle after long times. Here the reflective solution is not a valid comparison function and the rejection method cannot be adopted. One recourses to the straightforward method of inverting indefinite integrals.

The present publication reports a first application of BD for many-particle reversible reactions in three dimensions for the receptor-array of Fig. 1. It also describes a finite-differencing implementation for mixed boundary conditions in constructing the spatial operator for a single-particle DP calculation (Section IV). The two computations are compared. This is one in a series of tests reported in Section V, demonstrating the accuracy and reliability of our algorithm. Additional checks include one-dimensional BD [13] and analytic asymptotic behavior.

II. THE BROWNIAN RECEPTOR-ARRAY ALGORITHM

A. *Problem Definition*

We have developed a computer program for simulating many-particle reversible binding to the receptor array shown in Fig. 1. The system consists of two parallel planes representing inert and impermeable membranes. Their separation (in the z direction) is L_z. The space between the membranes is filled with liquid in which noninteracting, point particles diffuse (diffusion coefficient D). The "floor" (bottom plane) contains a rectangular "receptor array" of dimensions L_x and L_y (Fig. 2). The origin of the coordinate system is in the center of this rectangle. The array consists of M rectangles of dimensions l'_x by l'_y, where $l'_x \le l'_y$. Thus $L_x L_y = M l'_x l'_y$. Each rectangle contains a central, square binding site (length $l \le \min(l'_x, l'_y)$, area $a = l^2$) with unreactive margins around it. Regions not covered by binding sites (the margins, top plane, etc.) are reflective towards the moving particles. Particles hitting such regions bounce back into the simulation box.

A site may bind at most one particle at a time. When a particle touches its surface, binding may occur with a rate coefficient $a\kappa_r$ (κ_r is the reactive flux perpendicular to the area a), *provided* that it is unoccupied. If a particle is bound, the surface of the trap becomes inert (which is equivalent to setting $\kappa_r = 0$). In the present version, particles have no volume and do not interact, except for the case of an occupied trap, when a bound particle prevents other particles from binding. The bound particle may in turn dissociate to the surface with a rate coefficient κ_d. We assume that after particle dissociation it has equal probability of being located at any point on the site's surface.

Each binding site may represent a distinct (single-site) receptor or else the sites are paired-off and each pair is a receptor (double-site receptors). While it is simpler to analyze single-site receptors theoretically, the double-site receptors approximate more closely the experimental situation for many synaptic receptors [6].

The receptor array may be enclosed within four reflective walls (bold lines, Fig. 2, +W case). These prevent particles from escaping from the "active zone" above the array. In the case without the walls (−W) particles still cannot pass the top and bottom planes, so they spread between two infinite impermeable planes. This can be simulated as easily as the +W case, since Brownian dynamics is an off-grid method which keeps in memory only particle coordinates. The extent of the physical space is immaterial.

Initially, the N particles are released from some point within the simulation box. For $t > 0$ the particles move randomly (diffuse) in the three-dimensional space between the two planes. When a particle hits a plane it is reflected back into the diffusion space unless it hits the surface area of an unoccupied binding site, as described above. Thus at any time instant, particle i is either free within the diffusion space or bound to site j. The coordinates $(x_i(t), y_i(t), z_i(t))$, $i = 1 \cdots N$, provide a complete description of the many-body dynamics. The crux of the algorithm involves generation of these coordinates for such a stochastic trajectory using rules for particle motion corresponding as closely as possible to the solution of the diffusion equation for the given geometry and boundary conditions.

B. *The Way Particles Move*

For every single elementary step we choose randomly a particle to move. This particle, initially at (x_0, y_0, z_0), is moved along the three Cartesian coordinates, x, y, and z. The endpoint of its move is determined by three random numbers using an algorithm which depends on the time-step and geometry as described below. Subsequently another particle is picked randomly and moved by the same algorithm. After N elementary moves (i.e., after each particle moves on the average once) the clock is advanced by one time-step unit.

The procedure for moving a particle determines the accuracy and efficiency of the algorithm. Efficiency means being able to take large time-steps while accuracy requires small time-steps, particularly near the receptor array. It is the presence of this array which causes the many-particle trajectory to deviate from the simple solution of free-diffusion (in the presence of reflecting walls) for independent particles.

The strategy of the present algorithm involves a choice of both time-steps and random numbers. We use large time-steps (ΔT) far from the array and smaller steps (Δt) closer by. Far from the array the dynamics approaches free-diffusion. There we use three independent gaussian random numbers namely, random numbers out of a gaussian distribution with a width corresponding to the appropriate time-step. Closer to the array the possibility of binding need be taken into account. We assume that binding/unbinding occurs only perpendicular to the site's surface, i.e., for a move in the z-direction. Thus, while in the x and y directions gaussian random numbers are applied, the z-motion is instigated by a random number out of the analytic solution in the presence of a reversible trap [10, 35, 36].

Whatever the time-step, a relocated particle may find itself across an impermeable plane. Such planes include the upper membrane ("ceiling"), the lower membrane ("floor") outside the receptor array (in the −W case), the reflective walls (in the +W case), the unreactive margins of the binding sites and any occupied site. In all such cases the particle is reflected back from the impermeable plane.

Following a choice for the large time-step, ΔT, we distinguish between the following four cases:

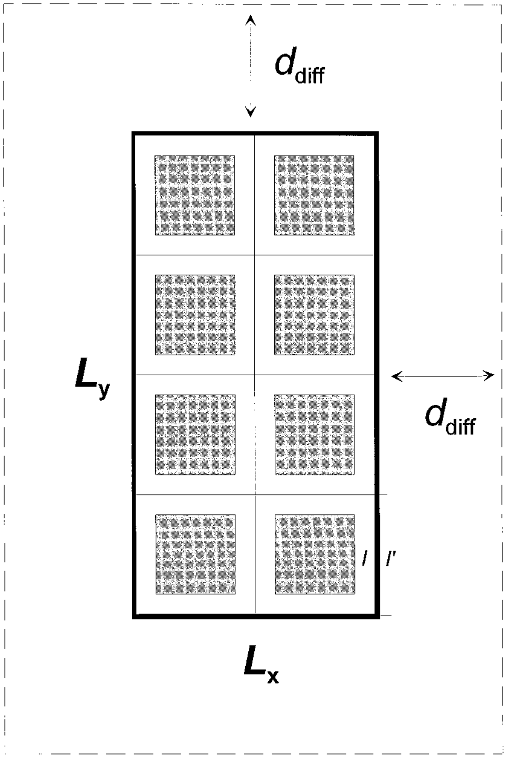

FIG. 2. The geometry of the "floor" of the simulation box. The figure shows a 2 × 4 array of square receptors (gray). The boundary of the array (thick line) is composed of impermeable walls (perpendicular to the floor) in the +W case. In the −W case, the floor (and ceiling) continue indefinitely in both directions. Here $l'_x = l'_y = l'$.

1. *Large time-step free diffusion.* For particles furthest away from the traps

$$z_0 > d_{\text{diff}} \equiv a_{\text{diff}} \sqrt{2D\,\Delta T},$$

or

$$|x_0| > L_x/2 + d_{\text{diff}}$$

or

$$|y_0| > L_y/2 + d_{\text{diff}}, \tag{1}$$

the large time-step, ΔT, is implemented. The last two lines in Eq. (1) are only relevant in the −W case, for particles outside the dashed rectangle in Fig. 2. We find that $a_{\text{diff}} = 7$ is sufficiently large to guarantee accuracy (Section V below).

The free diffusion mechanism is applied by selecting three random numbers from a gaussian distribution of width $\sqrt{4D\,\Delta T}$. For free diffusion in a given coordinate, e.g., z, the probability density $p(z, t \mid z_0)$ for a particle initially at z_0 to move to point z by time t is

$$p_{\text{free}}(z, t \mid z_0) = (4\pi Dt)^{-1/2}\, e^{(\zeta-\zeta_0)^2}, \tag{2}$$

where we have defined

$$\zeta \equiv z/\sqrt{4Dt}. \tag{3}$$

Since the *area* under a probability density function is uniformly distributed in [0, 1], given a uniformly distributed random number, $0 \leq \xi \leq 1$, the endpoint of the trajectory, z_ξ, is calculated by setting $t \equiv \Delta T$ in

$$\xi = \int_{-\infty}^{z_\xi} p_{\text{free}}(z, t \mid z_0)\, dz. \tag{4}$$

Therefore, one has

$$\zeta_\xi - \zeta_0 = \text{erf}^{-1}(2\xi - 1) \equiv \xi_{\text{gauss}}, \tag{5}$$

where $t \equiv \Delta T$, $\text{erf}(z) \equiv \sqrt{4/\pi} \int_0^z \exp(-z'^2)\, dz'$ is the error function, and ξ_{gauss} is a gaussian random number. Given a uniform deviate, ξ, there are more efficient ways for calculating a gaussian deviate than inverting an error function [30], which we apply. The same procedure is used to move the particle along x and y.

