Stochastic off-lattice modeling of molecular self-assembly in crowded environments by Green's function reaction dynamics

Byoungkoo Lee,[1] Philip R. LeDuc,[2] and Russell Schwartz[3]
[1]*Joint Program in Computational Biology, Carnegie Mellon University and University of Pittsburgh, 654 Mellon Institute, 4400 Fifth Avenue, Pittsburgh, Pennsylvania 15213, USA*
[2]*Department of Mechanical Engineering, Carnegie Mellon University, 415 Scaife Hall, 5000 Forbes Avenue, Pittsburgh, Pennsylvania 15213, USA*
[3]*Department of Biological Sciences, Carnegie Mellon University, 654 Mellon Institute, 440 Fifth Avenue, Pittsburgh, Pennsylvania 15213, USA*

(Received 31 March 2008; revised manuscript received 16 July 2008; published 12 September 2008)

The environment inside a living cell is dramatically different from that found in *in vitro* models, presenting a problem for computational models of biochemistry that are only beginning to capture these differences. This deviation between idealized *in vitro* models and more realistic intracellular conditions is particularly problematic for models of molecular self-assembly, but also specifically hard to address because the large sizes and long assembly times of biological self-assembly systems force the use of highly simplified models. We have developed a prototype of a molecular self-assembly simulator based on the Green's function reaction dynamics (GFRD) model to achieve more realistic models of assembly in the crowded conditions of the cell without unduly sacrificing tractability. We tested the model on a simple representation of dimer assembly in a two-dimensional space. Our simulations verify that the model is computationally efficient, provides a realistic quantitative model of reaction kinetics in uncrowded conditions, and exhibits expected excluded volume effects under conditions of high crowding. This work confirms the effectiveness of the GFRD technique for more realistic coarse-grained modeling of self-assembly in crowded conditions and helps lay the groundwork for exploring the effects of *in vivo* crowding on more complex assembly systems.

DOI: 10.1103/PhysRevE.78.031911 PACS number(s): 87.15.R−, 82.39.Rt, 83.10.Rs, 87.16.dr

I. INTRODUCTION

Intracellular biochemistry occurs in an environment that is strikingly different from that found in *in vitro* solution-based reaction systems and traditionally assumed by idealized computational models. The interior of a living cell is characterized by high heterogeneity of reactants, the presence of numerous components of cellular machinery that assist or regulate various kinds of chemistry, and dense crowding of macromolecules [1]. A typical concentration of macromolecules for an *in vitro* study is 1–10 mg/ml, whereas the total concentration of macromolecules in the cytoplasm is 50–400 mg/ml [2]. For example, the estimated concentration of macromolecules in *Escherichia coli* is 300–400 mg/ml [3]. Moreover, a cell contains many immobilized proteins, which further calls into question some of the basic assumptions drawn from models of solution-based chemistry. For instance, one-third of the glycolytic enzymes in the squid giant axon are nondiffusible [4]. Also, larger molecules are less diffusible in a cytoplasm than in an *in vitro* solution with almost the same molecular concentration as the cytoplasm [5–7]. These deviations between the intracellular condition and *in vitro* models have been found to have large effects on many biological reaction systems, especially those involving macromolecular assembly processes. For example, molecular crowding has been shown to drive the assembly of many kinds of macromolecular structures, such as actin filaments [8,9], spectrin [10], amyloids [11], viral capsids [12], and DNA polymerase [13], as well as numerous forms of protein folding and aggregation [14–16].

In the present work, we focus on modeling one specific difference between the intracellular environment and the assumptions of typical *in vitro* models: macromolecular crowding. Our long-term goal is to develop more realistic computational models of all of the ways in which the cell influences assembly chemistry. The experimental observations discussed above suggest that accurate models of crowding effects on assembly chemistry are a necessary, although likely far from sufficient, component to developing genuinely predictive models of assembly chemistry in the cell. By approaching this one issue in isolation, we hope to more precisely characterize the limits of our computer models in simulating assembly in crowded environments without having to consider confounding effects from other features of the cellular environment. Unfortunately, capturing overall assembly dynamics of some of the more important assembly systems, such as the actin or microtubule cytoskeletons or viral capsids, can require modeling of hundreds or thousands of components over time scales of seconds. Such simulations are computationally challenging for even the most coarse-grained models. Computational models for simulating assembly processes in crowded conditions are therefore far from achieving the necessary realism without requiring unrealistic computational resources [17–19]. Fast, space-free models—e.g., continuum mass action models and stochastic Gillespie models [20]—can handle large system sizes and complexities and can in principle be used to model assembly in crowded conditions through adjustments in reaction rates. The data needed to make these adjustments empirically are, however, not easily gathered and the corrections are difficult to make from first principles [17]. Various forms of Brownian dynamics models (see, for example, [21]) can directly represent crowded conditions, as well as many other features of the cellular environment, but only with computational costs too high for the time scales needed for the more challenging assembly systems. Discretized lattice Monte Carlo models [22–25] provide an intermediate between these two extremes, directly simulating crowding conditions at much lower cost than off-lattice spatial models. These discretized models, however, oversimplify the movement of each molecule in a lattice space and impose unrealistic constraints on particle diffusion and assembly structures. These models are thus likely to exaggerate crowding effects and have limited ability to model more complex assembly geometries.

In this paper, we examine the utility of Green's function reaction dynamics (GFRD) [26], a stochastic particle-based model recently developed by van Zon and ten Wolde, to capture the effects of cellular levels of crowding on assembly chemistry. GFRD uses an event-driven model to produce highly efficient off-lattice simulations of chemistry in crowded environments. For this reason, it shows promise as a means of developing more realistic models of large-scale reaction networks in the cell. GFRD has, for example, recently been adopted by several groups as a method for simulating generic cellular reaction and signaling networks [26–28]. GFRD's advantages in handling large numbers of particles and long time scales would seem to be particularly useful for modeling assembly chemistry. We have therefore developed a prototype assembly simulator using GFRD to implement a simple model of dimer assembly in a two-dimensional diffusive space, similar to what one would find at the leading edge of a cell. We then use a series of simulations to establish three key features of the model: that it efficiently scales to the system sizes needed for modeling cellular assembly reactions, that it provides reliable quantitative models of assembly under noncrowded conditions that match the behavior of simple mass-action models, and that it exhibits the expected effects of crowding under conditions of dense packing of reactant molecules or inert crowding agents. This work demonstrates the value of GFRD for overcoming some of the limitations of prior methods for modeling large-scale assembly reactions in the cell.

