First-Passage Kinetic Monte Carlo. I: Method and Basic Theory

Tomas Oppelstrup,[1,2] Vasily V. Bulatov,[1] Aleksandar Donev,[1]
Malvin H. Kalos,[1] George H. Gilmer,[1] and Babak Sadigh[1]

[1]*Lawrence Livermore National Laboratory, Livermore, California 94551, USA*

[2]*Royal Institute of Technology (KTH), Stockholm S-10044, Sweden*

Abstract

In this first part of a series of two papers we present a new efficient method for Monte Carlo simulations of diffusion-reaction processes. First introduced by us in [*Phys. Rev. Lett.*, 97:230602, 2006], the new algorithm skips the traditional small diffusion hops and propagates the diffusing particles over long distances through a sequence of super-hops, one particle at a time. By partitioning the simulation space into non-overlapping protecting domains each containing only one or two particles, the algorithm factorizes the N-body problem of collisions among multiple Brownian particles into a set of much simpler single-body and two-body problems. Efficient propagation of particles inside their protective domains is enabled through the use of time-dependent Green's functions (propagators) obtained as solutions for the first-passage statistics of random walks. The resulting Monte Carlo algorithm is event-driven and asynchronous; each Brownian particle propagates inside its own protective domain and on its own time clock. The algorithm reproduces the statistics of the underlying Monte-Carlo model exactly. Extensive numerical examples demonstrate that for an important class of diffusion-reaction models the new algorithm is efficient at low particle densities, where other existing algorithms slow down severely.

I. INTRODUCTION

Models in which the overall dynamics is represented by random walks are widely applied in science, engineering, medicine and finance. Probably the simplest example of a random walk is a sequence of steps taken randomly in two directions – left or right – along a line in one dimension. The object whose displacements follow such a sequence is referred to as a random walker or, simply, a walker. Of particular interest are diffusion-reaction systems in which multiple walkers walk simultaneously and independently and some significant events take place when two or more walkers find each other in space, or collide. Examples include formation and growth of aggregates of colloidal particles in suspensions, kinetics of aerosols in meteorology, diffusive phase transformations in solids [1], surface diffusion during crystal growth from vapor [2, 3], defect evolution in solids [4, 5], multi-particle diffusion-limited aggregation in physics, diffusion-controlled reactions in chemistry and biochemistry [6–8], population dynamics, quantum physics [9], and risk assessment and pricing models in finance to name a few. Numerical simulations of such processes often utilize various flavors of the Monte Carlo method.

Kinetic Monte Carlo (KMC) is a simple and robust computational approach for simulations of systems evolving through random walks. Mathematically, KMC derives from the theory of Markov processes in which the model evolves from state to state through a sequence of stochastic transitions whose rates depend on the current state alone. Random walks are typically simulated as sequences of hops, either from one lattice site to a neighboring one for discrete walks, or through finite displacements for continuum walks. When the system dynamics is defined by collisions among the walkers, the hops themselves are trivial changes of the system's state while significant events take place only when the walkers collide. A serious computational bottleneck is presented for the KMC method by situations when the density of walkers is low. Consider a system of randomly distributed walkers. It takes on average $\propto L^3$ hops for a walker to collide with another in $3d$ space (Here, L is the average spacing between the walkers expressed in the units of the lattice spacing or, in the continuum case, in the units of particle diameter). When L is large, it can take a great number of KMC cycles to evolve the model to a meaningful event, a collision. This is a serious drawback limiting applicability of the KMC method to diffusion-reaction simulations.

Several attempts have been made so far to overcome this notorious inefficiency in KMC

simulations. In [7], the equivalence between continuous random walks and diffusion is exploited by using the fundamental solution for the single particle diffusion to propagate the walkers over large distances. The JERK method [10] uses a known solution for the statistics of binary collisions between two diffusing particles to decide which of the $N(N-1)/2$ pairs of walkers should collide over the next time step [1]. These and similar methods achieve improvements in the efficiency of KMC simulations but at a cost of their accuracy. The fundamental difficulty that none of the mentioned methods can fully address is that statistics of collisions in the system of N walkers is an N-body problem. That is, the probability of collisions between, say, walkers 1 and 2, depends on all other $N-2$ walkers in the system. It is only in the limit of very small hops (in [7]) or vanishing time steps (in [10]) that such approximate methods become asymptotically exact. Unfortunately, in this same limit the mentioned methods lose their numerical efficiency.

Here we present a novel approach for KMC simulations that is both efficient and exact for a wide class of models involving collisions among multiple Brownian particles. Based on exact solutions for the first passage statistics of random walks, the new method is referred to as First Passage Kinetic Monte Carlo (FPKMC) in the following. In the new algorithm, rather than propagating the particles to collisions by small diffusional hops, the particles are propagated over long distances while each walker (particle) is protected within its own spatial region. The N regions are non-overlapping and partition the space into disjoint spatial domains in which the enclosed walkers are propagated individually. The use of first-passage statistics for walker propagation permits an elegant factorization of the N-body problem into a product of N single-body problems. Efficient implementation of the new method leads to an asynchronous event-driven algorithm [11] in which every walker propagates within its personal space and from its own time origin. The resulting speedup is most impressive when the density of diffusing particles is low and particle collisions are rare.

In this first part of a series of two papers we introduce the basic ideas behind the new FPKMC method and present a few simple but representative simulation tests on the method's accuracy and efficiency. The second part of this series presents a general implementation of the new method allowing simulations of complex reaction-diffusion processes in a vari-

[1] It was in fact the JERK method and its applications to modeling irradiated materials that inspired the key idea of the new algorithm presented in this paper.

ety of situations relevant in science and engineering. Part I is organized as follows. The next section introduces the basic ideas behind the FPKMC method using one-dimensional continuous random walks as a simple example. Section 3 describes the overall algorithm. In section 4 we focus on propagators, i.e., elementary solutions for first-passage statistics required for efficient propagation of multiple walkers to collisions, and describe extensions of the FPKMC algorithm to higher dimensions. Section 5 presents several computational experiments validating the new method's accuracy and efficiency. The results are summarized in Section 6. Appendix A describes a rejection sampling procedure used in the FPKMC algorithm and Appendix B contains a concise derivation of the propagators.

II. FIRST PASSAGE PROPAGATION IN $1d$

In this section we introduce the FPKMC algorithm using one-dimensional ($1d$) continuous time - continuous space random walks (diffusion) as an illustrative example. An extension to diffusion in dimensions higher than one will be described in section IV D. The definitions and the algorithms to be presented here remain essentially the same for discrete walks, but for discrete-valued space x and/or time t the integrals appearing in the discussion below correspond to sums over appropriate discrete values. We defer to a future publication algorithmic details specific to discrete random walks.

A. Single walker

To define the probability distributions to be employed in the FPKMC algorithm, let us first consider a single continuous random walk in one dimension ($1d$). Let x_0 and $t_0 = 0$ be the position and time origins of the walk and a be some other (barrier) position on the line $-\infty < x < \infty$. Through a sequence of random displacements the walker can at some future time reach the barrier position a for the first time. Similarly, for a closed interval $[a, b]$ such that $a < x_0 < b$, a first passage event occurs when the walker reaches either one of two barriers a or b. The theory of first passage processes [12] concerns itself with finding the probability that the walker will reach one of two barriers for the first time within time interval $[t, t + dt)$. The relevant statistical distribution is the probability density $c(x_0,x, t)$ to find the walker *surviving* at time t (having not reached either end of $[a, b]$) within infinitesimal

interval $[x, x+dx]$ inside $[a,b]$. By its definition, the integral $S(x_0,t) = \int_a^b dx' c(x_0,x',t)$ is the total probability for the walker to survive by time t regardless of its end position x. The splitting probability $j(a,t)$ is defined as the conditional probability that, given that the first-passage event occurs at time t, the walker reaches barrier a rather than b. For a random walk in $1d$, $j(a,t)+j(b,t)=1$ at all times. When the walk origin x_0 is exactly in the center of interval $[a,b]$, the splitting probabilities are equal $j(a,t)=j(b,t)=\frac{1}{2}$ and independent of the first passage time t. Finally, the no-passage (NP) PDF is defined as the conditional probability to find the walker at position x at time t, provided the first-passage event has not yet occurred, $g(x_0,x,t)=\frac{c(x_0,x,t)}{S(x_0,t)}$.

We defer to section IV the derivation of the probability distributions introduced above. For now let us simply assume that the functions $S(x_0,t)$, $j(a,t)$ and $g(x_o,x,t)$ are available and proceed to describe how they can be used to obtain statistical samples of random walks in various situations.

First consider the statistics of continuous random walks on the line $-\infty < x < \infty$ with no barriers. Assuming that the walks start at position $x_0=0$ and time $t_0=0$, the PDF of walker positions at time $t>0$ is given by the fundamental solution of the diffusion equation $c_\infty(x,t) = \frac{1}{\sqrt{4\pi t}}\exp(-\frac{x^2}{4t})$, where the effective diffusion coefficient is assumed to be unity. The same statistics can be obtained by randomly sampling from the first-passage (FP) and no-passage (NP) distribution functions as follows. Define an interval of length L_1 centered on the initial walker position $x_0=0$. Draw a random number ξ uniformly distributed on $[0,1]$ and solve $S(L_1,t_1)=\xi$ to sample the exit time t_1 out of interval $[-\frac{L_1}{2},\frac{L_1}{2}]$. If $t_1>t$, use the NP distribution $g_{L_1}(x,t)$ to sample the walker position inside the interval. If $t_1<t$, use another random number to sample at which end of interval L_1 the walker exits at time t_1. Define a new interval of length L_2 centered on the new walker position and sample a new time t_2 of first-passage out of interval L_2 using the survival probability distribution $S(L_2,t)$. Continue until the sum of first-passage times $T_k=\sum_{i=1}^k t_i$ exceeds t. Use the NP propagator $g_{L_k}(x,t-T_{k-1})$ to sample the end position of the walker. Proceeding in this manner, a random sample of the walker position for any time t is obtained through a sequence of $k\geq 0$ first-passage propagations ending in a single no-passage propagation. Repeating such stochastic sampling sequences many times, statistics of the end walker positions can be used to reproduce the fundamental solution $c_\infty(x,t)$ to any desired accuracy. The length L of the propagation intervals defines how many FP steps on average will be used to reach

time t but otherwise has no bearing on the resulting statistics.