2. *Short time-step free diffusion.* The second option applies for particles at an intermediate distance from the nearest trap

$$\begin{aligned} &z_{\text{rev}} < z_0 < d_{\text{diff}}, \\ &|x_0| - L_x/2 < d_{\text{diff}}, \\ &|y_0| - L_y/2 < d_{\text{diff}}, \end{aligned} \tag{6}$$

where all three inequalities hold and

$$z_{\text{rev}} = \begin{cases} \min(l'_x, l'_y, L_z), & (x_0, y_0, 0) \text{ vacant trap}, \\ 0, & (x_0, y_0, 0) \text{ inert}. \end{cases} \tag{7}$$

Roughly speaking, when $z < d_{\text{diff}}$ a particle senses the geometry of the receptor array, whereas if $z < z_{\text{rev}}$ and the trap is directly underneath the particle, it also senses the binding reaction.

Particles at distances larger than z_{rev} are far enough not to feel the effect of binding to the traps, yet we do not want them to "fly" above several traps or to jump from a distance directly onto the trap surface within one free-diffusion time-step. Therefore, the time-step is limited to Δt which is determined from

$$\begin{aligned} \min(l'_x, l'_y) &\equiv a_{\text{rev}} \sqrt{2D\, \Delta t'}, \\ n &= \max(\text{int}(\Delta T/\Delta t'), 1), \\ \Delta t &\equiv \Delta T/n. \end{aligned} \tag{8}$$

Here int(z) is the integer part of z and we find (Section V) as an optimal value $a_{\text{rev}} = a_{\text{diff}} = 7$.

The free diffusion mechanism is applied n times using the short time-step value Δt, meaning that three random, $\sqrt{4D\, \Delta t}$ wide gaussian numbers are generated for each of the n moves. To find z_ξ we set $t \equiv \Delta t$ in Eq. (5), and similarly for the x and y directions. If, however, during the small step $k < n$ the particle crosses the outer boundary at d_{diff}, it executes the remaining time-step, $(n - k)\, \Delta t$, in one move. If it crosses the inner boundary at z_{rev} the remaining steps are executed using the algorithm below.

3. *Receptor affected move.* This option applies to unbound particles located such that $z_0 < z_{\text{rev}}$ and $(x_0, y_0, 0)$ is on the surface of a vacant trap. The x and y motions are executed using gaussian random numbers with the small time-step Δt. The z motion is calculated using the following procedure with $t \equiv \Delta t$.

Denote by $P(t \mid z_0)$ the probability that a single particle, initially a distance z_0 from a reversible trap, will be bound by time t. In one-dimension and in the absence of particle–trap interactions [35]

$$P(t \mid z_0) = \frac{\kappa_r}{\Delta} \exp(-\zeta_0^2)\, [\phi(\zeta_0 + \mu_-) - \phi(\zeta_0 + \mu_+)], \tag{9}$$

where we have defined

$$\begin{aligned} \Delta &\equiv (\kappa_r^2 - 4D\kappa_d)^{1/2}, \\ \mu_\pm &\equiv (\kappa_r \pm \Delta)t/\sqrt{4Dt}, \\ \phi(z) &\equiv \exp(z^2)\, \text{erfc}(z). \end{aligned} \tag{10}$$

Erfc is the complementary error function which becomes complex if $\Delta^2 < 0$.

The endpoint of a z-move is found by comparing a uniformly distributed random number, $0 < \xi < 1$, with $P(t \mid z_0)$: if $\xi < P(t \mid z_0)$ the particle ends up in the trap, whereas if $\xi > P(t \mid z_0)$ it remains unbound. In the first case, additional moves along x and y are not executed. In the latter case, the final distance from the trap surface, z_ξ, is found from

$$\xi - P(t \mid z_0) = \int_0^{z_\xi} p(z, t \mid z_0)\, dz, \tag{11}$$

where $p(z, t \mid z_0)$ is the probability density for a particle initially at z_0 to be at point z after time $t \equiv \Delta t$, given a reversible trap at $z = 0$. For single-particle, one-dimensional diffusion with a reversible trap at the origin, this function is known analytically [35],

$$
\begin{aligned}
p(z,t\,|\,z_0) = (4\pi\,Dt)^{-1/2} \\
&\{\exp[-\,(\zeta-\zeta_0)^2] + \exp[-\,(\zeta+\zeta_0)^2]\} \\
&+ \frac{\kappa_r}{2D\Delta}\exp[-\,(\zeta+\zeta_0)^2] \qquad (12) \\
&[(\kappa_r-\Delta)\,\phi(\zeta+\zeta_0+\mu_-) \\
&-(\kappa_r+\Delta)\,\phi(\zeta+\zeta_0+\mu_+)].
\end{aligned}
$$

Consequently, given a uniform random number, ξ, the final value, $z_\xi \equiv \sqrt{4Dt}\zeta_\xi$, is

$$
\begin{aligned}
\xi - P(t\,|\,z_0) &= [\mathrm{erf}(\zeta_\xi+\zeta_0) + \mathrm{erf}(\zeta_\xi-\zeta_0)]/2 \\
&+ \frac{2\kappa_r}{\Delta}\{\mu_-\,[\Phi_-(\zeta_\xi+\zeta_0) - \Phi_-(\zeta_0)] \qquad (13) \\
&-\mu_+\,[\Phi_+(\zeta_\xi+\zeta_0) - \Phi_+(\zeta_0)]\},
\end{aligned}
$$

where we have defined

$$\Phi_\pm(z) = \int_0^z e^{-x^2}\,\phi(x+\mu_\pm)\,dx. \qquad (14)$$

First we apply the z-move, calculating the integral with the help of the lookup procedure described below. Subsequently, two random gaussian x and y moves are applied with the short time step $t \equiv \Delta t$. If the particle ends up out of the cube composed by the trap square as a base, the free-diffusion mechanism is applied for the next step.

4. *Initially bound particle.* When a particle is initially bound to a reversible trap in one dimension, the probability that it remains bound after time $t \equiv \Delta t$ is given by [36]

$$P(t\,|\,*) = [(\kappa_r+\Delta)\phi(\mu_-) - (\kappa_r-\Delta)\phi(\mu_+)]/2\Delta. \qquad (15)$$

Note that the initially bound state is denoted by an asterisk. Given a uniformly distributed random number ξ, the particle remains bound at the end of the time-step Δt if $\xi < P(\Delta t\,|\,*)$ and is released otherwise. If it remains bound, we proceed to the next time-step without executing the x and y moves.

If the particle is released, the z-endpoint of the elementary move is found from Eq. (11) with $z_0 = *$. Since the probability density for an initially bound particle is [36]

$$p(z,t\,|\,*) = \frac{\kappa_d}{2\Delta}\exp(-\zeta^2)\,[\phi(\zeta+\mu_-) - \phi(\zeta+\mu_+)], \qquad (16)$$

the final value, $z_\xi \equiv \sqrt{4Dt}\zeta_\xi$, is given by

$$\xi - P(t\,|\,*) = \frac{\kappa_d}{\Delta}\sqrt{Dt}\,[\Phi_-(\zeta_\xi) - \Phi_+(\zeta_\xi)], \qquad (17)$$

with $\Phi_\pm$ defined in Eq. (14) and $t \equiv \Delta t$.

The x and y coordinates of the released particle are determined with the aid of *four* random numbers. Two uniform random numbers are used to place the particle at a randomly chosen point on the trap surface. This assumes that all points on the surface have equal binding properties. Two gaussian random numbers are then used to move the particle from the release point in the x and y directions.

C. *The Simplified Flow-Chart of the Main Loop*

A flow-chart implementing the above criteria for particle motion is shown in Fig. 3. This is the main loop of the receptor-array algorithm, involving the motion of a single particle. (A)–(U) correspond to the notations in the figure:

(A) We choose a particle randomly; then (B) we check how far it is from the nearest trap, Eq. (1).