II. METHODS

For clarity, we briefly summarize the GFRD method. The reader is referred to the primary reference by von Zon and ten Wolde [26] for details on the method and the theory behind it. The method represents a set of diffusing particles, each of which is presumed to have a position in space at any given point in time. Collisions between particles provide opportunities for reaction events. The model is therefore similar to that assumed by a Brownian dynamics simulation. The key insight of GFRD is that we need not explicitly track particle positions at all points in time, as is done in Brownian dynamics. Rather, using a probabilistic model of the diffusion process, we can represent particle positions as probability distributions over the space that particles might occupy. We can then defer the actual computation of a particle's position until there is some appreciable probability that it is involved in a collision. Figure 1 illustrates the basic concept behind the GFRD method. If we have observed a particle at some position $x=0$ at time $t=0$, then we can derive its distribution of possible positions at some later time t' from a one-dimensional random walk model [29]. The resulting distribution of particle positions at time t' is the Gaussian

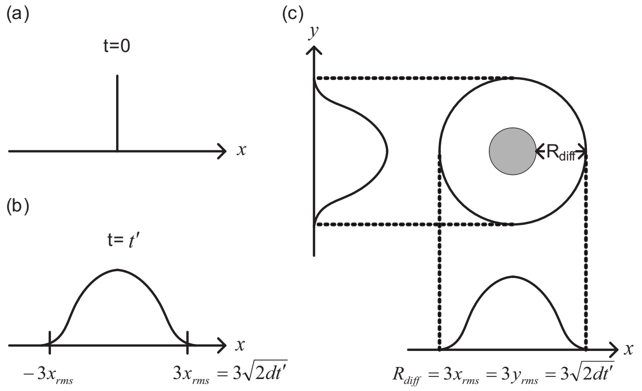
(a)
t=0
x
(b)
t= t'
x
$-3x_{rms}$
$3x_{rms} = 3\sqrt{2dt'}$
(c)
y
R_{diff}
x
$R_{diff} = 3x_{rms} = 3y_{rms} = 3\sqrt{2dt'}$

FIG. 1. Illustration of the GFRD model for diffusion in one (a),(b) and two dimensions (c). (a) Distribution of a particle's position at time $t=0$, when it has a fixed known position in space. (b) Distribution of positions at time t', described by a Gaussian centered on the initial position, where d is the diffusion coefficient. The diffusion limit region of this one-dimensional case is from $-3x_{\text{rms}}$ to $3x_{\text{rms}}$, which covers 99.73% of the full distribution of possible particle positions. (c) Diffusion limit circle for a two-dimensional model defined by a three standard deviation confidence interval from the Gaussian in each dimension. The radius of this limit circle (R_{diff}) is $3x_{\text{rms}}$, which covers 98.89% of the full distribution of possible particle positions.

$$P(x,t') = \sqrt{\frac{1}{4\pi dt'}} e^{-x^2/4dt'}, \tag{1}$$

where d is the diffusion coefficient in the random walk model, provided the particle has not been involved in any collisions between times $t=0$ and $t=t'$ [29]. The root mean square displacement of the particle in this one-dimensional case (x_{rms}) is given by $\sqrt{2dt'}$, which is equal to the standard deviation of the Gaussian distribution in Eq. (1). We can therefore place an outer bound on the possible positions of the particle with high confidence by defining a circle that encloses all but a small fraction of the Gaussian in each dimension. We call this outer bound a "diffusion limit circle." We only need to evaluate the position of a particle when its diffusion limit circle comes in contact with that of another particle, meaning that there is some appreciable probability that they have collided. A simulation with this model can proceed very efficiently by jumping between collisions of diffusion limit circles, as illustrated in Fig. 2. At the start of a simulation, we compute the times at which each pair of particles' diffusion limit circles will interact, ignoring all other particles. Each step of the simulation then consists of identifying two colliding diffusion limit circles, evaluating positions of the particles given their position distributions, testing for possible reaction events, and reevaluating collision times for those particles with each other or any others in the simulation.

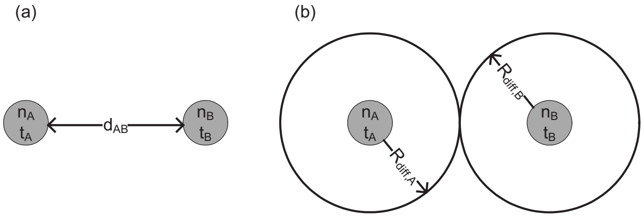

FIG. 2. Calculation of the collision time of two diffusion limit circles. (a) Two particles A and B initially separated by a distance d_{AB} at the original time. (b) The two particles at the time of their collision event (t'), defined as the time at which $d_{AB}=R_{\text{diff},A}(t'-t_A)+R_{\text{diff},B}(t'-t_B)$.

In order to implement a GFRD model, we first define the radius of a diffusion limit circle (R_{diff}), and then need a procedure to calculate collision times of diffusion limit circles. For the present simulator, R_{diff} is set to three times the standard deviation of the Gaussian distribution for a single coordinate in isolation. The square of the distance the particle has diffused in two dimensions can be expressed as the sum of the squares of two independent normal distributions, a quantity that is χ^2 distributed with two degrees of freedom. This concept is illustrated in Fig. 1. The probability the particle will be confined to the radius R_{diff} is then equivalent to the probability that the χ^2 random variable is within $(R_{\text{diff}})^2$, which is 98.89%. Therefore, the radius of a diffusion limit circle in two-dimensional (2D) space for a time duration (Δt) is defined as follows:

$$R_{\text{diff(2D)}}(\Delta t) = 3x_{\text{rms}} = 3y_{\text{rms}} = 3\sqrt{2d\Delta t}. \qquad (2)$$

We can calculate the collision time (t') of two diffusion limit circles, shown in Fig. 2, as follows:

$$\begin{aligned} d_{AB} &= R_{\text{diff},A}(t'-t_A) + R_{\text{diff},B}(t'-t_B) \\ &= 3\sqrt{2d\Delta t_A} + 3\sqrt{2d\Delta t_B} \\ &= 3\sqrt{2d(t'-t_A)} + 3\sqrt{2d(t'-t_B)}, \end{aligned} \qquad (3)$$

where d_{AB} is the distance between two particles A and B, and t_A and t_B are the times at which the positions of particles A and B were last determined. Eq. (3) can be analytically solved to determine the collision time of any two diffusion limit circles. The solution of Eq. (3) for the diffusion model assumed in the present work is given in Eqs. (11)–(13) later in this section.