The above example illustrates the use of NP and FP distributions for sampling random walks on $-\infty < x < \infty$. The resulting samples are statistically equivalent to the known distribution $c_\infty(x,t)$. While not necessary in this particularly simple case, the same sampling procedure based on the first-passage statistics can be effectively employed in considerably more complex situations, such as the one described below where the simple fundamental solution c_∞ no longer applies.

B. Multiple walkers

Consider now multiple objects moving randomly and simultaneously on a properly defined space and time. The walkers are assumed to walk independently of each other until two of them find themselves at a distance equal to or smaller than some interaction radius r (in $1d$ the interaction radius can be set to zero). As was discussed in the introduction, models of this kind represent a plethora of situations of practical interest. For our discussion here it is not necessary to define what specifically happens when the walkers reach the interaction radius; let us just assume that collisions somehow affect propagation statistics of two (or more) walkers involved in a collision.

The most straightforward numerical approach to modeling such systems is to use random numbers to move the walkers over space by small hops, one walker and one hop at a time, and checking after each such hop if any of the walkers have collided. Although widely used, this simple method is known to become less and less efficient with the decreasing density of walkers [13]. The idea of the method presented in this section is to circumvent the need for the numerous small hops by using the solutions for the first passage statistics of a single walker to efficiently bring the walkers to collisions.

Consider two simultaneous walks with the same time origin $t = 0$ but different position origins x_1 and x_2 in one dimension, such that $-\infty < x_1 < x_2 < \infty$. Is it possible to obtain statistics of collisions between two walkers using a sampling procedure similar to the sequence of FP and NP propagations described in the previous section? At a first glance, the answer should be negative because, in principle, a collision between the walkers can occur at any time (at least in the case of continuum diffusion), thus altering the statistics of both walkers. Hence, the simple solutions for first passage statistics of a single walker should no

longer apply. Fortunately, the trick of spatial protection enables the use of single-walker propagations.

Let us, at $t = 0$, surround the walkers by two non-overlapping segments L_1 and L_2 centered on the walk origins x_1 and x_2. For example, make the ends of two segments coincide at the mid-point between x_1 and x_2, as shown in Fig. 1. The key observation that enables the use of single-walker propagators is that, for as long as both walks remain inside their protective segments, there can be no interaction between the walkers. Hence, up until the time one of the walkers exits its protective segment, the statistics of the two walks remains independent of each other and the single walker propagators can be used. Let us now use the survival probabilities $S(L_1, t_1)$ and $S(L_2, t_2)$ to randomly sample the first-passage times t_1 and t_2 and find their minimum $t_{min} = \min\{t_1, t_2\}$. Say, $t_{min} = t_1$ which means that at time $t = t_1$ walker 1 reaches one of the ends of its protective segment L_1 while walker 2 remains inside its protective segment L_2. Let us randomly select to which end of its protective segment walker 1 propagates and advance the time clock by t_1, $t_c := t_1$. Since walker 2 has not exited its protective segment $L_{2,}$, its new position can be obtained by sampling from the NP distribution $g_{L_2}(x_{2,}t_c)$. Now the walkers find themselves in new positions x_1 and x_2 at time t_c, and the propagation cycle can be repeated: new protective segments L_1 and L_2 are defined around the new walker positions, two FP times are sampled and compared, the time clock is advanced and new walker positions are sampled from appropriate FP and NP distributions. As was first proposed in Ref. [9], such propagation cycles can be continued until two walkers collide.

Extension from two to N walkers is straightforward. One starts by defining non-overlapping protective segments $L_1, L_2, ..., L_N$ centered on each walker and sampling first passage times for every walker, $\{t_1, t_2, ..., t_N\}$. For as long as all walkers remain inside their protective segments, no walker can affect the statistics of any other walker. Therefore, the use of single walker propagators guarantees correct sampling of random walks at least until the next scheduled propagation at time $t_{min} = \min\{t_1, t_2, ..., t_N\}$. At this time, walker i with the shortest exit time $t_i = t_{min}$ is FP propagated to one of the ends of its protective segment and positions of all other $N - 1$ walkers are sampled from appropriate NP distributions (see Appendix Cfor additional details).

The sampling procedure described above allows a seemingly small but important modification: rather than canceling all exit times larger than t_m and NP-sampling new positions

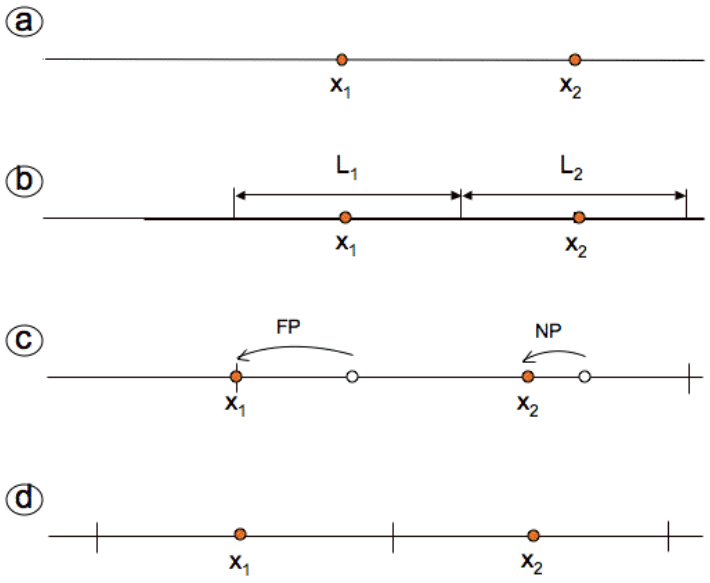

Figure 1: Simultaneous propagation of two walkers using first-passage and no-passage distributions. The filled circles are walker positions at different stages of propagation. (a) The walkers are initially at positions x_1 and x_2 at time $t = 0$. (b) The walkers are protected by two non-overlapping intervals L_1 and L_2 centered on the walker positions. (c) Random samples of two first passage times t_1 and t_2 are obtained and compared. Because $t_1 < t_2$, walker 1 is moved from its initial position (open circle) to one of the ends of its protective segment. At the same time, new position of walker 2 inside its segment L_2 is sampled from the NP distribution. Time advances to $t_c := t_1$. (d) A new propagation cycle begins by constructing non-overlapping protective segments around new walker positions. Although after the cycle illustrated by the (a)-(b)-(c) sequence the walkers find each other slightly farther apart, the next cycle is just as likely to bring them closer together.

for the corresponding $N - 1$ walkers, *all or almost all* of these $N - 1$ walkers can be left alone, protected inside their old segments and scheduled for propagation at their previously sampled exit times. First, the exit times sampled for all N walkers are arranged in a priority queue $t_i \leq t_k ... \leq t_m$. Second, walker i with the shortest exit time is FP propagated to one of the ends of its protective segment L_i. The NP propagation is only necessary if and when the segment end where walker i exits is also shared by a neighboring protective segment L_j.

When this happens, independence of two affected walks at later times is no longer assured because walker i now intrudes in the protective segment of walker j. The impasse is resolved by sampling a new position for walker j at time t_{min} using the NP distribution. Now that current positions of walkers i and j are decided, the time advances to t_c and walkers i and j are protected again in the space left available for them by all other $N-2$ protective segments. Two new exit times are sampled for walkers i and j, added to the current time and inserted in the queue. Preceding in this way, every cycle entails exactly one FP propagation and at most one NP propagation, rather than $N-1$ NP propagations as in the algorithm proposed in Ref. [9].

That it it is unnecessary to sample new exit times at $t = t_{min}$ for the walkers whose protective segments remain unaffected, follows directly from the basic property of the random walk as a memory-less stochastic process. In particular, for any $t_1 < t$ and $x \in [a, b]$ the Chapman-Kolmogorov-Smoluchowski identity holds

$$c(x_0, x, t) = \int_a^b dx' c(x_0, x', t_1) c(x', x, t - t_1).$$

Dividing both parts of the above equality by $S(x_0, t_1) = \int_a^b dx c(x_0, x, t)$ we obtain

$$\frac{c(x_0, x, t)}{S(x_0, t_1)} = \int_a^b dx' g(x_0, x', t_1) c(x', x, t - t_1).$$

The expression on the left hand side is the probability density at time t of walks that started at $t = 0$ and are known to have survived at time t_1. The expression on the right hand side defines the probability density of walks at time t that have survived to time t_1, when their positions x' inside the interval were sampled from the NP distribution $g(x_0, x', t_1)$, and the walk was then restarted from the new position origin x' and time origin t_1. Integration of both sides of the above equality over $\int_a^b dx$ yields the corresponding equality for the survival probabilities

$$\frac{S(x_0, t)}{S(x_0, t_1)} = \int_a^b dx' g(x_0, x', t_1) S(x', t - t_1).$$

The expression on the right is the probability to survive at time t for a walk that has survived at t_1 and whose new position x' inside the interval was sampled from the NP distribution

$g(x_0, x', t_1)$. The expression on the left is the probability to survive at time t for a walk that started at $t = 0$ and survived at t_1. The last two equalities mean that for all walkers whose protective intervals are unaffected by the FP propagation at t_1, there is no need to sample new positions and new exit times because the resulting distributions will be identical to the pre-sampled statistics. Thus, the two sampling procedures - one used in Ref. [9] and one proposed here - are statistically equivalent. Obviously, the new procedure is much preferred since the cost of its sampling cycle is not higher than the cost of the queue update, i.e., $O(\log N)$, whereas the cost of every sampling cycle in Ref. [9] is $O(N)$.

Just as in Ref. [9], in the new algorithm all N walks are initially protected and start from the same time origin $t_0 = 0$. The walker with the shortest exit time t_{min} is FP propagated and, perhaps, another neighboring walker is NP propagated to new positions. The global time clock advances to t_{min} and the affected walkers are protected by new segments. One or two new FP times are sampled, added to the new global time and inserted in the time queue. Over subsequent cycles, the time origins of the N protected walkers will gradually become desynchronized. Notwithstanding, statistical independence of protected walkers is guaranteed up to the shortest exit time in the current time queue. The resulting first-passage kinetic Monte Carlo (FPKMC) algorithm is asynchronous: every walker propagates within its personal space (protective segment) and from its own position and time origins. Sooner or later, a series of FP and NP propagations executed in this manner should bring a pair of walkers close to their interaction radius. Barring any inaccuracies in the propagators, the new algorithm is as exact as a Monte Carlo algorithm can be: for any number of walkers N, the statistics of simultaneous random walks with collisions is correctly reproduced in the limit of large number of independent Monte Carlo simulations.