(C) If it is sufficiently far for employing the long time-step free-diffusion mechanism, Eq. (1), we choose three random gaussian numbers, ξ_{gauss}, and a time-step ΔT, to move the particle along x, y, and z. Here, the elementary step is over and we return to stage A.

(D) If check B fails, we set the short time-step value, Δt, and initialize the counter ($k = 1, ..., n$) of Δt, Eq. (8).

(E) We check whether the particle is sufficiently far for using the free-diffusion mechanism, Eq. (6). If so, then after stepping the counter (F), we apply the three gaussian random numbers, ξ_{gauss}, to move the particle through a time-step Δt (G).

If by doing so we complete the necessary number of short time-steps (H), we go to the beginning of the loop.

If there are still some short time-steps to execute, we first check (I) if after the previous move the particle left the d_{diff} region and if so (J), it completes the remaining part of the time-step in one move.

If E failed and the particle is closer to the traps than z_{rev}, then after stepping the counter (K) we check if the chosen particle is trapped (L).

If so (M), we use a uniform random number, ξ, and Eq. (15) to decide whether it remains trapped (N).

If it does, then after stepping the counter (O) and checking the remaining moves to be completed (P), we pick up the corresponding chain.

If the check was negative and the particle leaves the trap (Q), then its endpoint along z is calculated with Eq. (17), while along x and y gaussian random numbers are applied to a random point within the trap surface.

If the particle was not initially trapped, stage (L), we check if it is above any trap (R) and if not, we apply the chain starting from (G) as described.

If the particle is above a vacant trap, we utilize Eqs. (9) and (13) to calculate its endpoint (S).

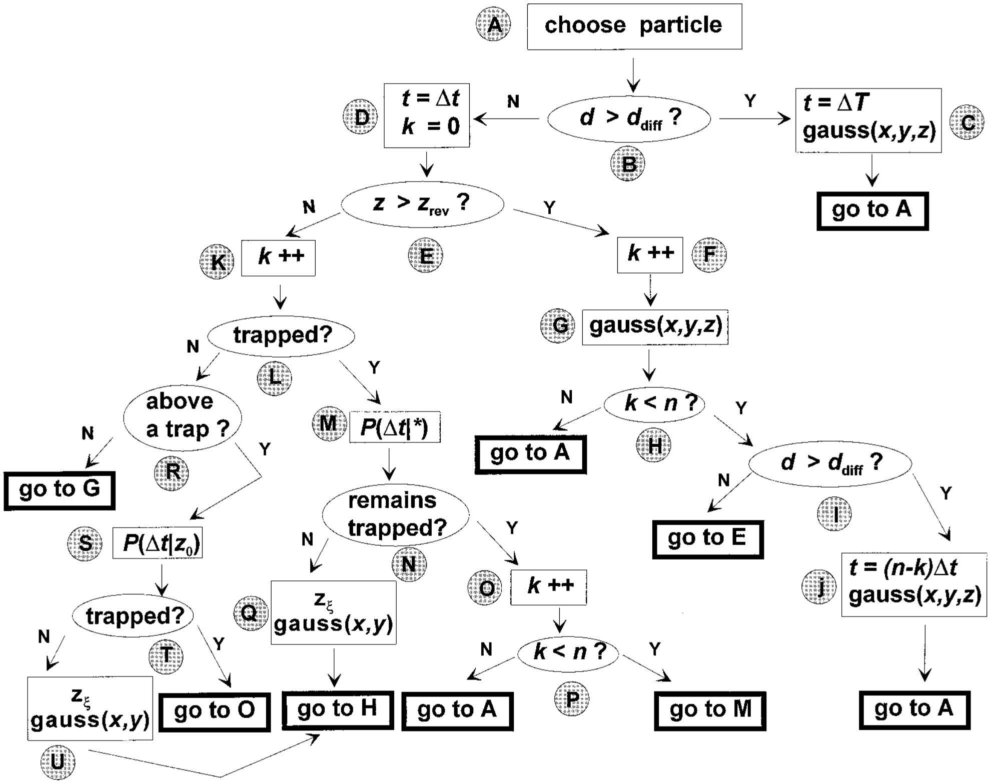

FIG. 3. A flow diagram of the main-loop in the simulation.

If the particle ends up trapped (T) we continue from (O). If not, the particle is moved to (x, y, z_ξ), where z_ξ is calculated from Eq. (13) and x and y with random gaussian numbers (U).

D. *The Lookup Procedure*

Lookup tables are used whenever the chosen particle is either bound or located within the interval $0 \le z_0 \le z_{\rm rev}$. The lookup algorithm below determines the elementary trajectory endpoint for a particle using a uniform random number ξ. Slightly different lookup procedures are applied to Eqs. (9), (13), and (17), as described below:

1. Eq. (9): *trapping an unbound particle initially at* z_0. The interval $[0, z_{\rm rev}]$ is divided into $n_{\rm bin}$ integration bins. Usually, a few thousand bins produce sufficient accuracy. In actual calculations we use $n_{\rm bin} = 20000$. $P(t \mid z)$ is evaluated at $z = iz_{\rm rev}/n_{\rm bin}$, where $i = 1, 2, ..., n_{\rm bin}$, and saved as a vector $P(i)$. Given an initial location $z_0 < z_{\rm rev}$, we define $j = n_{\rm bin}z_0/z_{\rm rev}$ and check whether $\xi < P(j)$, in which case the particle gets trapped. Otherwise, it remains free and we need to find its endpoint.

2. Eq. (13): *moving an unbound particle from* z_0 *to* z. Since $0 \le z \le 2z_{\rm rev}$, erf(z) is tabulated in the interval $[-z_{\rm rev}, 3z_{\rm rev}]$, using $4n_{\rm bin}$ points and $\mu_-\Phi_-(z) - \mu_+\Phi_+(z)$ is tabulated in $[0, 3z_{\rm rev}]$ using $3n_{\rm bin}$ points [10]. The two properly scaled functions are saved as vectors $f(i)$ and $g(i)$, respectively. The lookup procedure then reads

```
  scale=nbin/zrev
  l1=0
  l2=nbin*2
  irc=scale*z0
1 ircn=(l2+l1)/2
  ircnp=irc+ircn
  ircnm=irc-ircn
  IF(g(ircnp)-g(irc)+f(ircnm)
   +f(ircnp).gt.xi)                THEN
  l2=ircn
                                   ELSE
  l1=ircn                                    (18)
  ENDIF
  IF(l1.eq.l2-1)THEN
```

```
c adding the gaussian moves in X and Y
  Z=l2/scale
  X=X+gauss()*short_time_step
  Y=Y+gauss()*short_time_step
  ...
  ...
                        ELSE goto 1
  ENDIF
```

3. Eq. (17): *freeing a bound particle.* First we compare ξ with $P(t\,|\,*)$, Eq. (15), to determine if the particle remains bound. This is analogous to the binding of a free particle, item 1 above. If not, the particle is freed to a point z using Eq. (17). The last equation is evaluated using "inverted" tables; i.e., we keep the function value as the table index, whereas z_ξ is kept as its value. This speeds up the lookup procedure considerably since, instead of scanning the table, one just picks up the corresponding element. The x and y coordinates are then determined by adding a gaussian random number to a random point on the trap surface.

```
      Z=inverted_table(xi)
      X=trap_x+(.5-rndm())*trap_side
      Y=trap_y+(.5-rndm())*trap_side    (19)
      X=X+short_time_step*gauss()
      Y=Y+short_time_step*gauss()
```

E. *Output*

During a stochastic trajectory, the program retains information on the spatial location of each particle, $x_i(t)$, $y_i(t)$, and $z_i(t)$, i =1, ..., N, which is either within the simulation box or bound to site j. This allows calculating the binding state of each site,

$$P_j(t) = \begin{cases} 0, & \text{site free} \\ 1, & \text{site occupied,} \end{cases} \tag{20}$$

where $j = 1, ..., M$.

When the single-trajectory binding states are averaged over an ensemble of trajectories, one obtains the individual-site binding probabilities $\langle P_j(t)\rangle$. Summing over all sites gives the total binding (or "activation") probability,

$$\langle P(t)\rangle \equiv \frac{1}{M}\sum_{j=1}^{M}\langle P_j(t)\rangle, \tag{21}$$

for single-site receptors. Irrespective of the site-multiplicity, $M\langle P(t)\rangle$ is the average number of bound particles and $M\langle P(t)\rangle/N$ is the fraction of bound particles per the total bound plus free particles.