For the simulations presented here, we used a reflective hard boundary, meant to better model chemistry within a single prokaryotic cell or an enclosed compartment of a eukaryotic cell. A reflective boundary has a simple implementation in the GFRD model: when new particle positions are sampled, any sampled position outside the boundary of the space in either dimension is reflected inside the boundary by the same distance by which it was initially outside the boundary. So, for example, a particle sampled at 1 nm past the boundary will be reflected to a position 1 nm inside the boundary. The GFRD method can alternatively be used to represent a periodic boundary condition, although we do not use that model in the present work.

Given a method for computing collision times, the core of a GFRD simulation consists of a discrete event loop that steps between collision events. We adapt a prior method for queue-based simulation of generic assembly systems [30] to this GFRD model. At each step of a simulation, we first check an event queue to identify the next potential collision event. We then sample new positions for particles involved in that next event and determine from the sampled positions whether a collision has in fact occurred. In the event of a collision, we consider possible reaction events, as discussed below, and implement the reaction, if any. Finally, we identify new collision events for the particles just processed and any reactants produced from their collision and add these new events to the queue, preparing us for the next round of simulation.

A. Adapting GFRD for assembly reactions

For an assembly system, we must consider two kinds of reactions: association and dissociation. Association reactions can occur when two particles with compatible binding sites collide. The forward reaction rate (K_+) of any given association reaction thus depends in part on the collision rate, and hence the diffusion model, and in part on the probability of binding upon collision. We model waiting times to dissociation events by an exponential distribution, which has a single parameter uniquely determined by K_-. In an uncrowded system, the dissociation reaction rate (K_-) is independent of the diffusion model. Each time an assembly is created, we sample a dissociation time for that assembly from the exponential distribution and place it in the event queue with the collision events.

For both reaction types, however, crowding may influence reaction rates by sterically hindering reactions. Figure 3 illustrates cases of possible steric hindrance that may occur on reaction events. In Fig. 3(a), two monomer reactants (M_1 and M_2) have come sufficiently close as to form a dimer. These particles may either bind to form a dimer (D_1) or sample new positions to model a collision that does not result in binding. The simulation uses a user supplied binding probability (B) to choose whether to bind M_1 and M_2 into D_1. Either event might, however, be blocked by steric collision. If binding is selected but the dimer would produce a steric clash with another particle, as in Fig. 3(a)(1), then the event is reversed and it is assumed the binding reaction was blocked by steric hindrance. There is also a small probability that sampling new positions may result in overlap with another particle's diffusion limit circle, as in Fig. 3(a)(2). In that case, as well, the event is reversed and it is assumed to have been hindered. Similarly, if a dissociation event rises to the top of the queue, as in Fig. 3(b), then the system would ordinarily split the D_1 into M_3 and M_4 and sample positions for M_3 and M_4 relative to the center of D_1. If that event produces no steric collisions [Fig. 3(b)(1)] then the simulator can sample new collision times for M_3 and M_4 and proceed. If, however, the separation would produce a steric clash [Fig. 3(b)(2)], then we presume that reaction to have been hindered, block the dissociation, and sample a new dissociation time for D_1. Crowded conditions thus have the potential to reduce effective rates of both forward and reverse reactions.

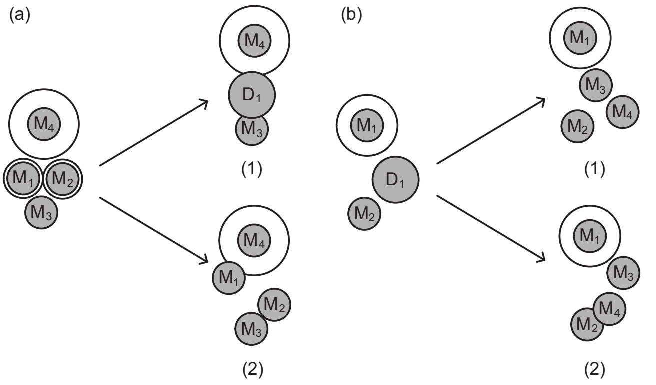

FIG. 3. Model of the excluded volume effect; M, monomer; D, dimer. (a) Excluded volume effects in the case of potential collision and binding events. When a binding event produces an overlap of particles, as in (1), that binding event is presumed to have been blocked by steric hindrance and is reversed. A collision resulting in resampling of positions without binding (2) can also be blocked by steric hindrance from surrounding particles (M_3 or M_4). (b) Excluded volume effects impeding dissociation. For a dissociation event to proceed, it must produce no overlap between particles, as in (1). When a dissociation event produces overlap of particles (2), that dissociation is presumed to have been hindered and is reversed by the simulator.

One important consideration in adapting GFRD to general assembly chemistry is accounting for the differential diffusion rates of complexes. We use a correction for diffusion of assembled structures based on the Stokes-Einstein diffusion theory. The Stokes-Einstein theory estimates the diffusion coefficient to be

$$d = \frac{kT}{6\pi\eta r}, \tag{4}$$

where k is Boltzmann's constant, T is the absolute temperature of the solution, η is the viscosity of the solute, and r is the radius of the spherical solute molecule [29]. We assume that all particles are spherical and placed on the plane, so that individual particles are considered as circles in our 2D model. The diffusion rate should thus vary inversely with the square root of particle radius, giving us a diffusion limit circle in the two-dimensional case of

$$R_{\mathrm{diff(2D)}} = 3\sqrt{\frac{kT}{3\pi\eta r}\Delta t}, \tag{5}$$

which is determined by substituting Eq. (4) into Eq. (2). Figure 4 provides pseudocode for our model of GFRD for generic assembly simulations.