Additional justification for the FPKMC algorithm is given in Appendix C.

III. THE FPKMC ALGORITHM

Here we give a brief description of the algorithmic components necessary for an FPKMC implementation.

First, one has to obtain first-passage (FP) and no-passage (NP) propagators. The NP propagators are needed when a walker propagates right on or close to the boundary of a neighboring protective segment. In such a case, the new protective segment for the just

propagated walker will have very small (or even zero) length. Consequently, the new time for FP propagation of the "squeezed" walker is likely to be so short that the same walker will be selected for the very next propagation again. Because its protection is tightly constrained by the protective segments of its neighbor walkers, the squeezed walker would continue to perform a series of very short time FP propagations resulting in little and, eventually, no advance of the global clock. The solution to this inefficiency of the *Achilles and Turtle* type is to identify which neighbor (or neighbors) limit the space available for protection of the squeezed walker. Then, NP propagation of the constraining inactive neighbor walker(s) typically results in more equitable partitioning of space with the squeezed walker. This is achieved at the cost of canceling the earlier scheduled FP propagation of the constraining walker(s) and propagating it (them) using the NP propagator to the current time. Proceeding in this manner, every Monte Carlo cycle entails one FP propagation and, possibly, one or few NP propagation, while all other $N-2$ walkers stay inactive, scheduled for propagations at their own times in the future. Calculation and efficient use of FP and NP propagators are discussed in the next section and, in more detail, in the Appendices.

In FPKMC, FP and NP propagations replace numerous short diffusive hops. At the same time, much of the computational effort is shifted to maintaining efficient space partitioning among the protective domains of the walkers. It is useful to observe that for the FPKMC method to work, space can be partitioned in an arbitrary manner for as long as the protective domains remain non-overlapping. One can use this freedom to simplify implementation and to maximize computational efficiency. To minimize implementation effort, in paper I we use particularly simple protective domains, i.e. centered segments (in $1d$) and hyper-cubes (in dimensions $d \geq 2$). Protection by spheres is developed in paper II.

Generally, one would want to partition space for maximum computational efficiency, for example to maximize the expectation time of the next FP propagation event. Globally optimal space partitioning for arbitrary positions of the N walkers is possible and can be accomplished in $O(N)$ operations - we do it only sparingly, such as in the beginning of each simulation run. During the run, the conditions of space sharing are inspected only for the walkers that were just propagated, and their neighbors. Definition of optimal space partitioning depends on the relative mobilities of the walkers. In paper I we only consider models in which all N walkers have exactly the same mobility properties, i.e., the same diffusion coefficient for the case of continuum random walks or the same hopping rates for

the case of discrete walks. Paper II describes the space partioning algorithms in significant detail, including also the case when some walkers are much more mobile than others.

The FPKMC algorithm allows exact and efficient treatment of particle collisions, by protecting and propagating groups of walkers, e.g. pairs. As we show in the next section, multi-particle propagators needed for such an extension are particularly simple in $1d$ and for hypercube-shaped walkers for $d \geq 2$. The evolution of various diffusion-reaction models is defined by what actually happens to colliding walkers - annihilation, coalescence, reflection, etc. - and how the collisions are handled. The same details will obviously affect the efficiency of FPKMC simulations but, in our view, the ultimate purpose of the FPKMC method is to enable efficient propagation of walkers to collisions whereas handling of collisions events is outside of the method's main scope. Thus, FPKMC can be viewed as a universal accelerator for particle diffusion or random walks by which the particles or walkers are brought to or *close to collisions*. We leave it for the method's users to define collision outcomes and to develop accurate and efficient methods for collision handling. To keep it simple, in paper I we consider only annihilation and coalescence reactions leaving more complicated collision scenarios for paper II.

In summary, the FPKMC algorithm entails the following steps:

1. Set the global time clock to zero. Construct non-overlapping protective domains around all walkers - use individual protection for single walkers and group protection for close pairs, as seems most efficient.

2. Sample an exit time for each domain (in the case of protected pairs this can mean a scheduled collision). Put the sampled event times in an event queue (e.g., implemented as a heap), so that the shortest time can be efficiently found.

3. Find the shortest exit time and identify the corresponding walker and domain. Sample the exit position for the selected walker. If the new position corresponds to a collision, take appropriate action.

4. Check if any of the existing protective domains are close to the new position of the particle. If necessary to make more space available for protection of the propagated particle, use no-passage propagators to sample new locations for the particles in the neighboring domains.

5. Construct new protective domains for all particles that changed their positions in steps (3) or (4).

6. Sample new event times for the particle(s) protected in step (5), as in step (2).

7. Insert the new event time(s) into the event queue. Go to step (3).

IV. THE PROPAGATORS

The FPKMC algorithm relies on the first-passage (FP) and no-passage (NP) propagators to skip the numerous small steps and to propagate the walkers to collisions. For the new algorithm to be efficient, Monte Carlo sampling from these propagators should not entail significant computational overhead. In this section, the elementary mathematical theory behind the propagators is presented along with explicit propagator formulas for the case of continuous diffusion in $1d$. We also discuss two methods appropriate for Monte Carlo sampling from the FP and NP propagators. Propagators suitable for efficient FPKMC simulations of N simultaneous random walkers on lattices will be given in future publications.

A. Continuous space - continuous time

The problem we consider here is to find when and where a particle performing a random walk exits a specified domain. Figure 2 shows a schematic of the setting.

Statistics of continuum random walks is equivalent to diffusion. Consider a very large ensemble of non-interacting random walkers starting simultaneously from the same origin. The concentration of walkers in this ensemble is the solution to the diffusion equation with a delta function at the walk origin as the initial condition

$$D\Delta c(\bar{x},t) = \frac{\partial c(\bar{x},t)}{\partial t}, \quad c(\partial\Omega,t) = 0, \quad c(\bar{x},0) = \delta(\bar{x}-\bar{x}_0).$$

Here, D is the diffusion coefficient, $c(\bar{x},t)$ is the probability density of finding the diffusing particle in an infinitesimal volume around $\bar{x}$ at time t given that it started at $\bar{x}_0$ at time $t = 0$.

The survival probability $S(t)$ is defined as the probability that at time t the particle has still not crossed the boundary of Ω. $S(t)$ can be found by integrating c over Ω, or by

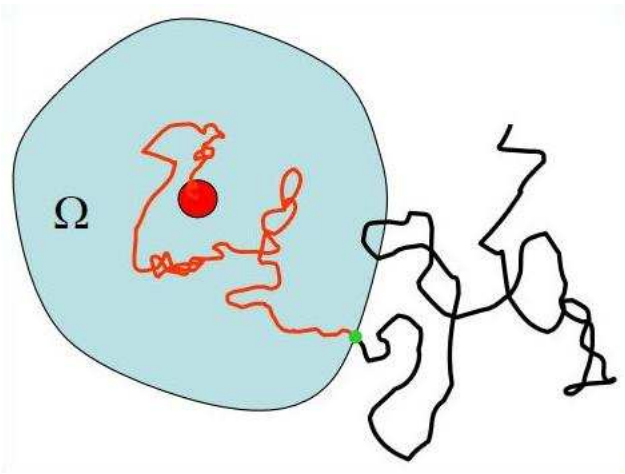

Figure 2: Consider a random walk and a domain Ω enclosing the walk origin. The walk can be thought to consist of two parts: the part shown in red that is entirely contained inside Ω, and the rest of the walk shown in black.

integrating the probability flux $(\nabla c \cdot \hat{n})$ out of Ω,

$$S(t) = \int_\Omega c(\bar{x}, t) d\bar{x} = 1 - \int_0^t \int_{\partial\Omega} \nabla c(\bar{x}, \tau) \cdot \hat{n} | d\bar{S}| d\tau,$$

where dS is the element of the surface area of $\delta\Omega$. Conversely, the exit probability per unit time (exit current) is:

$$p(t) = -\int_{\partial\Omega} \nabla c(\bar{x}, \tau) \cdot \hat{n} | d\bar{S}| = -\frac{\partial}{\partial t} \int_\Omega c(\bar{x}, t) d\bar{x}. \tag{1}$$

The above boundary and volume integral expressions are equal by the Gauss's theorem and the diffusion equation. The probability density for the exit location on $\delta\Omega$, i.e. the splitting probability is:

$$j(\bar{x}, t) = \frac{\nabla c(\bar{x}, t) \cdot n_{\bar{x}}}{-p(t)}, \quad \bar{x} \in \partial\Omega. \tag{2}$$

The first-passage (FP) propagation consists of sampling from the exit-time probability $p(t)$ and the splitting probability $j(\bar{x}, t)$. The no-passage (NP) propagation entails sampling from the probability density to find the particle near $\bar{x}$ at time t under the condition that the particle has not exited Ω by time t:

$$g(\bar{x}, t) = \frac{c(\bar{x}, t)}{S(t)}. \tag{3}$$

B. Propagators on a segment in *1d*

In $1d$, each protective domain is a line segment of length L. After translation and expressing the particle position in the units of L and expressing time in the units of L^2/D, particle diffusion on segment $[a, b]$ is described by the following equation on $[0, 1]$:

$$\frac{\partial^2 c}{\partial x^2} = \frac{\partial c}{\partial t}$$

with the boundary conditions $c(0, t) = c(1, t) = 0$ and the initial condition $c(x, 0) = \delta(x - x_0)$, where x_0 is the initial position of the particle. The solution can be written as the eigenfunction expansion,

$$c(x, t) = 2 \sum_{k=1}^{\infty} \sin(k\pi x) \sin(k\pi x_0) e^{-k^2\pi^2 t}. \tag{4}$$

This series converges quickly for $t \gtrsim \frac{1}{\pi^2}$. An alternative is to take advantage of the fundamental solution (Gaussian) and express $c(\bar{x}, t)$ as a sum of properly shifted images with alternating positive and negative signs, as shown in Fig. 3,

$$c(x, t) = \frac{1}{\sqrt{4\pi t}} \sum_{k=-\infty}^{\infty} (-1)^k \exp\left\{ -\frac{\left[x - \left(k + \frac{1}{2} + (-1)^k (x_0 - \frac{1}{2})\right)\right]^2}{4t} \right\}. \tag{5}$$

This expression converges quickly when $t \lesssim \frac{1}{4}$. The needed FP and NP propagators can be now obtained by substituting either of these two solutions into Eqs. (1), (2) and (3). Further technical details can be found in Appendix B.