For double-site receptors, assuming independent sites, the total activation probability, $\langle P^{(2)}(t)\rangle$, is calculated from

$$\langle P^{(2)}(t)\rangle \equiv \frac{2}{M}\sum_{j/2=1}^{M/2}\langle [P_{j-1}(t) + P_j(t)]_{\text{mod}2}\rangle, \quad M \text{ even.} \tag{22}$$

Only pairs in which *both* sites are bound contribute to $\langle P^{(2)}\rangle$. The program can report any of the four types of binding probabilities as output.

III. MONTE-CARLO VS BROWNIAN DYNAMICS

Broadly speaking, Monte-Carlo (MC) and Brownian dynamics (BD) methods differ in the rules for moving a particle: in the first method [27] one uses the infinitesimal transition probability (usually on a grid) while in BD one applies an exact local solution (never on a grid). Existing algorithms for simulating reversible binding to receptor arrays [18, 20] represent a hybrid approach: the exact gaussian propagator has been applied in free space, but for the reversible binding process a MC procedure was implemented.

To compare the two methods, let us consider the dissociation step with the fundamental dissociation rate parameter κ_{d}. When the time-step $\Delta t \to 0$, the dissociation probability $1 - P(\Delta t \mid *) \to \kappa_{\text{d}}\,\Delta t$. For a particle correctly dissociated based on the comparison with a uniform random number, $0 \le \xi \le 1$, Δt has to be really "infinitesimal." To circumvent this requirement, the MC algorithms [18, 20] compared ξ with

$$P(\Delta t\,|\,*) \approx \exp(-\kappa_{\text{d}}\,\Delta t). \tag{23}$$

This is a solution for $\kappa_r = 0$, which reduces to $1 - \kappa_{\text{d}}\,\Delta t$ in the limit that $\Delta t \to 0$. Unfortunately, it is not much of an improvement because the limit $\kappa_r = 0$ is never realized in the present problem.

The exact solution for dissociation in one dimension on a semi-infinite line when $\kappa_r \neq 0$ is given by Eq. (15). Its small Δt limit is obtained by approximating $\phi(z) \to \exp(z^2)$, giving

$$P(\Delta t\,|\,*) \to [(\kappa_r + \Delta)\exp(\mu_-^2) - (\kappa_r - \Delta)\exp(\mu_+^2)]/2\Delta. \tag{24}$$

This agrees with Eq. (23) only when $\kappa_r = 0$, since then $\mu_\pm = \pm i\sqrt{\kappa_{\text{d}}\,\Delta t}$, where $i \equiv \sqrt{-1}$ (see Eq. (10)). But in receptor binding κ_r is typically large, so that Eq. (23) is not even the correct short-time approximation to Eq. (15). When Δt becomes really "infinitesimal," $\exp(z^2) \to 1 +$

z^2, so that Eq. (24) tends to $1 - \kappa_d \, \Delta t$ (see Eq. (10)). This is the only limit where the expressions agree. In MC algorithms that have not implemented geometry-sensitive time-steps [18, 20], Δt should be severely restricted.

The problem compounds when one considers how far from a trap a dissociating particle moves in the z direction. The exact one-dimensional solution, Eq. (16), implies that the endpoint is chosen from a range of distances with the probability dictated by Eq. (17). This takes into account the distribution of dissociation times within Δt and the possibility of several binding cycles. The MC simulations assumed that the particle is moved a fixed distance proportional to $\sqrt{D \, \Delta t}$, which does not even seem to be the correct average of Eq. (16). Only when $\Delta t \to 0$ does this distance vanish and the two approaches coincide.

The MC recipes did not specify to where, in the xy plane, the particle is moved. It is incorrect to move it to a fixed point above the trap center. Rather, it should be moved with equal probability to an arbitrary point on the trap surface followed by gaussian moves in the x and y directions. We have found that omission of even this single detail results in wrong long-time behavior (Section V.F).

IV. DIRECT PROPAGATION WITH MIXED BOUNDARY CONDITIONS

"Direct propagation" (DP) is the solution of the partial differential equation for the probability density, p, in space and time. For one particle there are three spatial coordinates, whereas the joint probability density for N-particles depends on $3N$ spatial coordinates. Unlike Brownian dynamics (BD) which is an off-grid method, here a $3N$-dimensional grid is required to hold the values of p at any given time. Therefore we apply DP in the single particle case as a check of the BD algorithm. While DP, especially with the Chebyshev propagator [22], can be extremely accurate, it is limited by the grid size to rather simple geometries and just a few particles. For a single particle, it is definitely the method of choice.

Here we apply DP as a means of checking the BD algorithm. However, the DP algorithm may itself be nontrivial, due to the "mixed" boundary conditions on the "floor" of the simulation box. These involve reversible boundary conditions on the surface of the site and reflective elsewhere. With care, such boundary conditions may be implemented using a minimal number of grid points. The algorithm for doing so is described below and implemented for a single trap occupying the center of the floor's area.

A. *Basic Equations*

In this case of no long-range interactions a simple three-dimensional diffusion equation for the probability density $p(x, y, z; t)$ is solved,

$$\frac{\partial p(x,y,z;t)}{\partial t} = D\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) p(x,y,z;t). \tag{25}$$

The walls and ceiling are impermeable to particles; hence they are reflective. Thus the derivative of $p(x, y, z; t)$ perpendicular to these planes vanishes. The floor is reflective outside the site area (centered at (0, 0, 0)) and may bind a particle reversibly within the site perimeters, where the "back-reaction" boundary conditions [35] are imposed,

$$D \left.\frac{\partial p(x,y,z;t)}{\partial z}\right|_{z=0} = \begin{cases} 0, & \text{if } |x| \text{ or } |y| > l/2 \\ \kappa_r p(x,y,0;t) - \kappa_d P(t)/a, & \text{otherwise.} \end{cases} \tag{26}$$

This is the mixed boundary condition which complicates the algorithm. The particle is released uniformly over the area $a = l^2$ of the binding site. Finally, the binding probability,

$$P(t) \equiv 1 - \int_{-L_x/2}^{L_x/2} \int_{-L_y/2}^{L_y/2} \int_0^{L_z} p(x,y,z;t) \, dx \, dy \, dz, \tag{27}$$

obeys the "rate equation"

$$\frac{dP(t)}{dt} = \kappa_r \int_{-l/2}^{l/2} \int_{-l/2}^{l/2} p(x,y,0;t) \, dx \, dy - \kappa_d P(t), \tag{28}$$

which is obtained by integrating Eqs. (25) and (26) over the volume of the box. This $P(t)$ is equivalent to the ensemble-averaged $\langle P(t) \rangle$ generated by the corresponding BD simulation.

Due to the local nature of the boundary conditions, the operator in space is evaluated by finite differencing, whereas temporal propagation is carried out with the Chebyshev propagator [22], which allows large time-steps with high accuracy. A fixed grid is used in the three directions, $\Delta x = \Delta y = \Delta z$. Discretizing, $x = i \, \Delta z$, $y = j \, \Delta z$, and $z = k \, \Delta z$ with $-n_x \le i \le n_x$, $-n_y \le j \le n_y$, and $1 \le j \le n_z$. Inside the diffusional domain,

$$\begin{aligned} \frac{\partial p(i,j,k;t)}{\partial t} = D[&p(i-1,j,k;t) + p(i+1,j,k;t) \\ &+ p(i,j-1,k;t) + p(i,j+1,k;t) \\ &+ p(i,j,k-1;t) + p(i,j,k+1;t) \\ &- 6p(i,j,k;t)]/\Delta z^2, \end{aligned} \tag{29}$$

whereas on its boundaries the equation is modified to take the boundary conditions into account. First we set reflective boundaries on all sides of the diffusion box and then modify them on the trap surface to account for back-reaction.

B. *Reflective Boundaries*

Every grid point in the interior of the diffusion box has six neighbors, leading to Eq. (29). Points on its sides have only five neighbors. For example, points on the ceiling do not have a $k + 1$ neighbor, so that a term $p(i, j, k + 1; t) - p(i, j, k; t)$ drops from the equation, resulting in the appropriate boundary condition there. Points along the intersection of two boundary planes have only four neighbors, whereas points in the corners of the box have three neighbors each. The boundaries of a box thus include 6 planes, 12 lines, and 8 corners. If each of these is treated separately, the subroutine for calculating the spatial operator requires over 100 lines of programming.