B. Implementing a model dimer system

In order to validate this approach as a model of assembly in crowded environments, we chose to examine the simplest possible assembly system: a homodimer described by the reaction system

```
① Set the input parameters: the number of reactants, the number of inert particles, the
  radius of a monomer, the size of a square boundary, the binding probability, the
  mean time of the exponential distribution for breaking events, the threshold distance
  for a binding process, and the quasi steady state time to stop the simulation.
② Initialize all variables.
③ Set the positions of all particles, which are uniformly distributed in the area enclosed
  by the square boundary.
④ Calculate the next event time for each particle and then push this event into the
  priority queue.
⑤ Extract the minimum-time event from the priority queue, check the validity of the
  event, pick the new position from the truncated Gaussian distribution, and finally test
  the binding event.
⑥ Calculate the next event times for any particles affected by the previous event and
  push these events into the priority queue.
⑦ Go to step 5 unless some user-specified stopping condition has been reached.
```

FIG. 4. Pseudocode for GFRD-based simulation of generic assembly reactions.

$$M + M \underset{K_-}{\overset{K_+}{\rightleftarrows}} D.$$

We can define the diffusion limit radii for a dimer system in terms of the monomer radius r_1 and the dimer radius

$$r_2 = \sqrt{2\alpha} r_1. \tag{6}$$

α is the ratio of the area of the dimer to that of two monomers. The radii of the diffusion limit circle of a monomer and a dimer for a given time interval (Δt) are therefore as follows:

$$R_{\mathrm{diff(2D)monomer}} = 3\sqrt{\frac{kT}{3\pi\eta r_1}\Delta t} = C\sqrt{\Delta t}, \tag{7}$$

$$R_{\mathrm{diff(2D)dimer}} = 3\sqrt{\frac{kT}{3\pi\eta r_2}\Delta t} = 3\sqrt{\frac{kT}{3\pi\eta\sqrt{2\alpha}r_1}\Delta t} = C\frac{\sqrt{\Delta t}}{2^{1/4}\alpha^{1/4}}, \tag{8}$$

where

$$C = 3\sqrt{\frac{kT}{3\pi\eta r_1}}. \tag{9}$$

In the present work, we define $\alpha = 1$ in order to conserve the total excluded volume over time in each simulation. We will therefore omit α in subsequent formulas.

After substituting these equations into Eq. (3), the equation for calculating the collision time of two diffusion limit circles follows:

$$C\frac{\sqrt{\Delta t_A}}{n_A^{1/4}} + C\frac{\sqrt{\Delta t_B}}{n_B^{1/4}} = C\frac{\sqrt{t' - t_A}}{n_A^{1/4}} + C\frac{\sqrt{t' - t_B}}{n_B^{1/4}} = d_{AB}, \tag{10}$$

where n_A and n_B are the relative areas of particles A and B. The solution of Eq. (10) follows:

$$t' = \begin{cases} \dfrac{(d_{AB}/C)^2\sqrt{n}}{4} + \dfrac{t_A + t_B}{2} + \dfrac{(t_A - t_B)^2}{4(d_{AB}/C)^2\sqrt{n}} & \text{if } n_A = n_B = n, \\ \dfrac{-b - \sqrt{b^2 - 4ac}}{2a} & n_A \neq n_B, \end{cases} \tag{11}$$

TABLE I. Particle counts used to simulate variation of reactants (R) and inert crowding agents (I) to produce total concentrations from 10% to 42.5% on a 100×100 nm^2 simulation space.

Concentration (%)	10	12.5	15	17.5	20	22.5	25
R [Fig. 8(a)]	51	64	76	89	102	115	127
$R+I$ [Fig. 8(b)]	51+0	51+13	51+25	51+38	51+51	51+64	51+76
Concentration (%)	27.5	30	32.5	35	37.5	40	42.5
R [Fig. 8(a)]	140	153	166	178	191	204	216
$R+I$ [Fig. 8(b)]	51+89	51+102	51+115	51+127	51+140	51+153	51+165

where

$$a = (n_A^{-1/2} - n_B^{-1/2})^2,$$

$$b = -2\left(\frac{t_A}{n_A} + \frac{t_B}{n_B} - \frac{(t_A + t_B)}{\sqrt{n_A n_B}} + (d_{AB}/C)^2(n_A^{-1/2} + n_B^{-1/2})\right),$$

$$c = \left(\frac{t_A}{n_A^{1/2}} - \frac{t_B}{n_B^{1/2}}\right)^2 + 2(d_{AB}/C)^2\left(\frac{t_A}{n_A^{1/2}} + \frac{t_B}{n_B^{1/2}}\right) + (d_{AB}/C)^4. \tag{12}$$

In practice, we used a more complicated formulation that is mathematically equivalent in exact arithmetic but reduces round-off errors when dealing with small time differences. The formula solves for

$$\frac{\sqrt{A}}{n_A^{1/4}} + \frac{\sqrt{A-\Delta}}{n_B^{1/4}} = \frac{d_{AB}}{C},$$

where $A = t' - t_A$, $\Delta = t_B - t_A$, $A - \Delta = t' - t_A - t_B + t_A = t' - t_B$. This alternative formulation is solved by

$$t' = \left(\frac{d_1 - d_2\sqrt{1 + d_3 d_2^2 - d_1^2 d_3}}{(d_1^2 - d_2^2)}\right)^2 \left(\frac{d_{AB}}{C}\right)^2 + t_A, \tag{13}$$

where $d_1 = n_A^{-1/4}$, $d_2 = n_B^{-1/4}$,

$$d_3 = \frac{\Delta}{(d_{AB}/C)^2} = \frac{t_B - t_A}{(d_{AB}/C)^2}.$$

C. Simulation experiments

In order to validate the efficiency and correctness of the model, we ran a series of simulation experiments using variants on the dimer system. These experiments were designed to examine a range of conditions including systems with little or no crowding effect, as might be typical of an *in vitro* system, and those experiencing high levels of crowding as might be expected in the cellular environment. We use standard MKS units for all constants defined below, except where otherwise noted for clarity. The cytoplasmic viscosity of Swiss 3T3 fibroblasts and Madin-Darby canine kidney cells is slightly larger than the viscosity of water [31]. Therefore, we set the viscosity value in Eq. (9) as 0.0012 Pa s, which is 20% larger than the viscosity of water at 293.15 K [32]. The radius of a monomer, r_1, was set to 2.5 nm. The radius of a dimer, r_2, was set to $2.5 \times \sqrt{2}$ nm, in order to provide no net entropic favorability toward binding under high crowding conditions. The scaling constant C in Eq. (9) was 3.59×10^{-5} m s$^{-1/2}$. The binding probability B was set to 0.75 and the mean dissociation time of dimers, M, was set to 10^{-9} s for our all simulation cases. Finally, the threshold distance d, within which particles are allowed to bind, was set to 0.5 nm. The size of the simulation space and the initial concentrations of reactant monomers and inert particles varied from experiment to experiment, as described below.