C. Pair propagators in *1d*

In order to make the FPKMC algorithm more efficient and to enable exact sampling of particle collisions, it is necessary to be able to protect particle in pairs. Appropriate FP and NP propagators for the protected pairs should allow correct sampling of the collision time and particle positions. Consider two particles at x and y on a unit segment $[0, 1]$ so that $0 < x < y < 1$. To obtain the propagators one can solve the following two-dimensional diffusion problem

$$\left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right) = \frac{\partial c}{\partial t},$$

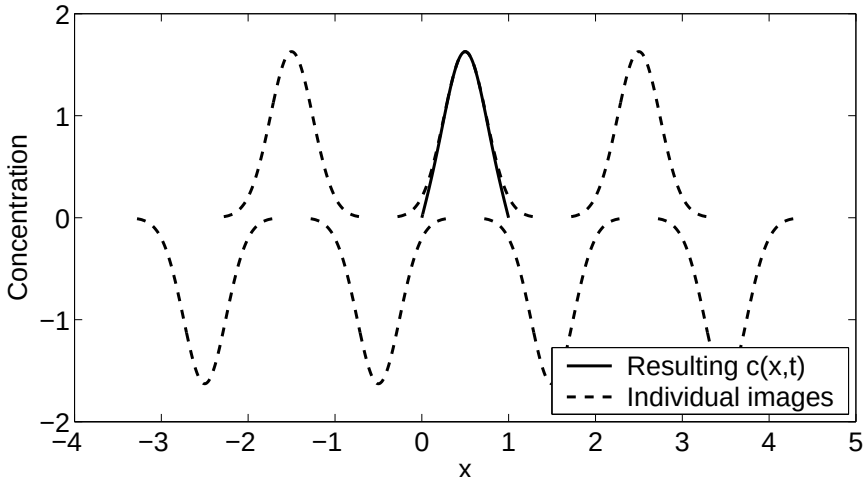

Figure 3: The solution for the probability density $c(x,t)$ is obtained by summing the images of the fundamental (Gaussian) solution placed at appropriate positions along the x axis. The individual images are shown as dashed lines and the solid line is the resulting solution.

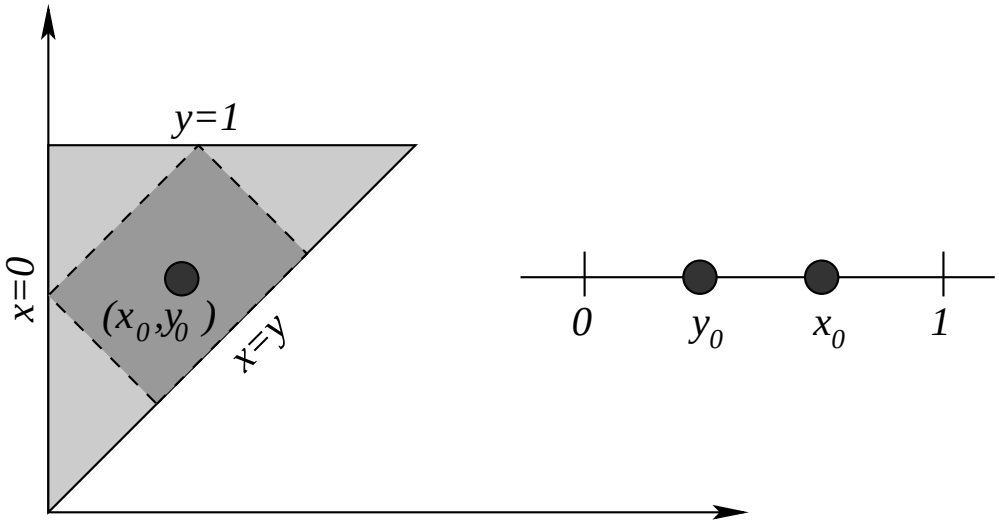

Figure 4: Two particles on a line protected by a single domain $[0,1]$. The edges on the triangle are Dirichlet boundaries for the diffusion problem, and correspond to the three possible event outcomes: x exits to the left, y exits to the right, and x and y collide.

with the boundary conditions $c(t, x = 0, y) = c(t, x, y = 1) = c(t, x, x) = 0$ and the initial condition $c(0, x, y) = \delta(x - x_0, y - y_0)$. This diffusion equation has to be solved on the triangle shown in Fig. 4. Absorption of the pair at the boundary $x = y$ corresponds to a collision between the two particles.

Rather than solving this two-dimensional problem, we note that there is a simpler diffusion

problem whose solution may allow us to propagate the pair almost as efficiently. Namely, we can use the solution of the same equation on any domain that is entirely contained inside the triangle. To retain the ability to sample collisions, a finite fraction of the $x = y$ line should be included in the new domain boundary. With this in mind, let us introduce new variables for the center of mass $u = \frac{1}{\sqrt{2}}(x + y)$ and the difference $v = \frac{1}{\sqrt{2}}(y - x)$ and define the new domain as the maximal rectangle that can be inscribed in the triangle so that one of its sides coincides with the collision line $v = 0$ (that is $x = y$). In these new coordinates, the two-dimensional diffusion problem on the rectangle separates into two *1d* problems, one for u and one for v, with the absorbing boundary conditions on all rectangle sides. This is convenient since one can use the FP and NP propagators already derived for a unit segment in *1d*. First, one samples two exit times, one for "walker" u and another for "walker" v. The exit time out of the inscribed rectangle is the shorter of the two. The exit coordinates are sampled using the splitting FP probability j for the "walker" whose exit time is shorter and using the NP propagator for the other "walker". The two particles collide when "walker" v exits to $v = 0$. All other outcomes correspond to pair propagation. Note that this algorithm preserves exact statistics of diffusive propagation and collisions of the protected pair. The sampled time increments are somewhat smaller on average than could be achieved by using the full triangular domain, which is an acceptable cost to pay for eliminating the need to compute and sample from the more complicated *2d* propagators on the triangular domain.

D. Generalization to higher dimensions

For the case of isotropic diffusion in any number of dimensions m, the use of hyper-rectangles or hyper-cubes for protecting the particles is convenient because the diffusion equation separates into m one-dimensional diffusion equations, one for each Cartesian direction. The same holds for anisotropic diffusion provided the edges of the protective hyper-rectangles are oriented along the principal axes of the diffusion tensor. In both cases the FP and NP probability distributions for m dimensions are the products of m one-dimensional distributions. Therefore, one can use the one-dimensional propagators to sample time and location of exit out of the protective hyper-rectangle. To do this, m exit times are sampled from the corresponding m one-dimensional FP propagators and the shortest of them, say t_k, is taken as a sample of the exit time. Then the splitting probability function $j(t_p)$ is used

to sample the exit location for the Cartesian direction k and the NP distributions at t_k are sampled to obtain the walker exit position for the remaining $m-1$ Cartesian directions.

A similar method can be used to propagate protected pairs and possibly larger groups of particles. For example, for the case of a pair of square-shaped particles protected by a square in $2d$, the change of variables is used to transform the problem to diffusion on two rectangles, one for each Cartesian coordinate. Accordingly, four one-dimensional FP propagators are used to sample an exit time t_p and an exit dimension k in the transformed coordinates. Then, three NP propagators are used to sample particle positions at the exit time t_p in the three other dimensions. Using the splitting probability function $j(t_p)$ one decides if the sampled exit indicates a collision along one of the two Cartesian directions. If so, whether or not the pair has actually collided is determined by the particle separation along the other Cartesian direction. For a collision to occur, the latter should be smaller than the sum of the half-widths of the square-shaped particles.

E. Sampling

Given a random number r uniformly distributed on $r \in [0,1)$, a sample first passage time t_p is obtained by solving $S(t_p) = r$ or simply as $t_p = S^{-1}(r)$, where $S(t)$ is the survival probability function on domain Ω. A exit location sample $x \in \delta\Omega$ can be obtained from the splitting probability density $j(\boldsymbol{x}, t_p)$. Splitting probabilities for exit to the ends of a segment in *1d* are given by two numbers $j_1(t_p)$ and $j_2(t_p)$, $j_1(t_p)+j_2(t_p) = 1$. When the initial position of the walker is at the center of a protective segment, the splitting probabilities are equal and independent of the first passage time, $j_1 = j_2 = \frac{1}{2}$. Thus, using protective segments (or hyper-rectangles) concentric with the initial particle positions is particularly convenient. Given a NP probability density function $g(x,t)$, a sample of the no-passage position x_{np} at time t' can be obtained by solving $G(x_{np};t') = r$, or $G^{-1}(r;t') = x_{np}$, where r is a random number uniformly distributed on $r \in [0,1)$, $G(x_{np};t') = \int_{x_1}^{x_{np}} g(x,t')dx$ is the cumulative NP distribution function at time t'.

Even for the simple case of diffusion on a *1d* segment with two absorbing ends, no closed form solution exist and the FP and NP propagators are available only in the form of series expansions. We have implemented and tested two techniques for sampling FP and NP propagators, both taking advantage of the fast convergence of the expansion series. The

first technique uses pre-tabulated propagators or, rather, appropriate inverse functions for fast lookup and interpolation. This method is particularly simple for the case of continuous diffusion on a segment in *1d* because the propagators can be stored as one-dimensional (FP) or two-dimensional (NP) tables (the time and position variables in the propagator tables are stored in the units of L^2/D and L, respectively).

The second technique is rejection sampling that relies on a converging series of upper and lower bounds to exactly sample from the FP and NP distribution density functions at a the cost of an occasional rejection. The needed series of bounds is obtained by integrating the series expressions (4) and (5) and observing that the terms in the resulting series solutions for the propagators have alternating signs and absolute values that monotonically decrease with the increasing term order. Thus, subsequent partial sums of the alternating series present an alternating sequence of increasingly tight upper and lower bounds. Taking advantage of particularly fast convergence of series (5) and (4) for short ($t \lesssim \frac{1}{4}$) and long ($t \gtrsim \frac{1}{\pi^2}$) times, respectively, it is possible to construct very tight bounds to the exact distribution functions. As a result, rejection sampling is efficient because rejections are infrequent and it rarely takes more than two bound evaluations to accept or reject a sample. This sampling technique is especially useful when it is difficult or impossible to evaluate and/or invert the cumulative probability distribution functions. However, when the inverse distributions are available, such as in the form of look-up tables, we found both table lookup and rejection techniques similarly efficient.