An alternative approach evaluates the operator in just a few lines as follows. First set dp(i,j,k) to zero. This matrix is going to contain the result of operating on p(i,j,k) with the discretized Laplacian. The main loop, written in FORTRAN, may then look like

```
    Dbar=D/dz/dz
    do 1 i=-nx,nx
    do 1 j=-ny,ny
    do 1 k=1,nz
    if(k.lt.nz)  dp(i,j,k)=dp(i,j,k)
     +Dbar*(p(i,j,k+1)-p(i,j,k))
    if(k.gt.1)   dp(i,j,k)=dp(i,j,k)
     +Dbar*(p(i,j,k-1)-p(i,j,k))
    if(j.lt.ny)  dp(i,j,k)=dp(i,j,k)      (30)
     +Dbar*(p(i,j+1,k)-p(i,j,k))
    if(j.gt.-ny) dp(i,j,k)=dp(i,j,k)
     +Dbar*(p(i,j-1,k)-p(i,j,k))
    if(i.lt.nx)  dp(i,j,k)=dp(i,j,k)
     +Dbar*(p(i+1,j,k)-p(i,j,k))
    if(i.gt.-nx) dp(i,j,k)=dp(i,j,k)
     +Dbar*(p(i-1,j,k)-p(i,j,k))
  1 continue
```

It is easy to verify that this single loop contains all the different cases discussed above.

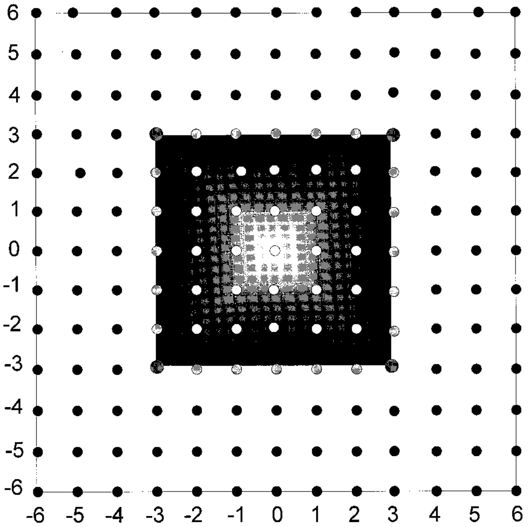

FIG. 4. The first layer of the three-dimensional grid on the diffusion box "floor" with a square binding site (black). Here $n_x = n_y = 6$ and $n_l = 3$. "Interior" grid points are white, "rim" points are light gray, "corner" points are gray (and larger), grid points external to the binding site are black. Rim points are $\frac{1}{2}$-interior $\frac{1}{2}$-exterior while corner points are $\frac{1}{4}$-interior $\frac{3}{4}$-exterior.

C. *Mixed Reversible/Reflective Boundary Conditions*

The "floor" of the diffusion box is a L_x by L_y rectangle that contains a $l \times l$ square trap in its center (Fig. 4). On this plane one has mixed boundary conditions: when either $|x| > l/2$ or $|y| > l/2$ the floor is reflective, whereas when both $|x|$ and $|y|$ are smaller than $l/2$ a reversible boundary condition, Eq. (26), is imposed. Suppose that $L_x/2$, $L_y/2$, and $l/2$ are divisible by Δz. Thus $l/2 = n_l \Delta z$, so that the trap is represented by the $(2n_l + 1)^2$ grid points for which $-n_l \leq i \leq n_l$ and $-n_l \leq j \leq n_l$ and $k = 1$.

Out of these points, those with $i = n_l$ or $j = n_l$ are on the boundary of the trap. Thus there are $(2n_l - 1)^2$ interior points, $4(2n_l - 1)$ rim points and four corner points, $(2n_l + 1)^2$ in total, as seen in Fig. 4. Points in the interior represent an area Δz^2 each. A rim point, which is divided equally between the trap and the reflective surrounding, represents a reactive area of $\Delta z^2/2$, whereas a corner point accounts for an area $\Delta z^2/4$. Since the binding probability per unit time is proportional to the surface area involved, an accurate representation of the binding reaction requires dividing κ_r by the corresponding factor, 2 and 4 for rim and corner points, respectively.

When unbinding, a particle is assumed to end with equal probability at an arbitrary point on the trap surface. Discretizing, the dissociation rate coefficient for interior grid points on the trap surface is $k_\text{d} = \kappa_\text{d}\,\Delta z^2/a = \kappa_\text{d}/(4n_l^2)$, where $a = 4(n_l\,\Delta z)^2$ is the trap area. Subsequently, the dissociation rate coefficient into rim points is $k_\text{d}/2$ and into corner points $k_\text{d}/4$. Since there are $(2n_l - 1)^2$ interior points, $4(2n_l - 1)$ rim points, and 4 corner points, the total dissociation rate coefficient is conserved

$$(2n_l - 1)^2 k_\mathrm{d} + 4(2n_l - 1)(k_\mathrm{d}/2) + 4(k_\mathrm{d}/4) = \kappa_\mathrm{d}. \quad (31)$$

In FORTRAN, the calculation of the operator with the mixed boundary conditions continues from Eq. (30) with the following loop

```
    kr=kappar/dz
    kd=kappad/(2*nl)**2
    dPP=0.d0
    do 1 i=-nl,nl
    do 1 j=-nl,nl
    ddp=kd*PP-kr*p(i,j,1)                  (32)
    if(i.eq.-nl.or.i.eq.nl) ddp=ddp/2
    if(j.eq.-nl.or.j.eq.nl) ddp=ddp/2
    dPP=dPP-ddp
    dp(i,j,1)=dp(i,j,1)+ddp
  1 continue
```

The variable PP is the binding probability, $P(t)$, and dPP the outcome of the operation on PP (its time derivative).

D. *Symmetry Reduction*

The geometry of the problem under investigation has a fourfold symmetry, namely, both xz and yz planes are mirror planes. If the initial distribution possesses the same symmetry, for example, when it is a point delta-function at the center of the ceiling, the numerical effort may be reduced by a factor of 4. The symmetry-reduced problem involves just $\frac{1}{4}$ of the box say for $0 \le x \le L_x/2$ and $0 \le y \le L_y/2$, imposing reflecting boundary conditions at $(0, y, z)$ and $(x, 0, z)$.

The first step of setting the surfaces of the quadrant reflecting is easily achieved by replacing -nx and -ny by 0 in Eq. (30). The second step, of setting reversible boundary conditions on a quarter of the trap surface, $0 \le x \le l/2$ and $0 \le y \le l/2$, should be done with care, since the bound state distributes population to all four quadrants.

Imagine the floor folded into four along the x and y axes. Then every grid point not on a fold gains a multiplicity of 4; i.e., it represents four coalescing grid points. Every grid point on a fold has a multiplicity of 2 and a point at the origin (on both folds) has a multiplicity of 1. In the recombination direction this multiplicity is of no concern. The recombination rate parameter from every grid point on the trap surface is $\kappa_r/\Delta z$. The multiplicity is taken care of by the initial distribution which is multiplied by the corresponding factors so that it normalizes to unity in the $\frac{1}{4}$-box.

For a reversible reaction, the dissociated population density is released in proportion to the surface area associated with a given grid point. Thus κ_d should be scaled by the multiplicity factors. The do loop in Eq. (32) is therefore replaced by

```
    do 1 i=0,nl
    do 1 j=0,nl
    kr4=kr
    kd4=kd
    if(i.eq.0.or.i.eq.nl) kd4=kd4/2
    if(j.eq.0.or.j.eq.nl) kd4=kd4/2          (33)
    if(i.eq.nl) kr4=kr4/2
    if(j.eq.nl) kr4=kr4/2
    ddp=kd4*PP-kr4*p(i,j,1)
    dPP=dPP-ddp
    dp(i,j,1)=dp(i,j,1)+ddp
  1 continue
```

V. TESTS AND RESULTS

Testing a new program is important since even small errors in the algorithm may lead to unexpected artifacts. Simple problems usually have several limits which can be worked out analytically. Spherically symmetric diffusion is such an example [37]. The testing task becomes more difficult for more complex problems, when analytical solutions are either not available or describe limits which are too trivial to be meaningful. Our approach for checking the algorithm involves three stages. First we perform internal convergence test(s) to select the range of internal parameters (e.g., the time-step) for which convergence is obtained. Next, we compare the output from the new program with previously developed algorithms, in this case other DP and BD programs that tackle a simplified yet nontrivial part of the problem. Finally, we consider the physical behavior of the BD solution at long times to see whether the correct equilibrium limit is approached and whether this occurs with a reasonable limiting behavior. In all the calculations described below, dimensionless time and distance units are used in which $D = 1$ and $\kappa_r = 1$. The initial condition is a delta function, with all particles at the center of the ceiling.