We first sought to establish empirically the efficiency of the model, verifying an expected linear cost in system size per simulation step. We conducted simulations for eight particle counts—from 500 to 4000 in increments of 500 particles—at a fixed total concentration of 5%. We therefore varied the simulation space proportionally to particle count, with sizes of 443×443 nm^2 (500 particles), 627×627 nm^2 (1000 particles), and so on. All particles were reactants for this experiment, initially in monomer form. We ran ten independent simulations for each condition for a fixed simulation time of 5 μs. We estimated the order of run time using log-log plots of run time versus particle count and iterations versus particle count.

We next sought to establish that the model reasonably captures the transition from uncrowded to crowded conditions for simulations consisting only of reactant monomers. We used a fixed simulation space of 100×100 nm^2, varying the total concentration of initial monomers from 10% to 42.5% in units of 2.5%. These values corresponded to variations in the number of initial monomers from 51 to 216. Table I provides the initial monomer counts for each simulation. We recorded the numbers of monomers and dimers at intervals of 31.25 ns from 2.5 to 5 μs of simulation time. We empirically determined that 2.5 μs was sufficient for the system go to equilibrium and therefore treated all time points beyond 2.5 μs as equilibrium measurements. Each simulation was repeated 30 times. The equilibrium dimer concentration was estimated for each condition by the mean over the 30 trials and the 81 time points from 2.5 to 5 μs for each trial. Error bars were similarly calculated as the standard deviations over these same data points for each condition. For comparison, we also determined analytic dimer concentrations on the assumption of no crowding effects. This curve was determined by estimating k_{eq} from the 10% crowding simulation and solving for the equation $k_{\mathrm{eq}} = D/(M_0 - 2D)^2$, where D is the dimer concentration and M_0 is the initial monomer concentration.

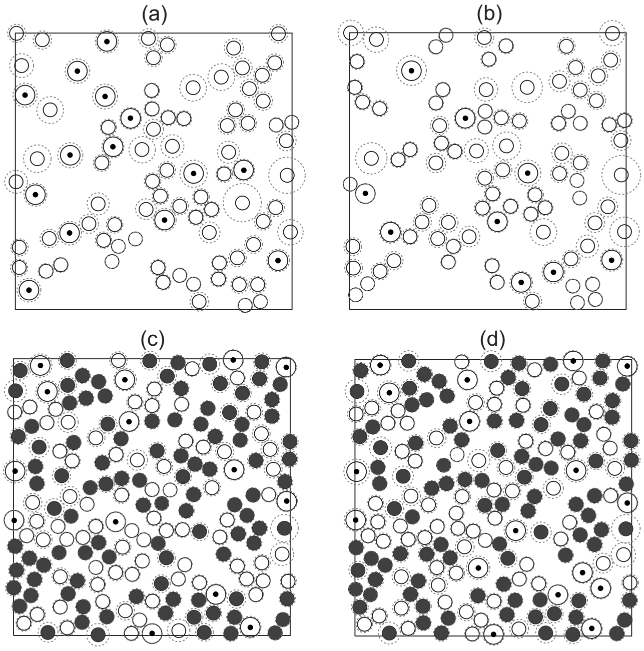

FIG. 5. Snapshots of simulator progress from two simulations with binding probability $B=0.75$, and mean time in exponential distribution for a breaking event $M=10^{-9}$ s. (a),(b) 20% reactant only; (c), (d) 20% reactant+20% inert particles, where open circles are reactants, closed (dark) circles are inert particles, the outer circles of particles are diffusion limit circles, and open circles with a dot in the center are dimers. Amount of simulated time as of the pictured state is (a) 4.5318, (b) 4.5346, (c) 4.9448, and (d) 4.9455 μs.

Finally, we sought to test the existence of a crowding effect in isolation by modeling reactions in the presence of varying concentrations of inert monomers. We used a fixed simulation space of 100×100 nm^2, and a fixed reactant monomer concentration of 10%. We then added varying amounts of inert particles to produce total occupied volumes of 10%–42.5% in increments of 2.5%. Table I provides initial counts of reactant monomers and inert monomers over these conditions. As with the pure reactant experiments, we completed 30 repetitions per parameter value, running each for 5 μs of simulation time and measuring progress in units of 31.25 ns. Equilibrium concentrations were again established by averaging over the 30 trials and 81 time points from 2.5 to 5 μs in each trial, and error bars were established by the standard deviations over these data points. For comparison, we also plot the constant dimer concentration we would observe on addition of inert particles under the assumption of no crowding effects.

III. RESULTS

Figure 5 shows screen snapshots illustrating the operation of our dimer model. Figures 5(a) and 5(b) show an example of a low-concentration simulation containing only reactant particles. The simulation space is 20% occupied, with a mixture of monomers and dimers. Pairs of concentric circles show the actual radii of the particles (inner circles) and their diffusion limit circles (outer circles) at a fixed point in time. Figures 5(c) and 5(d) show a similar system in a highly crowded condition, 40% total occupancy, through the addition of inert monomers (closed dark circles). The inert monomers also diffuse, and thus have diffusion limit circles, but are not capable of reacting. They would therefore be expected to have no effect on reaction progress under low-concentration conditions but to produce crowding effects under high-concentration conditions such as those in the figure. The temporally produced dimers in Fig. 5(a) are quickly dissociated in the low crowding condition, but the dimers in Fig. 5(c) are stabilized by additional crowding agents in the high crowding condition. Figure 6 shows an example of the reaction progress of the system for a single high-crowding parameter set (216 reactant monomers at 42.5% total concentration), averaged over 30 trials. The figure covers the time scale from 0 to 5 μs in Fig. 6(a), and shows the magnified region 0–3.75 ns in Fig. 6(b). The figure shows a rapid rise of dimers early in the simulation, settling into an equilibrium of monomers (dotted line) versus dimers (solid line) within about 1 μs. Error bars reveal that there is a moderate level of reaction noise at this scale, shown by variability from simulation to simulation. The level of noise appears essentially constant from the beginning of the simulation until equilibrium, suggesting that it results predominantly from a continuing random exchange of particles between monomer and dimer.