Further technical details on computing and sampling the $1d$ propagators are given in the Appendices.

V. COMPUTATIONAL TESTS ON ACCURACY AND EFFICIENCY

In this section we apply the FPKMC algorithm to several simple problems as a way to validate the method and compare its efficiency to traditional algorithms. The first test is for one-species coalescence on a discrete lattice in $1d$, the second for one species annihilation in $3d$, and the last is for two-species annihilation in $3d$.

A. Irreversible coalescence on a $1d$ lattice

Our first test of the FPKMC algorithm is simulation of a diffusion-controlled reaction of irreversible coalescence $A + A \Rightarrow A$ on a discrete lattice in *1d*. On every MC step every particle hops to the right or to the left with probability $\frac{1}{2}$ and, whenever two particles are found to occupy the same lattice site, one of them is eliminated. To obtain the needed FP and NP propagators for this discrete space-discrete time model it is necessary to solve for the PDF of a simple unbiased random walk of length n on an interval with two absorbing ends (barriers) at positions 0 and l (here n and l are positive integers). Similar to the continuous case discussed in the preceding section, the PDF for the random walk can be obtained as an expansion over appropriate eigen-functions or as a sum over image solutions. In the limit of $n, l \to \infty$, the PDF and the propagators asymptotically converge to the solutions for the continuum diffusion in *1d* [2].

We performed this simulation in two different ways: (1) using standard hopping [14] and (2) using the FPKMC algorithm in which on every MC step only one particle is advanced to the boundary of its protective domain (a segment in $1d$). In both series of simulations periodic boundary conditions were applied. Plotted in Fig. 5 are the simulated kinetics of this diffusion-controlled reaction averaged over 70 independent realizations. The two kinetics agree within small statistical errors. At the same time, in each simulation using single hops it takes on average $5 \cdot 10^9$ particle hops to reach the end of this reaction (i.e., annihilation of all but one particles) whereas the new algorithm accomplishes the same in just $4 \cdot 10^4$ steps (on average).

B. Annihilation in $3d$ continuum space

The next test is a simulation of a diffusion-controlled reaction of particle annihilation $A + A \Rightarrow 0$ but this time in continuous space and continuous time in $3d$. Figure 6 compares the annihilation kinetics simulated using the FPKMC method and a standard KMC algorithm in which the particles are propagated by small hops [4, 5, 15]. Each simulation starts with 8000 cube-shaped particles occupying initially a volume fraction of 0.004 of the simulation

[2] The FP and NP propagators for discrete random walks on lattices in $1d$ and in higher dimensions will be presented in future publications.

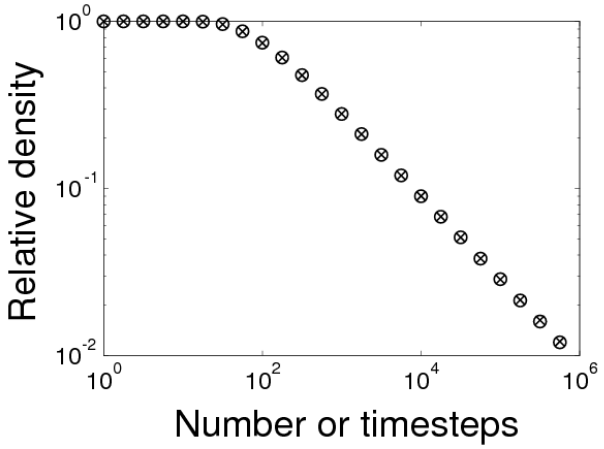

Figure 5: Comparison between the standard KMC calculations (crosses) and the first-passage KMC (circles) results for of $A+A \rightarrow A$ coalescence reaction by discrete-space discrete-time random walks in *1d*.

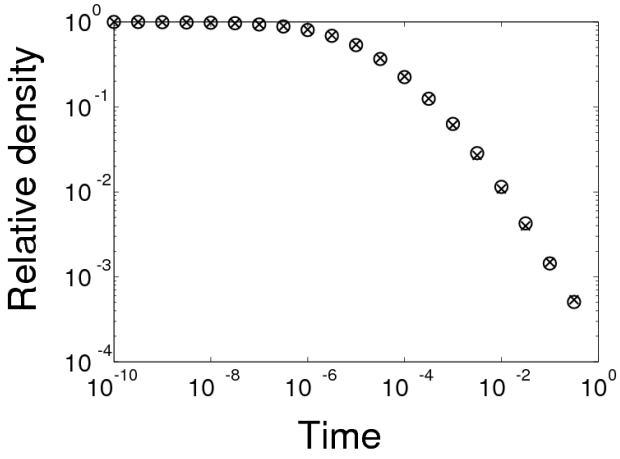

Figure 6: Comparison between the standard KMC calculations (crosses) and the first-passage KMC (circles) results for of $A + A \rightarrow A$ coalescence reaction by continuous-space continuous-time Brownian motion in $3d$.

cube volume with periodic boundary conditions applied in all 3 dimensions. The two kinetics are seen to be identical within small statistical errors.

The greater efficiency of the FPKMC method allows simulations of very large numbers of diffusing and reacting particles at a modest computational effort. An example is shown in Fig. 7 for the same annihilation reaction in $3d$ but starting with 125 million particles. The reaction completes in 45 CPU hours on a modest workstation. By comparison, we estimate

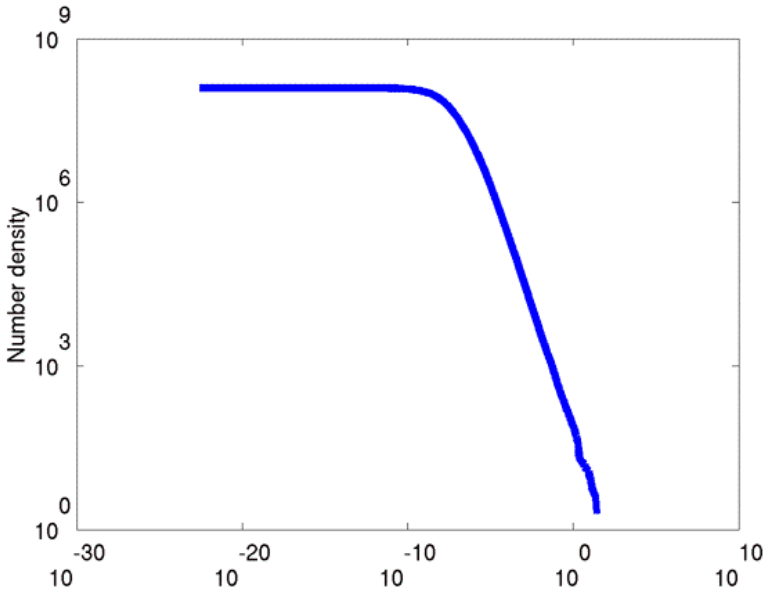

Figure 7: Simulated kinetics of $A + A \rightarrow 0$ reactions in $3d$. The initial number of particles is $125 \cdot 10^6$ and the reaction completes in 45 CPU hours.

that it would take tens of CPU years on the same workstation to complete this simulation using the standard KMC algorithm [3].

Each step of the FPKMC algorithm requires more calculations than one step of the standard KMC algorithm. The obvious overhead due to the need to sample from the more complicated distributions is relatively minor whereas more serious amount of computational effort in FPKMC is spent on keeping track of near neighbor particles, space partitioning and other tasks associated with particle protection, as well as on maintaining the event queue after every event. Note that all of these elements appear in other KMC algorithms in one form or another, and therefore standard techniques can be used. In paper II we discuss in more detail the computational costs of various elements of the FPKMC algorithm as implemented in a more advanced code. FPKMC codes used for simulations presented in this paper have not been extensively optimized although some of the more obvious inefficiencies

[3] The code we refer to here as "standard KMC algorithm" is *BigMac* [4] that has been extensively used for simulations of various diffusion-reaction processes. *BigMac* has not been specifically optimized for conditions of low particle density.

have been addressed. Nevertheless, it should be of interest to compare the net efficiency of the FPKMC simulations to that of the standard KMC method in the units of CPU time per particle collision.

A relevant comparison is given in Fig. 8, where the number of particle collisions (annihilation) per second of CPU time is plotted as a function of particle density for two series of simulations of $A + A \Rightarrow 0$ annihilation reaction in $3d$ using the FPKMC and the standard KMC algorithms. Each of the two series consists of three simulations starting from the same high initial volume fraction of particles of 0.05 and ending at a much lower volume density 10^{-6} (the simulations proceed from right to left). Because in the beginning some of the particles are very close to each other, it is necessary to use very small hop sizes in order to ensure that collision sequences are properly resolved. In the course of the simulation the nearest neighbor particle pairs progressively annihilate and the time step gradually increases. Eventually, as the average particle spacing gradually increases due to continued annihilation, the average number of hops between any two collisions also increases and efficiency of the standard method deteriorates inversely proportionally to the particle volume fraction (density) ρ. At the same time, FPKMC automatically selects the propagation step size to meet the local geometrical requirements, achieving an exact solution without any tuning. Efficiency of the FPKMC algorithm is proportional to $\rho^{-\frac{1}{3}}$ throughout the whole range of simulated particle densities. Thus, while FPKMC is competitive with the standard method even at high particle densities, at a sufficiently low density the new algorithm is certain to outperform the standard method.

Exactly how the new method's efficiency compares to traditional hop-by-hop KMC with varying particle density depends on dimensionality and, possibly, on whether or not the walks are continuous or discrete. For the case of continuous diffusion in $1d$, simulation efficiency is manifestly independent of particle separation. In higher dimensions, efficiency of the FPKMC method diminishes with decreasing density. Luckily, the need for efficient and accurate simulations of diffusion-reaction processes is limited to dimensions $d < 4$ because for $d \geqslant 4$ the effect of correlations can be neglected and the mean field kinetics becomes accurate [16].