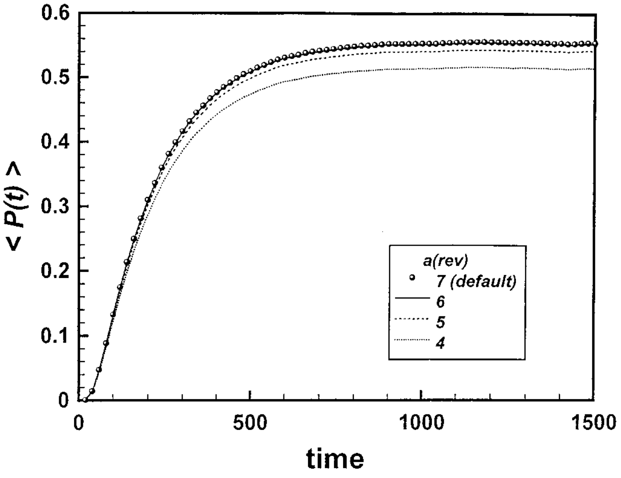

FIG. 5. Convergence of the time-dependent binding probability with respect to the time-step as determined by a_{rev}, Eq. (8). In the present case $d_{\text{diff}} = z_{\text{rev}}$ so that $\Delta t = \Delta T$. Calculated is the +W single-site, single-particle binding probability for a 5 × 5 site on a 10 × 10 floor, where $l = 5$, $l'_x = l'_y = 10$, $L_z = 20$, and $\kappa_d = 0.01$. The particle is initially at (0, 0, 20). In all of our calculations $\kappa_r = 1$ and $D = 1$.

A. *Convergence Tests*

The most important convergence test concerns the choice of the time-step. When the boundary condition on the floor is homogeneous (one binding site covering the whole floor) and the initial distribution uniform in the *xy* plane the problem reduces to one dimension (z). If, in addition, the ceiling is infinitely far away the algorithm becomes exact because it samples from the analytic semi-infinite reversible solution. In this limit the time-step may be arbitrarily large. When the ceiling is a finite distance from the floor the time-step is restricted by the requirement that a particle does not cross this distance in a single hop [13]. When the boundary condition on the floor is inhomogeneous, the time step is further limited to avoid taking big jumps across inhomogeneous boundaries parallel to the floor.

(i) *Optimizing* a_{rev}. In the present algorithm the small time-step, Δt, is related to the inhomogeneity repeat length (l'_x or l'_y) by the parameter a_{rev} in Eq. (8). Thus convergence depends on the choice of a_{rev}. This is tested in Fig. 5 for a single particle and a single site covering $\frac{1}{4}$ the floor area. For $a_{\text{rev}} > 6$ the calculation is fully converged. Since this was the observation also for other trap geometries, we use $a_{\text{rev}} = 7$ as the default value in our calculations. When applicable, we also set $a_{\text{diff}} = 7$ in Eq. (1).

(ii) *Time-step independence.* We next test the accuracy and efficiency with respect to the choice of the large time-step, ΔT. As Fig. 6 shows, with the above choice of a_{rev} and a_{diff} the −W simulations are independent of ΔT. In this example the time-step above the array is restricted to Δt, so that ΔT is only used for particles displaced sideways from the d_{diff} perimeter of the array (see Fig. 2). We also find that for the present algorithm, increasing ΔT by a factor of 100 makes the computation only 1.5 times faster. This is because many of the particles are close to the receptor-array during most of the simulation time. When κ_{d} increases, more particles escape and computer time savings are more substantial. For $\kappa_{\text{d}} = 10$ we find under the same conditions that increasing ΔT by a factor of 100 results in a 2.5 times faster computation.

B. *Comparison with One-Dimensional Dynamics*

The accuracy of treating the many-body aspects of the problem can be evaluated from a comparison with previous one-dimensional BD results. When the site covers the whole floor and the initial distribution is uniform in the *xy* plane, the problem becomes one-dimensional (in z) and thus solvable with the previous one-dimensional version of the program [13]. Since here the initial distribution is a point (rather than a planar) delta-function, we need to take a very narrow "pipe" (L_x, $L_y \ll L_z$) in order for its solution to approach the one-dimensional case. Now the boundary conditions are simpler than in the previous test; they are homogeneous, not "mixed." However, many-particle aspects come into play since we can now have $N > 1$.

Figure 7 compares three-dimensional BD in a narrow pipe (symbols) with our previous one-dimensional BD simulations [13] (lines). The good agreement demonstrated for two rather different values of κ_{d} leads us to believe

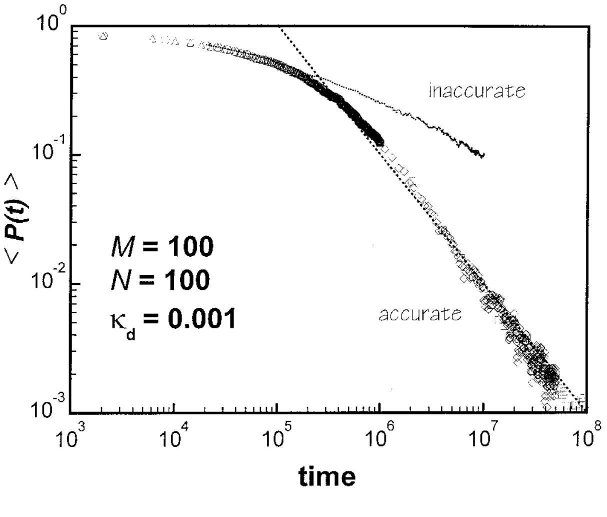

FIG. 6. The long-time decay of the average binding probability to a receptor array without walls. Here $l = l'_x = l'_y = 10$, $L_x = L_y = 100$, and $L_z = 10$. Circles, triangles, diamonds, and squares are from Brownian simulations with $\Delta T = 1$, 100, 500, and 1000 time units, respectively. Dashed line is a $1/t$ asymptotic decay. The "inaccurate version is discussed in Section V.F."

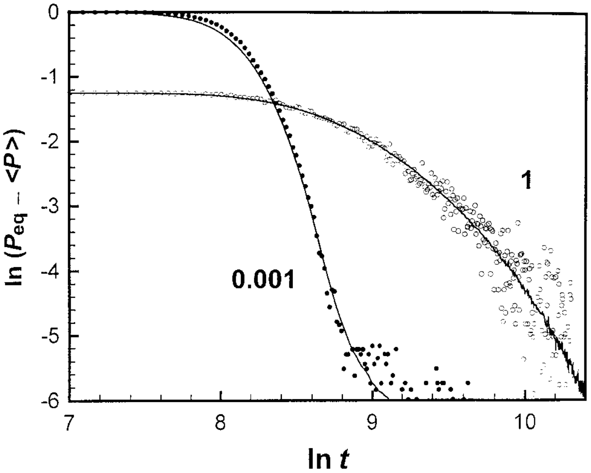

FIG. 7. Three-dimensional Brownian simulations in a narrow pipe (circles) are compared with one-dimensional simulations (solid lines, from Fig. 7 of Ref. [13]) for the two indicated values of κ_d. The pipe involves a single 0.001 × 0.001 site covering the whole floor of a cell of height L_z = 250. $P(t)$ is averaged over about 5000 trajectories of N = 100 particles.

that the many-body effect is described correctly in the new three-dimensional BD program. Our one-dimensional version has been previously validated by comparison with MC [27] and analytical one-dimensional solutions [10, 11]. The comparison with MC Ref. [11, Fig. 2], showed reasonable agreement, but only up to intermediate times (approached using logarithmic time-steps). Due to increasing errors in the MC routine, it was impossible to propagate it any further in order to obtain the asymptotic solution. The long-time behavior for different κ_d values could only be obtained with the BD routine.