In order to validate the model, we first examined efficiency of the system by examining the scaling of run times for eight simulation sizes—from 500 particles to 4000 particles in increments of 500 particles—over a fixed simulation time of 5 μs. Figure 7(a) shows the run time versus system size, providing a clear linear fit on a log-log plot with a slope of 2.00. This slope reveals that the run time increases quadratically with the particle count for a fixed simulation time. Figure 7(b) shows the number of iterations (simulation steps) required to simulate 5 μs as a function of particle count, showing a linear dependence (slope 1.01). We can thus conclude that the run time per iteration grows linearly with particle count. Figure 7(c) confirms that the run time does indeed vary linearly with particle count, with an average run time of 275 ns per particle per iteration. The model therefore performs in accordance with theoretical models and appears scalable to handle systems of thousands of particles on time scales of microseconds, as we would require for many cellular assembly systems.

We next examined the behavior of the system in the absence of inert particles. Figure 8(a) shows equilibrium concentrations of dimers as a function of initial monomer concentration, where the simulation space is 100×100 nm^2, $B=0.75$, and $M=10^{-9}$ s. Table I provides the reactant monomer count for each simulation. The number of reactants varies from 51 at 10% concentration to 216 at 42.5% concentration. The average number of dimers, the products of the homodimerization reaction, increases as the concentration of reactants increases. The upper curve, marked with error bars, represents the simulation result of our model. The lower

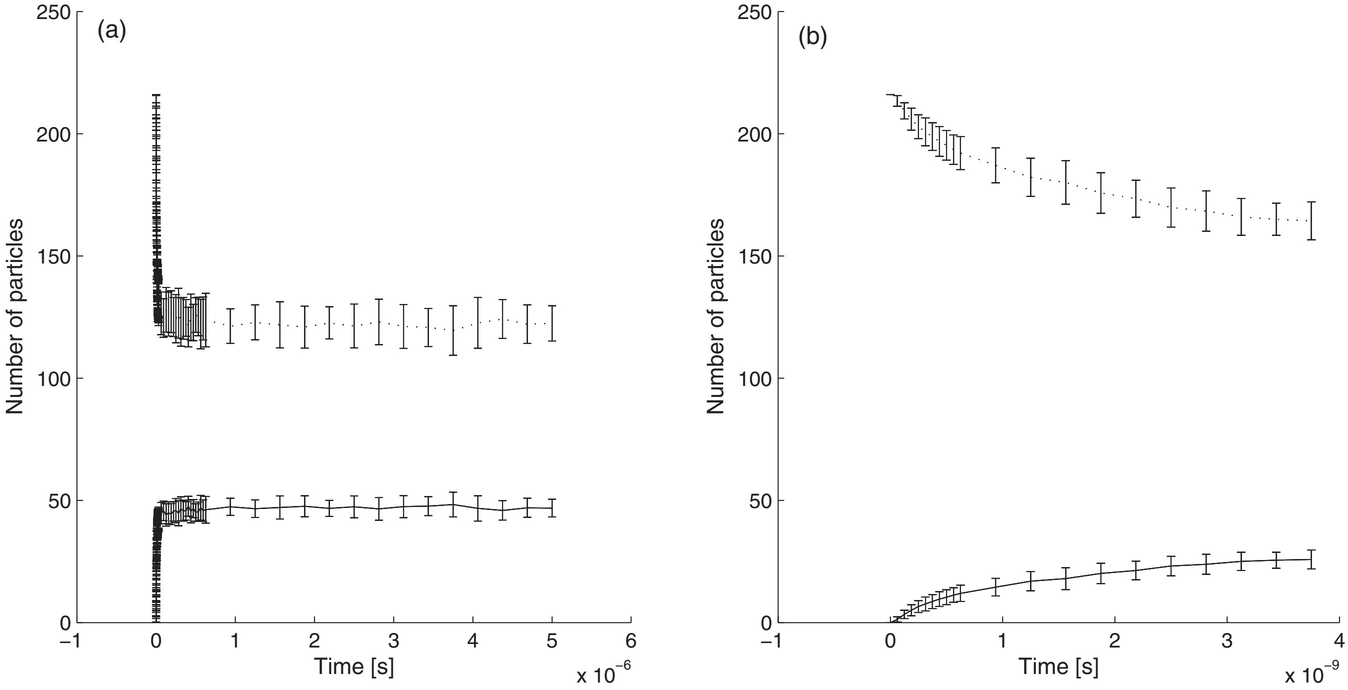

FIG. 6. Reaction progress versus time for a sample parameter set. The reaction was run with purely reactant monomers using $B=0.75$, $M=10^{-9}$ s, and total concentration 42.5%. Times (a) 0–5 and (b) 0–3.75 ns. The dotted line is the reaction progress for monomers, and the solid line is the reaction progress for dimers. Error bars show standard deviations across 30 simulations for each time point.

curve, marked with squares, represents an analytically determined equilibrium derived from the assumption of no crowding effect. The figure shows the dimer counts in higher-concentration cases in our model to be larger than the dimer counts calculated by the law of mass action, which means our model captures an increased crowding effect that the continuum mass-action model does not.

Figure 8(b) shows the results from simulations for testing the effects of varying inert particle concentration for a fixed 10% concentration of reactant particles. Table I shows the particle counts for each simulation. In the absence of any crowding effect, we would expect to observe a constant equilibrium dimer concentration independent of inert monomer concentration. The lower curve, marked with squares, illustrates the constant dimer count expected in the absence of crowding effects. The figure, by contrast, shows a steady increase in equilibrium dimer concentration as a function of inert monomer concentration. In particular, the simulations show an approximately 1.5-fold increase in equilibrium dimer concentration from 10% total crowding to 42.5% total crowding. Crowding in the simulations can slow association reactions through reduced diffusion rate, and steric hindrance from nearby particles can slow dissociation reactions through the excluded volume effect. The increase in polymerization