Efficiency of Monte Carlo simulations of diffusion-controlled annihilation or coalescence reactions can be further enhanced by periodic replication. Although not directly related to the methods discussed in this paper, periodic replication is especially efficient when used

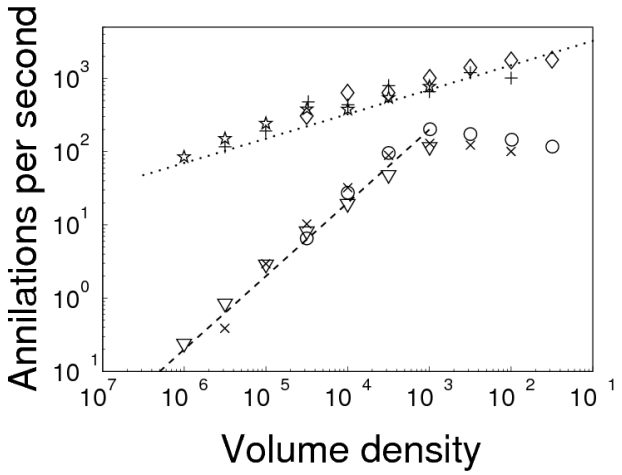

Figure 8: Computational performance as a function of particle density measured in six independent realizations of $A + A \rightarrow 0$ annihilation reaction in *3d* performed on a single CPU workstation using the standard KMC calculations with a finite hop distance (lower symbols) and the FPKMC algorithm (upper symbols). The dashed and dotted lines are theoretical estimates with slopes of 1 for the standard KMC calculations and slope $\frac{1}{3}$ for the new algorithm.

in combination with the FPKMC method. In has been noted in the past [17] that in the reactive systems where the number of particles steadily decreases, the simulated kinetics of the late stages of the reaction are not representative of the kinetics of interest. This is due to an inevitable growth of correlations among particles remaining in the box as the simulation progresses. In annihilation and coalescence reactions, correlations develop because the surviving particles are less likely to have reacting neighbors in close proximity. This "correlation hole" effect spaces the particles more evenly than in a random configuration, thus affecting the distribution of times of particle collisions. If and when the growing correlation length becomes comparable to the simulation box size, the kinetics becomes distorted, as a finite size effect. Reduction in the number of surviving particles compounds the difficulties making the tail kinetics noisy. For these reasons, the last 2-3 decades of simulated kinetics are typically discarded [17]. On the other hand, the first few decades of the simulated kinetics are often discarded because they reflect more the initial particle distribution (usually random) rather than the reaction kinetics of interest. Combined with the usual limit on the number of particles (typically 10^6), these unwanted behaviors limit to 4-6 decades the useful

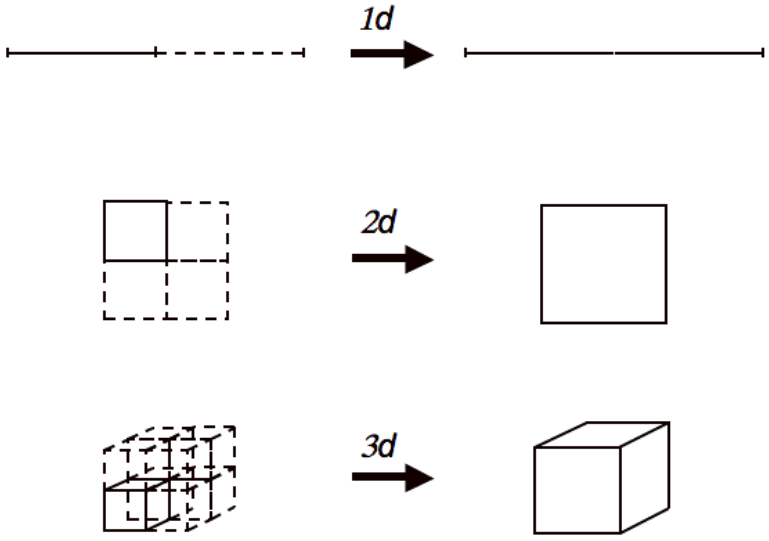

Figure 9: Schematic of the periodic replication procedure.

time interval over which one can observe and quantify diffusion-reaction kinetics.

Periodic replication works as follows [18]. First, the simulation starts from its initial configuration in a periodic box and proceeds as usual. Then, once the correlation length grows comparable to the box size, the particles are synchronized to the current point in time and 2^d neighboring periodic replicas are combined in a new box double the linear size of the old box, see Fig. 9. The simulation restarts in the new box in which the particles that were previously periodic image slaves of each other are now treated independently. Such box doubling should be repeated whenever the correlation length approaches half of the current box size. Given that correlations propagate by diffusion, we estimate that the correlation length should grow as $\sim \sqrt{Dt}$ where D is the diffusion coefficient. Thus, the volume should be replicated whenever $\sqrt{Dt}$ grows to become an appreciable fraction of the linear box size L. Hence, the physical time t elapsed between replications should at least quadruple with each replication. Assuming that in the interval between two box replications the number of surviving particles decreases as $t^{-\alpha}$, each replication increases the number of particles by a factor $2^{d-2\alpha}$.

Remarkably, for $A+A \Rightarrow 0$ and $A+A \Rightarrow A$ reactions in *1d*, $\alpha = \frac{1}{2}$ [19] and the doubling

in the number of particles caused by each replication is compensated by the reduction by half caused by annihilation or coalescence taking place between two replications. Thus, in these particular cases replications do not cause the number of particles to grow and can continue indefinitely. Combined with the fact that FPKMC efficiency does not depend on particles density in $1d$, we can simulate such processes to an arbitrarily long physical time, as shown in Fig. 10.

In other cases when $d-2\alpha > 0$, periodic replications will result in increases in the number of particles. For example, for the case $d = 2$, $\alpha = \frac{3}{4}$, the number of particles is expected to double after two replications, whereas for the case $d = 3$, $\alpha = 1$, the number of particles will double after each replication. Therefore, sooner or later the number of particles will grow too large to continue. Thus, rather than follow the common practice to start from a maximum size that fits into memory and then simulate the reaction to the end, it may be better to start from a small number of particles and let the system grow by replications to a maximum size afforded by the computer memory. Assuming that with FPKMC we can handle simulations with at most 10^9 particles and that is is safe to start with just 10^3 particles in the box, this allows as many as $\log_2(\frac{10^9}{10^3}) \approx 20$ replications for the $d = 3$, $\alpha = 1$ case and 40 replications for the $d = 2$, $\alpha = \frac{3}{4}$ case. Thus, replications should allow extension of the useful time horizon of such reaction-diffusion simulations to 4^{20} (12 decades of time) and 4^{40} (24 decades of time), respectively. Obviously, this recipe also eliminates the earlier mentioned tail effects since correlations are never allowed to catch up with the growing box size and the number of particles in the end of the simulation is large.

C. Two-species annihilation in continuum $3d$

The last computational test we report in this paper is a simulation of $A + B \Rightarrow 0$ annihilation reaction in $3d$. In this reaction, particles do not interact with particles of its own kind but annihilate on collisions with unlike particles. As has been first observed in Ref. [16], when the numbers of A and B particles are close to the stoichiometry (50:50), this reaction does not follow the mean field asymptotic kinetics t^{-1} but rather $t^{-\frac{3}{4}}$ (most other diffusion-reaction systems follow the mean field behavior for $d > 2$). This peculiar scaling was attributed to the emergence and growth of alternating A-rich and B-rich domains that effectively limit the annihilation reactions to inter-domain boundaries.

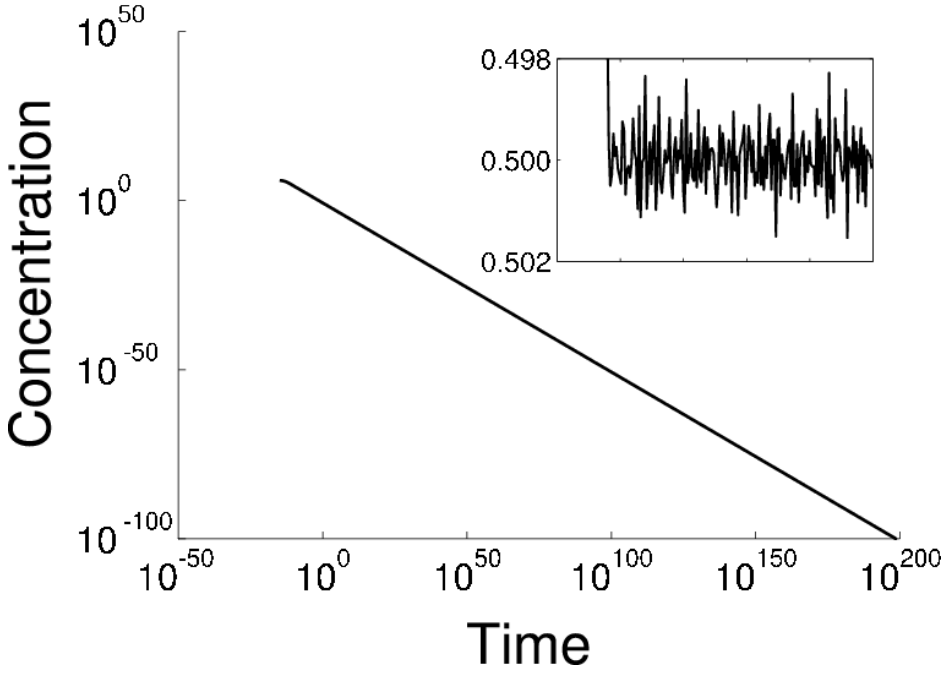

Figure 10: Kinetics of $A + A \rightarrow 0$ annihilation reaction in $1d$, using replication. The inset is the logarithmic derivative of the kinetic curve taken over the same time interval reproducing the theoretical exponent $\alpha = 0.5$ for this reaction.

An important physical realization of such a situation is recombination of vacancies and interstitials produced in crystal by neutron, ion or electron irradiation [5]. We deal with such reactions in considerable detail in paper II. Here we limit our study to a model system in which particles A and B are cubes with the same size and diffusion coefficients and react only with particles of the opposite species. The slow-down caused by domain growth combined with steadily decreasing particle density makes standard KMC simulations particularly inefficient which has so far prevented quantitative investigations of reaction kinetics and domain geometry in such systems, especially in $3d$. The FPKMC method handles this reaction with relative ease in arbitrary dimensions. Figure 11 shows the geometry of a thin slice through the domain configuration produced in a FPKMC simulation of $A + B \Rightarrow 0$ reaction in $3d$ starting with 10^6 particles (no replication).