C. *Comparison with Direct Propagation*

We can estimate the accuracy of treating the "mixed," inhomogeneous boundary conditions on the cell's floor by comparing DP and BD calculations for a single particle in a box containing one site occupying $\frac{1}{4}$ of the floor area. Because of the spatial restriction on the diffusion space, $+W$, the binding probability [$P(t)$ for DP, $\langle P(t) \rangle$ for BD] increases to an equilibrium plateau. Figure 8 (upper panel) shows a difference of no more than 1–2% between the two methods. It is probably due to the finite grid size in the DP method. Although DP has a much higher (double-precision) accuracy irrespective of Δt, this refers to the *discrete* operator defined on a given spatial grid. The grid size determines how well this approximates the continuum limit, which is simulated by BD. The figure is also a check on the correct procedure for constructing the diffusion operator in the DP method; without the proper scaling of rim and corner points described in Section IV, the discrepancy increases to about 6%.

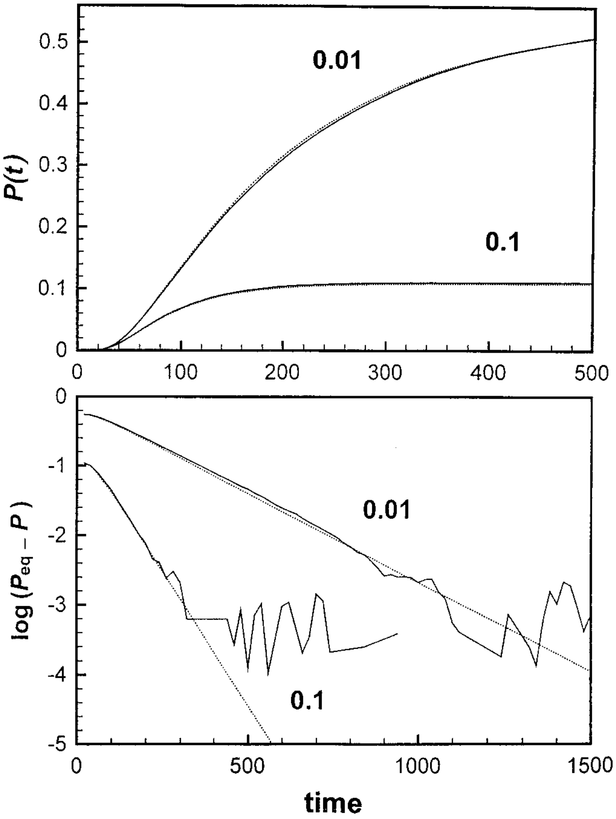

FIG. 8. Comparison of Brownian simulations (solid lines) with direct propagation results (dashed lines) for the geometry of Fig. 5 and the two indicated values of κ_d. The latter method used a 21 × 21 × 81 symmetry-reduced grid with $n_x = n_y = 20$, $n_z = 81$, and $n_l = 10$; see Section IV.

Despite the small difference between the two calculations, when each is subtracted from its corresponding P_{eq} limit a three-digit agreement is obtained (lower panel). The exponential decay of the DP calculation, performed in double-precision accuracy, continues for many more decades. It is governed by the lowest eigenvalue, λ, of the diffusion operator which has been calculated from the slopes of these curves and collected in Table I. The comparison in Fig. 8 is limited to three digits by the accuracy of the BD simulation, which depends on the square root of the number of stochastic trajectories, n_t. Since the N parti-

TABLE I

Approach to Equilibrium in the Single-Site Single-Particle Case

κ_d	P_{eq}	Eq. (35)	λ	Eq. (36)
0.1	0.111 ± 0.001	0.11111	7.80×10^{-3}	1.15×10^{-2}
0.01	0.555 ± 0.001	0.55555	2.56×10^{-3}	8.7×10^{-3}

Note. Data (P_{eq} and λ) from Fig. 8.

cles are nearly independent, the statistical noise actually scales as $\sqrt{Nn_t}$, where Nn_t is the total number of single-particle trajectories.

D. *The Ultimate Equilibrium Limit*

In the +W case of reflecting walls surrounding the receptor array, the system approaches an ultimate equilibrium limit. We can use this limit as yet another check on the algorithm. For M binding sites and N particles, the law of mass-action predicts that P_{eq}, the binding probability for any one of the M sites at equilibrium, is given by

$$P_{\text{eq}} = c_{\text{free}}K_{\text{eq}}/(1 + c_{\text{free}}K_{\text{eq}}). \tag{34}$$

Here $c_{\text{free}} \equiv [N - (M - 1)P_{\text{eq}}]/V$ is the concentration of *free* particles in the volume $V = L_xL_yL_z$. The free particles are not bound to any of the other $M - 1$ sites (except the one site under consideration). $K_{\text{eq}} = a\kappa_r/\kappa_{\text{d}}$ is the equilibrium recombination constant for each site. Substituting, one obtains the following quadratic equation for P_{eq}

$$(M - 1)P_{\text{eq}}^2 - (V/K_{\text{eq}} + N + M - 1)P_{\text{eq}} + N = 0. \tag{35}$$

Although for the present problem there is no rigorous proof for the law of mass-action, thus of Eq. (35), it is quite plausibly exact.

In the single-particle single-site case of Fig. 8, Table I shows that the equilibrium limit of the BD simulations indeed agrees with Eq. (35). To check the situation in the many-particle multi-site +W case, we calculated $\langle P(t)\rangle$ for different values of κ_{d} and N (see Fig. 9). The solution of Eq. (35), dashed lines, agrees nicely with the asymptotic limit of our BD simulations. The results in Fig. 9 suggest that both Eq. (35) is exact and the BD algorithm is correct.

E. *The Approach to Equilibrium*

In the +W case the ultimate approach to equilibrium is exponential, $\exp(-\lambda t)$. This is due to the reflective walls; λ is the lowest eigenvalue of the diffusion operator within the finite volume involved. In one dimension [13] it is possible to derive a rigorous expression and show that it reduces approximately to

$$\frac{1}{\lambda} \approx \frac{1}{ca\kappa_r + \kappa_d} + \frac{L_z^2}{D\pi^2}. \tag{36}$$

This resembles a familiar recipe for the effective rate-constant of serial processes, which for the present problem involve diffusion to the site and reaction at its surface. By a similar logic one derives the steady-state rate coefficient for surface-bound receptors [3]. For the problem treated in Fig. 8, $N = 1$ so that Eq. (36) is only a rough approximation (Table I) which cannot be used as a check on the algorithm.

The −W case does not approach equilibrium. Here the binding probability decays to zero as shown in Fig. 6. For a single site in an infinite d-dimensional space the asymptotic decay of the binding probability follows the $t^{-d/2}$ law [5, 11, 12, 38]. It is plausible (although, again, without rigorous proof) that a similar asymptotic decay holds also in the many-site case. This follows because at long times particles diffuse to large distances from the receptor-array. From a distance, the whole array looks like a single site. Since diffusion here is in between two planes, it should appear two-dimensional at long times, leading to an asymptotic $1/t$ decay. This is indeed observed in the simulation shown in Fig. 6 (dashed line).

F. *An Inaccurate Version*

In order to see how meaningful these checks are, an "inaccurate version" of the program has been produced. This version contains small modifications of the accurate version which alter the x, y coordinates of the released particle. Specifically, a dissociated particle is moved to a single point above the trap center (rather than uniformly to any point on its surface) and without adding two gaussian steps in the x and y directions. This saves four random numbers (cf. discussion following Eq. (17)), but makes this version noticeably less accurate than the correct version.

An example is shown in Fig. 6: after some time the binding probability is overestimated by the inaccurate version. The reason is that keeping the dissociated particle above the trap from which it was just released increases its rebinding probability. In comparison, the accurate treatment allows for lateral motion during the elapsed time. This added motion may bring the particle to a location above an occupied trap or an inert surface, decreasing its recombination probability. We have checked that the "inaccurate version" produces similar errors also for the examples in Figs. 8 and 9. This demonstrates how seemingly innocent changes in the algorithm may lead to sizeable errors.