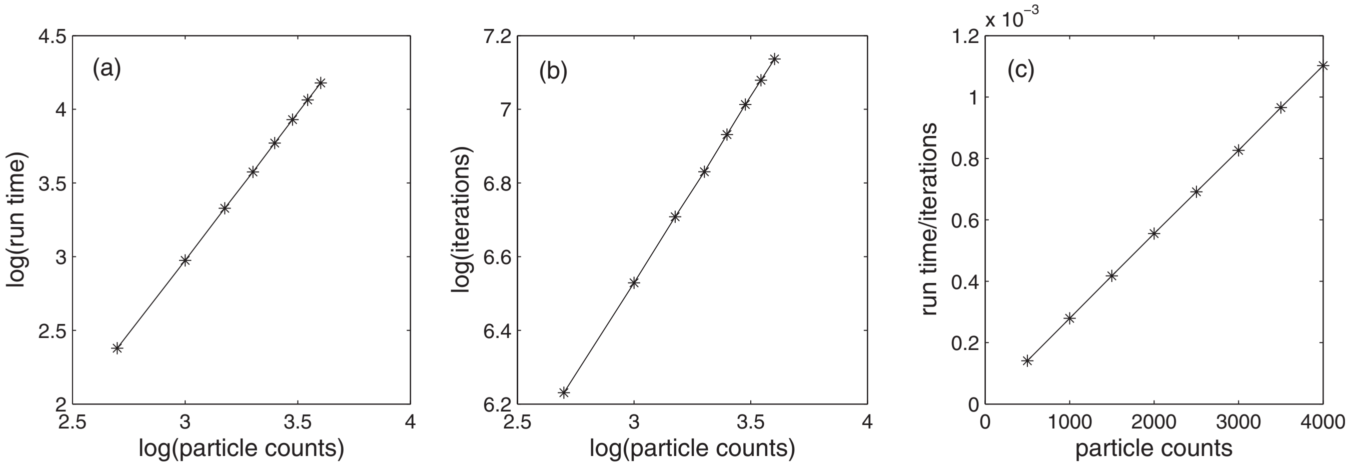

FIG. 7. Simulator efficiency in run time and iterations as functions of particle count. (a) $\log_{10}$-$\log_{10}$ plot of run time versus particle counts. (b) $\log_{10}$-$\log_{10}$ plot of iterations versus particle counts. (c) Run time per iteration versus particle counts. For all simulations described here, $B=0.75$, $M=10^{-9}$ s, and all particles are reactants.

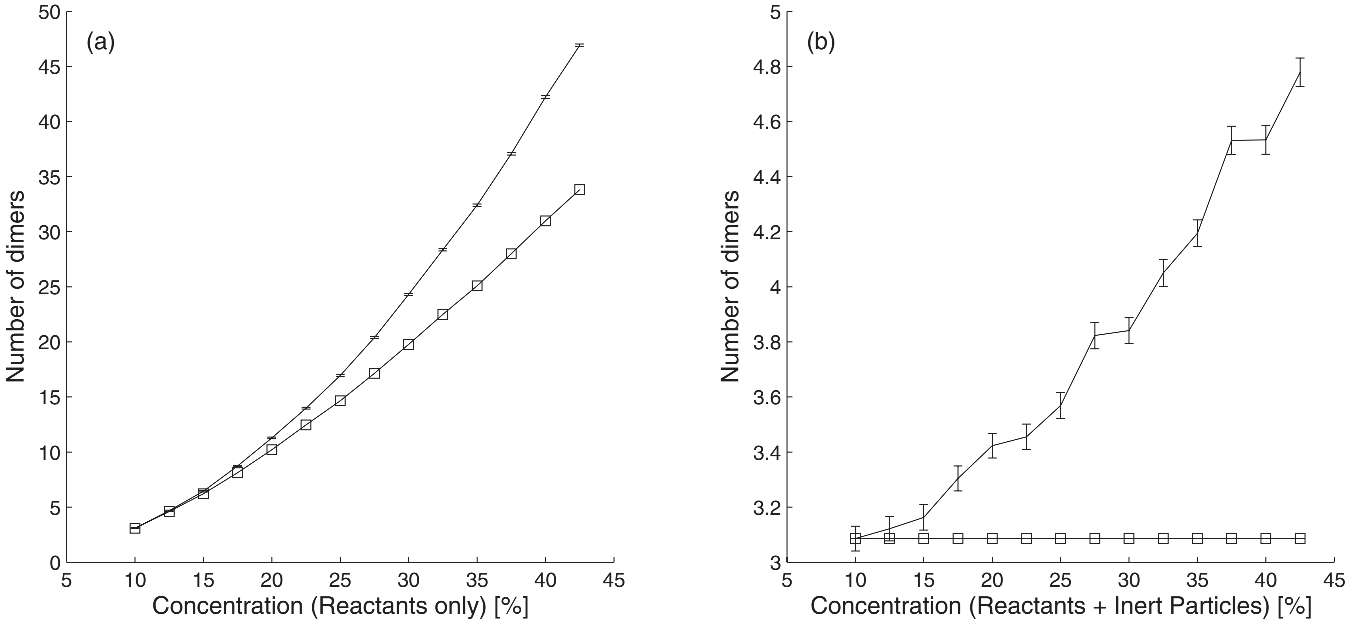

FIG. 8. Equilibrium dimer counts as functions of total crowding over a range of 10–42.5 % crowding. (a) Results of simulations using exclusively reactant monomers. (b) Results of simulations with fixed initial reactant monomer concentration 10% and varying inert monomer concentration. Each figure shows one curve derived from GFRD simulations with error bars showing the standard deviations across 30 simulations and 81 equilibrium time points. A second curve shows analytically determined dimer solutions assuming no crowding effect based on k_{eq} estimated from the 10% crowding simulations (squares). Both simulations used parameters $B=0.75$, $M=10^{-9}$ s.

observed here would be consistent with a reduction in the dissociation rate relative to the association rate. This result shows that the nonbinding particles likely play a crucial role in assembly kinetics, especially in high-concentration cases.

IV. DISCUSSION

We have constructed a prototype simulator to assess the ability of the recent GFRD method [26] to provide a computationally efficient model of assembly chemistry in cellular levels of crowding. We developed several extensions to the method designed to make it suitable for generic models of assembly chemistry and tested these extensions for a model dimer assembly system. A series of simulations with this model system confirm: that the model is efficient enough to model systems of several thousand reactant particles on time scales of microseconds, that it accurately models reaction kinetics under low-crowding conditions, and that it exhibits the expected macromolecular crowding effects under conditions of dense packing of either reactant monomers or inert crowding agents. These results support the contention that GFRD will provide a valuable supplement to existing computational tools for modeling assembly chemistry that is particularly well suited to modeling spatial and crowding effects difficult to examine in other comparably efficient models.