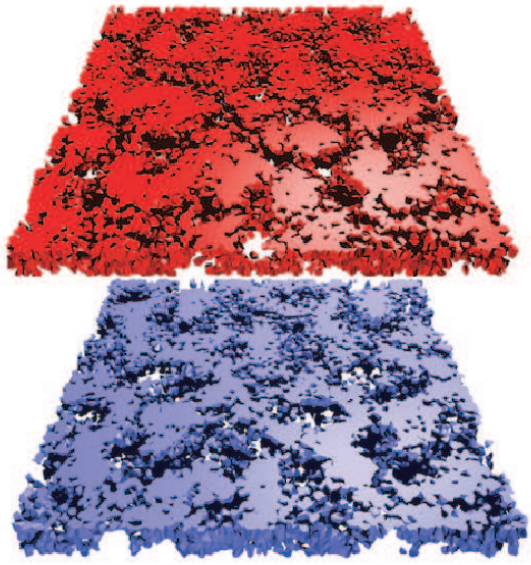

Figure 11: A thin slice through the domain structure formed in a simulation of $A+B \to 0$ reaction in *3d*. The boundaries of the A-rich (red) and B-rich (blue) domains are identified with the sides of Voronoi polyhedra shared by unlike particles. The two domains are complementary and fill the space when brought together.

VI. SUMMARY

We have developed the method of First-Passage Kinetic Monte Carlo (FPKMC) for simulations of diffusion-reaction processes. By partitioning the space into non-overlapping protective domains around each particle and/or particle pair, the N-body problem of collisions among N Brownian particles or random walkers is factorized into N single-body problems or, alternatively, K_1 single body and K_2 two-body problems, $K_1 + 2K_2 = N$. Rather than performing small diffusional hops, exact solutions for first-passage and no-passage statistics are used to propagate the particles inside the domains. On each Monte Carlo cycle a single particle or a single particle pair is propagated to the boundary of its protective domain. This is sometimes followed by a no-passage propagation of one or few neighboring particles or pairs.

The resulting algorithm is event-driven and asynchronous: each protected particle or pair propagates in its own spatial domain, from its own spatial and time origin and following its own propagation time clock. The new method remains efficient at low densities because only

one or a few particles are propagated on every cycle over distances close to the inter-particle spacing. The FPKMC method is exact for a wide class of diffusion-reaction models in which Brownian particles or random walkers do not interact until they collide (hard-core models). The accuracy and efficiency of the new method is demonstrated in simulations of several well-studied diffusion-reaction models that have previously presented serious computational challenges for Monte Carlo simulations.

We would like to emphasize that the FPKMC method focuses on bringing the particles close to collisions leaving aside the nature of reactions taking place on collisions. Thus, although statistics of first passage processes finds its uses in efficient handling of the reaction events [20], such issues are outside the scope of this paper in which we consider only the simplest collision outcomes - annihilation and coalescence. In the second part of this series of two papers we will present methods for simulations of more complicated reaction kinetics in which diffusional propagation takes place simultaneously with other competing stochastic processes.

VII. ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work was supported by the Office of Laboratory Directed Research at LLNL and the Office of Basic Energy Sciences U. S. Department of Energy. The authors would like to express their gratitude to G. Martin, S. Redner, W. Cai, C. Mailhiot, F. Willaime, M.-C. Marinica and T. Diaz de la Rubia for fruitful discussions.

VIII. APPENDICES

Appendix A: SAMPLING FROM SERIES EXPANSIONS

The distributions sampled in FPKMC are given in the form of series expansions. This appendix describes a general rejection technique and its application to sampling from such series expansions. In the following appendix B, first-passage (FP) and no-passage (NP) propagators suitable for this technique are derived for the case of continuous diffusion. FP

and NP propagators for discrete random walks on lattices will be presented in forthcoming publications.

Here by a distribution $c(x)$ we mean a function that is non-negative everywhere and whose integral is bounded, i.e. $c(x) \geq 0$ and $\int c(x)dx < \infty$. Sampling from $c(x)$ means drawing random numbers x distributed according to the probability density $c(x)/\int c(x')dx'$. Rejection sampling is often used when it is difficult to invert the cumulative distribution, e.g. to solve $\xi = \int^x c(x')dx'$ for x, and, at the same time, a majoring distribution $C(x)$ exists such that it is easy to sample and $C(x) \geq c(x)$ for all x. Rejection sampling proceeds as follows:

1. Let x_{trial} be a sample of C.

2. Pick a uniformly distributed random number $0 \leq y < C(x_{trial})$.

3. If $y < c(x_{trial})$, accept x_{trial} as a sample of c, otherwise reject x_{trial} and go to step 1.

If $\int_0^\infty C(t)dt/\int_0^\infty c(t)dt - 1$ is small, rejection is infrequent and the resulting sampling is efficient while still exact in the sense of sampling from the true distribution $c(x,t)$.

When the distribution c is available in the form of a converging series expansion, the partial sums of the series can be used in the acceptance/rejection test in step 2 of the rejection sampling algorithm above. Suppose $c(x) = \sum_{k=0}^\infty c_k(x)$, define the partial sums $S_m = \sum_{k=0}^m c_k(x)$ and assume that upper and lower bounds U_m and L_m of the remainder term are available, so that $L_m \leq c(x) - S_m \leq U_m$ and both bounds become tighter with each added term. Then, in step 3 above, if $y < S_m + L_m$ the sample is accepted without evaluating any terms beyond m. Conversely, if $y \geq S_m + U_m$ the sample is rejected. If however $S_m + L_m < y \leq S_m + U_m$ no decision can be made and the next order terms have to be calculated in order to repeat the test with the same sample y but using S_{m+1}, U_{m+1} and L_{m+1}. Especially simple is the case when the sign of the remainder term $c(x) - S_m$ is known; the sample is accepted if $y < S_m$ and the remainder is positive and rejected if $y \geq S_m$ and the remainder is negative. Our experience with the FPKMC algorithm suggests that the rejection procedure requires computing only two terms on average before the sample is accepted or rejected.

Appendix B: PROPAGATORS FOR CONTINUOUS DIFFUSION

Here we derive the first passage and no passage propagators suitable for rejection sampling on one dimensional line segments for the case of continuous diffusion.

1. First-passage propagator

Assume $c(x,t)$ is the solution to the diffusion equation given in section IV. From Eq. (1) it follows that the exit probability per unit time is $p(t) = \frac{\partial c}{\partial x}(0,t) - \frac{\partial c}{\partial x}(1,t)$. Taking the solution in the form of the image series suitable for short times [c.f. Eq. (5)] we obtain:

$$p(t) = \frac{2\pi}{\sqrt{4\pi t}^3} \sum_{j=-\infty}^{\infty} (-1)^j \left\{ \left(j+\frac{1}{2}\right) e^{-\left(j+\frac{1}{2}\right)^2/4t} - \left(j-\frac{1}{2}\right) e^{-\left(j-\frac{1}{2}\right)^2/4t} \right\}.$$

Retaining the two most significant terms in this series yields an approximate expression accurate for short times

$$p_s(t) = \frac{4\pi}{\sqrt{4\pi t}^3} e^{-\frac{1}{16t}}.$$

Similarly, retaining the first term in the long time expansion (4)

$$p(t) = \sum_{k=1}^{\infty} 2k\pi \sin(k\pi x_0)[1-(-1)^k] e^{-k^2\pi^2 t},$$

and using $x_0 = 1/2$, we obtain an approximation that is accurate for long times,

$$p_l(t) = 4\pi e^{-\pi^2 t}.$$

In turns out that $p_s > p$ and $p_l > p$ for all times t and, furthermore, p_s and p_l intersect at $\tau_0 \approx 0.0796$. Thus, $\min(p_s, p_l)$ is a tight majoring function for the true distribution p that it is accurate to within 0.6% for all t, as shown in Fig. 12.

Defining the integrals

$$\begin{aligned} F_l(\tau) &= \textstyle\int_\tau^\infty p_l(t)dt = \frac{4}{\pi} e^{-\pi^2\tau} \\ F_s(\tau) &= \textstyle\int_\infty^\tau p_s(t)dt = 2\left[1 - \operatorname{erf}\left(\frac{1}{\sqrt{16t}}\right)\right] \end{aligned},$$

a sample of the exit time t_{trial} from the majoring distribution is

$$t_{trial} = F_s^{-1}(r) = \frac{1}{16\left[\operatorname{erf}^{-1}\left(1-\frac{r}{2}\right)\right]^2},$$

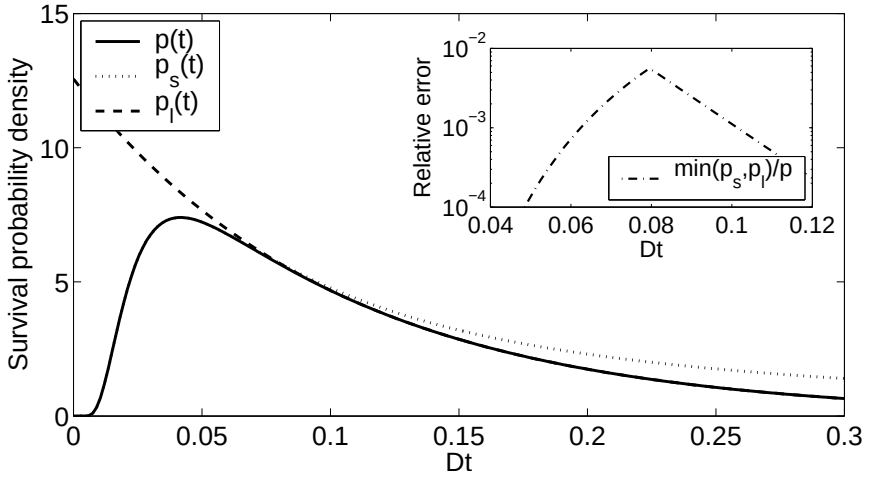

Figure 12: Comparison of one-term expansions versus full probability density. The inset shows the relative error of using only the proposed majoring function without subsequent rejection sampling.

if $r < F_s(\tau_0)$, and

$$t_{\text{trial}} = F_l^{-1}(r - F_s(\tau_0)) = \tau_0 - \pi^{-2} \log \frac{r - F_s(\tau_0)}{F_l(\tau_0)},$$

otherwise. Here, r is a random number uniformly distributed on the segment $0 < r \leq F_s(\tau_0) + F_l(\tau_0)$. Using this trial value of the exit time, it is now straightforward to employ the rejection sampling technique described in appendix A. For the short time series, the terms of the series alternate in sign and decrease in magnitude with increasing m so that the partial sums S_m provide an alternating sequence of upper and lower bounds. The same holds for the long time series, provided $\tau_{trial} \geq \frac{1}{18\pi^2} \approx 0.0056$. Depending on the value of t_{trial}, one or the other series converges faster. It is possible to choose among the two alternatives by comparing t_{trial} to τ_0. However, it is more efficient to use a different switchover time t_{switch} that optimizes the computational cost of the sampling routine. For our implementation $t_{switch} \approx 0.033$ turned out to be optimal.