VI. CONCLUSION

The present work detailed the theory and practice of constructing a three-dimensional Brownian simulation algorithm for many-particle reversible binding to a receptor array. The algorithm combines geometry-sensitive time-steps with analytical local solutions to produce an accurate representation of the diffusion equations depicting reversible binding under the given geometrical constraints. Checking and debugging such an algorithm is quite demanding. One must search for nontrivial limits of the problem which are either known analytically or solvable by

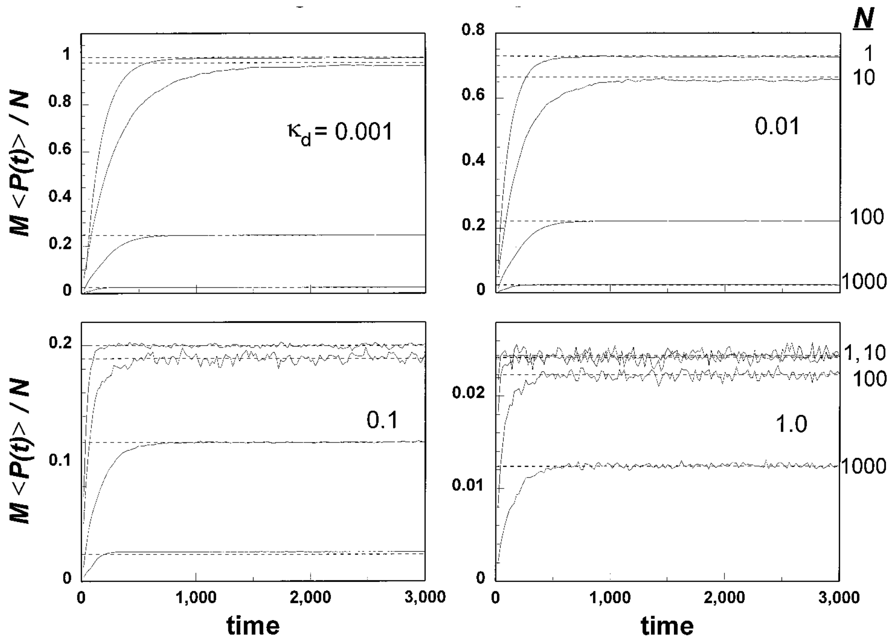

FIG. 9. The fraction of bound particles as obtained from a Brownian simulation of a square receptor-array with walls. It involves 5 × 5 array of $M = 25$ sites, each a 10 × 10 square on a 100 × 100 floor. Thus $L_x = L_y = 100$, $L_z = 10$, $l = 10$, and $l'_x = l'_y = 20$. Each panel refers to a different κ_d value, as indicated, for which simulations with $N = 1, 10, 100$, and 1000 particles were performed (solid lines). The dashed lines are the equilibrium limits, $MP_{\rm eq}/N$, with $P_{\rm eq}$ from Eq. (35).

other algorithms. The present algorithm has been subjected to an extensive series of such tests.

The algorithm has many potential applications, from the action of neurotransmitters in the synaptic gap between nerve cells to modelling reversible attachment to chromatography columns in analytical chemistry. The question is whether the added accuracy and flexibility is really essential? Given the quality of the present experimental data, perhaps less sophisticated Monte-Carlo routines are sufficient.

We think a moral can be learned from our experience in solution-phase simulations of reversible reactions [11]. First, the accurate simulations allowed for theoretical development; approximate solutions could be compared with the simulation results. By using less sophisticated Monte-Carlo routines it was impossible to reach the asymptotically long-time behavior which distinguishes between the various approximations. The theory for reversible binding to receptor arrays is at its infancy. The availability of a well-defined model which can be solved numerically with sufficient accuracy is the only meaningful test for such theories because experiment might contain additional complicating factors.

Second, our solution-phase simulations predicted a switch-over into an asymptotic power-law phase in the approach to equilibrium. Such behavior has not been observed before due to the relatively low resolution of experimental techniques. Following our prediction, the pH effect on proton recombination to excited dye molecules has been remeasured with an enhanced-resolution time-correlated single-photon counting apparatus [39]. The new data show a switch-over into an asymptotic power-law behavior. In the more complicated case of receptor arrays, the range of expected physical behaviors is yet unexplored. We hope to use the Brownian dynamics routine for mapping out the possible physical behaviors.

ACKNOWLEDGMENTS

Work supported by The Israel Science Foundation, administered by The Israel Academy of Sciences and Humanities. The Fritz Haber Research Center is supported by the Minerva Gesellschaft für die Forschung, Munich, Germany.

REFERENCES

1. H. C. Berg and E. M. Purcell, *Biophys. J.* **20,** 193 (1977).
2. C. DeLisi and F. W. Wiegel, *Proc. Nat. Acad. Sci. USA* **78,** 5569 (1981).
3. D. Shoup and A. Szabo, *Biophys. J.* **40,** 33 (1982).
4. O. G. Berg, *Chem. Phys.* **31,** 47 (1978).
5. N. Agmon and A. Szabo, *J. Chem. Phys.* **92,** 5270 (1990).
6. P. Jonas and N. Spruston, *Curr. Opinion Neurobiol.* **4,** 366 (1994).
7. K. L. Magleby and C. F. Stevens, *J. Physiol.* **223,** 173 (1972).
8. J. C. Eccles and J. C. Jaeger, *Proc. R. Soc. London Sect. B* **148,** 38 (1958).
9. H. Korn and D. S. Faber, *Trends Neurosci.* **14,** 439 (1991).
10. A. L. Edelstein and N. Agmon, *J. Chem. Phys.* **99,** 5396 (1993).

11. N. Agmon and A. L. Edelstein, *J. Chem. Phys.* **100,** 4181 (1994).
12. A. L. Edelstein and N. Agmon, *J. Phys. Chem.* **99,** 5389 (1995).
13. N. Agmon and A. L. Edelstein, *Biophys J.* **68,** 815 (1995).
14. J. C. Wathey, M. M. Nass, and H. A. Lester, *Biophys. J.* **27,** 145 (1979).
15. H. Parnas, M. Flashner, and M. E. Spira, *Biophys. J.* **55,** 875 (1989).
16. A. Friboulet and D. Thomas, *J. Theor. Biol.* **160,** 441 (1993).
17. W. R. Holmes, *Biophys. J.* **69,** 1734 (1995).
18. T. M. Bartol Jr., B. R. Land, E. E. Salpeter, and M. M. Salpeter, *Biophys. J.* **59,** 1290 (1991).
19. D. S. Faber, W. S. Young, P. Legendre, and H. Korn, *Science* **258,** 1494 (1992).
20. L. M. Wahl, C. Pouzat, and K. J. Stratford, *J. Neurophysiol.* **75,** 597 (1996).
21. S. Chandrasekhar, *Rev. Mod. Phys.* **15,** 1 (1943).
22. N. Agmon and R. Kosloff, *J. Phys. Chem.* **91,** 1988 (1987).
23. N. Agmon, *Phys. Rev. E* **47,** 2415 (1993).
24. D. Shoup, G. Lipari, and A. Szabo, *Biophys. J.* **36,** 697 (1981).
25. A. Szabo, *J. Phys. Chem.* **91,** 3108 (1987).
26. A. J. Bard *et al., J. Anal. Chem.* **63,** 1282 (1991).
27. N. Agmon, H. Schnörer, and A. Blumen, *J. Phys. Chem.* **95,** 7326 (1991).
28. D. L. Ermak and J. A. McCammon, *J. Chem. Phys.* **69,** 1352 (1978).
29. G. Lamm and K. Schulten, *J. Chem. Phys.* **78,** 2713 (1983).
30. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing* (Cambridge Univ. Press, Cambridge, 1986).
31. S. H. Northrup, M. S. Curvin, S. A. Allison, and J. A. McCammon, *J. Chem. Phys.* **84,** 2196 (1986).
32. J. A. McCammon, S. H. Northrup, and S. A. Allison, *J. Phys. Chem.* **90,** 3901 (1986).
33. S. H. Northrup, *J. Phys. Chem.* **92,** 5847 (1988).
34. H.-X. Zhou, *J. Phys. Chem.* **94,** 8794 (1990).
35. N. Agmon, *J. Chem. Phys.* **81,** 2811 (1984).
36. N. Agmon, E. Pines, and D. Huppert, *J. Chem. Phys.* **88,** 5631 (1988).
37. E. B. Krissinel' and N. Agmon, *J. Comput. Chem.* **17,** 1085 (1996).
38. A. Szabo, *J. Chem. Phys.* **95,** 2481 (1991).
39. E. Pines, unpublished data.