There are many avenues by which this work can be advanced to handle more challenging examples of assembly chemistry in crowded environments. Our model currently assumes spherical particles with spherical diffusion limit circles. It will be necessary to allow more complicated assembly geometries to simulate large structures, such as cytoskeletal elements or viral capsids. We can use for this purpose a hierarchical structure model similar to that used in prior Brownian [33] and stochastic simulation algorithm (SSA) based [34] simulators we have developed for studying viral capsid assembly. While one can continue to use spherical diffusion limit circles for arbitrary-shaped structures, efficiency is likely to demand tighter bounding curves accounting for both translational and rotational diffusion. One particularly important generalization for many such systems will be converting the model from two to three dimensions. While the two-dimensional model may be suitable for some forms of cellular assembly, such as assembly on membranes or at the leading edges of migrating cells, three dimensions will be more generally applicable. Converting from two to three dimensions requires only minor changes in the mathematics of the model, but it can be expected that larger numbers of particles and thus greater run times will be needed to develop realistic models of assembly in crowded three-dimensional spaces. Many other factors, such as distributions of inert particle sizes and realistic spatial models of the cell environment and its various immobile components, may also become important to models of such systems. Experimental validation is an important, but challenging, question. Simulations of this sort are needed because of the difficulty of experimentally monitoring large assembly reactions *in vivo*. Furthermore, analysis of *in vitro* systems in the presence of crowding agents, as has been used to probe crowding effects on many assembly reactions [1,2], could provide a partial solution. New technologies are likely to be needed, however, to definitively determine how closely this or other simulation methods approximate true *in vivo* assembly chemistry.

ACKNOWLEDGMENTS

We thank Navodit Misra for discussions of Eq. (13), and Zhanwu Liu for discussions of the χ^2 distribution. We thank Dr. Robert Murphy and the Center for Bioimage Informatics at Carnegie Mellon University for providing use of their computer cluster for this research. This work is supported by U.S. National Science Foundation Grant No. 0346981 and the Korea Science and Engineering Foundation (Grant No. M06-2004-000-10505).

[1] A. P. Minton, J. Cell. Sci. **119**, 2863 (2006).
[2] A. P. Minton, J. Biol. Chem. **276**, 10577 (2001).
[3] S. Zimmerman and S. Trach, J. Mol. Biol. **222**, 599 (1991).
[4] A. B. Fulton, Cell **30**, 345 (1982).
[5] K. Luby-Phelps, P. E. Castle, D. L. Taylor, and F. Lanni, Proc. Natl. Acad. Sci. U.S.A. **84**, 4910 (1987).
[6] L. Hou, F. Lanni and K. Luby-Phelps, Biophys. J. **58**, 31 (1990).
[7] N. Muramatsu and A. P. Minton, Proc. Natl. Acad. Sci. U.S.A. **85**, 2984 (1988).
[8] D. Drenckhahn and T. D. Pollard, J. Biol. Chem. **261**, 12754 (1986).
[9] R. A. Lindner and G. B. Ralston, Biophys. Chem. **66**, 57 (1997).
[10] R. Lindner and G. Ralston, Biophys. Chem. **57**, 15 (1995).
[11] D. M. Hatters, A. P. Minton, and G. J. Howlett, J. Biol. Chem. **277**, 7824 (2002).
[12] M. E. del Alamo, G. Rivas, and M. G. Mateu, J. Virol. **79**, 14271 (2005).
[13] S. B. Zimmerman and B. Harrison, Proc. Natl. Acad. Sci. U.S.A. **84**, 1871 (1987).
[14] A. P. Minton, Mol. Cell. Biochem. **55**, 119 (1983).
[15] A. P. Minton, Biophys. J. **78**, 101 (2000).
[16] M. S. Cheung, D. Klimov, and D. Thirumalai, Proc. Natl. Acad. Sci. U.S.A. **102**, 4753 (2005).
[17] P. R. LeDuc and R. Schwartz, Cell Biochem. Biophys. **48**, 16 (2007).
[18] K. Takahashi, S. N. Vel Arjunan, and M. Tomita, FEBS Lett. **579**, 1783 (2005).
[19] D. Ridgway, G. Broderick, and M. J. Ellison, Curr. Opin. Biotechnol. **17**, 493 (2006).
[20] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[21] A. H. Elcock, Proc. Natl. Acad. Sci. U.S.A. **100**, 2340 (2003).
[22] L. Apeltsin, K. Puskar, P. R. LeDuc, and R. Schwartz (unpublished).
[23] K. Puskar, S. Ta'asan, R. Schwartz, and P. R. LeDuc, Cell Biochem. Biophys. **45**, 195 (2006).
[24] K. Puskar, L. Apeltsin, S. Ta'asan, R. Schwartz, and P. R. LeDuc, Mech. Chem. Biosyst. **1**, 123 (2004).
[25] K. M. Puskar, A. Parisi-Amon, S. Ta'asan, R. Schwartz, and P. R. LeDuc, Phys. Rev. E **76**, 041904 (2007).
[26] J. S. von Zon and P. R. ten Wolde, J. Chem. Phys. **123**, 234910 (2005).
[27] J. S. van Zon, M. J. Morelli, S. Tanase-Nicola, and P. R. ten Wolde, Biophys. J. **91**, 4350 (2006).
[28] M. Dobrzynski, J. V. Rodriguez, J. A. Kaandorp, and J. G. Blom, Bioinformatics **23**, 1969 (2007).
[29] G. B. Benedek and F. M. H. Villars, *Physics with Illustrative Examples from Medicine and Biology: Statistical Physics*, 2nd ed. (Springer-Verlag, New York, 2000), Chap. 2.
[30] F. Jamalyaria, R. Rohlfs, and R. Schwartz, J. Comput. Phys. **204**, 100 (2005).
[31] S. Bicknese, N. Periasamy, S. B. Shohet, and A. S. Verkman, Biophys. J. **65**, 1272 (1993).
[32] J. V. Sengers and J. T. R. Watson, J. Phys. Chem. Ref. Data **15**, 1291 (1986).
[33] R. Schwartz, P. W. Shor, P. E. Prevelige, Jr., and P. W. Shor, Biophys. J. **75**, 2626 (1998).
[34] T. Zhang, R. Rohlfs, and R. Schwartz, in *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines (INFORMS, Orlando, FL, 2005), p. 2223.