2. No-passage propagator

The no-passage propagator is the distribution density of particle positions at time t conditioned on the fact that the particle has not reached the boundary of its protective domain by that time. As is the case for the FP propagator described above, sampling from the NP distribution is most efficiently done using two different expansion series at short times

and at long times. Using the sampling procedure described below, rejections are infrequent (less than 1%) requiring on average less than two terms in the series expansions to accept or reject the sample.

For short times, the probability density $c(x,t)$ is best represented by the image sum [c.f. Eq. (5)]. The $m = 0$ term of this expansion is the fundamental solution $C(x,t)$ for diffusion on $-\infty < x < \infty$. $C(x,t)$ is a simple over-estimator $C(x,t) \geq c(x,t)$ that can be used to obtain a trial sample for the particle position x_{trial} on $(0,1)$. Since $C(x,t)$ is a Gaussian, a trial position can be obtained by scaling and translation of a normally distributed random number r_n, $x_{\rm trial} = (1 + r_n\sqrt{8t})/2$ [x_{trial} can occasionally fall outside $(0,1)$ in which case it is discarded]. With a trial position so selected, the partial sums S_m of the image series for concentration $c(x,t)$ are used as an alternating sequence of increasingly tight upper and lower bounds convenient for rejection sampling, as described in appendix A.

At long times, as an approximation for the particle position distribution it seems reasonable to take the first term of the eigenfunction expansion [c.f. Eq. 4], $\tilde{c}(x,t) = \sin(k\pi x)e^{-\pi^2 t}$. Although this function is smaller than the full solution $c(x,t)$ for some x, it is still possible to use it to construct a tight majoring function $C(x,t) \geq c(x,t)$ by multiplying $\tilde{c}$ with a factor $1 + g(t)$, so that $C(x,t) = [1 + g(t)]\,\tilde{c}(x,t) \geq c(x,t)$, for all x. One possible choice for $g(t)$ is

$$g = \frac{e^{-8\pi^2 t}}{1 - e^{-16\pi^2 t}}.$$

This particular factor was derived by taking the absolute value of every term in the eigenfunction series expansion, noting that $|\sin x| \leq 1$ and that $x^2 \geq x$ for $x \geq 1$, and replacing the square in the exponential by a linear function. The resulting sum is a geometric sum and can be evaluated analytically.

Sampling x_{trial} from the majoring distribution is then performed by picking a uniformly distributed random number $-1 \leq r < 1$, and setting $x_{trial} = \frac{1}{2} + \frac{1}{2\pi}\arccos r$. In addition to x_{trial} we also need an estimate for the remainder of the long-time series. Following a derivation similar to that of g, we find that the function

$$d_m = 2\frac{e^{-\pi^2 t(2m+3)^2}}{1 - e^{-4\pi^2 t}},$$

bounds the series remainder so that $c_m(x,t) - d_m \leq c(x,t) \leq c_m(x,t) + d_m$. The so defined $C(x,t)$ and d_m can be employed for rejection sampling. To reduce the cost, when it is

necessary to compute the higher order terms of the series expansion for $c(x,t)$, we re-use the already calculated time exponentials which requires a few multiplications for each iteration. The time t_{switch} for switchover from the short time series to the long time series can be selected to optimize the cost of rejection sampling, similar to the FP propagators described in the preceding section.

We note in passing that still tighter bounding functions g and d_m can be derived by replacing the infinite sum in $c(x,t)$ with a majoring integral. However the resulting expressions contain the error function, $\mathrm{erf}(x)$, which can be expensive to numerically evaluate.

Appendix C: JUSTIFICATION FOR THE ALGORITHM

The goal of the FPKMC algorithm is to generate a stochastic trajectory of the simulated system of N Brownian particles with the correct probability. To achieve this goal the FPKMC algorithm samples the system's configuration $\mathbf{R} = \{\mathbf{r}_1, \ldots, \mathbf{r}_N\}$ at time t from the Green's function of the Master equation. In the considered case of particle diffusion, the Master equation is a diffusion equation in the dN-dimensional space for the probability density $G(\mathbf{R}, t; \mathbf{R}_0)$ to find the system in configuration $\mathbf{R}$ at time t, starting from $\mathbf{R}_0$ at time $t_0 = 0$,

$$\frac{\partial G}{\partial t} = \mathcal{D} \cdot \boldsymbol{\nabla}^2 G \text{ in } \Omega, \text{ where } G(\mathbf{R}, 0) = \delta(\mathbf{R} - \mathbf{R}_0) \text{ and } G(\partial\Omega, t) = 0,$$

where $\mathcal{D} = \{D_1, \ldots, D_N\}$ is a (block-diagonal) diffusion tensor. Here Ω denotes the region of phase space bounded by the non-overlap conditions between the hard particles. As soon as the configuration leaves region Ω through the bounding surface $\partial\Omega$ a hard-sphere collision occurs [4].

What needs to be demonstrated is that the FPKMC algorithm correctly samples from $G(\mathbf{R}, t; \mathbf{R}_0)$ even if the latter Green's function is unknown. More specifically, the algorithm should correctly sample the time $\tilde{t} < t$ and configuration $\tilde{\mathbf{R}} \in \partial\Omega$ when the first collision occurs or, alternatively, sample a configuration $\mathbf{R}$ given that no collision has occurred by time t.

Rather than trying to solve for the exact Green's function $G(\mathbf{R}, t; \mathbf{R}_0)$ - a difficult task indeed given the complicated shape of the collision boundary $\partial\Omega$ - the FPKMC algorithm

[4] Exactly how the particles react on collisions is outside the scope of the FPKMC algorithm.

relies on a much simpler Green's function obtained for an auxiliary sub-problem in which the particles are spatially protected against each other,

$$\frac{\partial G_0}{\partial t} = \mathcal{D} \cdot \boldsymbol{\nabla}^2 G_0 \text{ in } \Omega_0 \text{ where } G_0(\mathbf{R}, 0) = \delta(\mathbf{R} - \mathbf{R}_0) \text{ and } G_0(\partial\Omega_0, t) = 0,$$

where now $\Omega_0 \subset \Omega$ is a Cartesian product of subspaces, each of which is a domain only of the coordinates of a single particle, $\Omega_0 = \mathcal{P}_1 \otimes \mathcal{P}_2 \otimes \cdots \otimes \mathcal{P}_N$. This sub-problem is fully separable into corresponding sub-problems for the Green's function $g_i(\mathbf{r}, t; \mathbf{r}_i^0)$ for each of the independent particles

$$\frac{\partial g_i}{\partial t} = D_i \boldsymbol{\nabla}^2 g_i \text{ in } \mathcal{P}_i \text{ where } g_i(\mathbf{r}, 0) = \delta(\mathbf{r} - \mathbf{r}_i^0) \text{ and } g_i(\partial\mathcal{P}_i, t) = 0,$$

to give $G_0(\mathbf{R}, t; \mathbf{R}_0) = \prod_{i=1}^{N} g_i(\mathbf{r}_i, t; \mathbf{r}_i^0)$.

As elaborated in Ref. [9], using Green's theorem it can be shown that

$$G(\mathbf{R}, t; \mathbf{R}_0) = G_0(\mathbf{R}, t; \mathbf{R}_0) + \int_{\tilde{\mathbf{R}} \in \partial\Omega_0} \int_{\tilde{t}=0}^{t} \mathbf{J}_n(\tilde{\mathbf{R}}, \tilde{t}; \mathbf{R}_0) G(\mathbf{R}, t - \tilde{t}; \tilde{\mathbf{R}}) d\tilde{t} \partial S,$$

where $\mathbf{J}_n(\tilde{\mathbf{R}}, \tilde{t}; \mathbf{R}_0) = -\mathcal{D} \cdot \boldsymbol{\nabla}_{\mathbf{n}} G_0(\tilde{\mathbf{R}}, \tilde{t}; \mathbf{R}_0)$ denotes the normal component of the current at the auxiliary surface $\partial\Omega_0$. The probablistic meaning of the above equation is that the system's configuration at time t can be sampled by first sampling an exit time $\tilde{t}$ and location $\tilde{\mathbf{R}} \in \partial\Omega_0$ on the auxiliary boundary. Then, if the time of interest is shorter than the sampled exit time, $t < \tilde{t}$, the exit location is discarded and the system's configuration is sampled from (the normalized form of) $G_0(\mathbf{R}, t; \mathbf{R}_0)$. Otherwise one recursively repeats the process by constructing a new Ω_0 and samples an exit location and time starting from the new time origin, until the walk hits $\partial\Omega_0$ (this requires at least one pair protection). In this case, the exit location on the first-passage boundary $\partial\Omega_0$ serves as a source for subsequent propagation since the random walk is a memory-less Markov process that can continue from the last sampled point forgetting the past.

Owing to the separability of the protected walkers, the current is expressed as [9]

$$\mathbf{J}_n(\tilde{\mathbf{R}}, \tilde{t}; \mathbf{R}_0) = \sum_m \left(\prod_{l \neq m} g_l(\mathbf{r}_l, \tilde{t}; \mathbf{r}_l^0) \right) \left[-\boldsymbol{\nabla}_n g_m(\tilde{\mathbf{r}}_m, \tilde{t}; \mathbf{r}_m^0) \right],$$

where now m denotes the walker that first leaves its protection through $\tilde{\mathbf{r}}_m \in \partial\mathcal{P}_m$ at time $\tilde{t}$. The probabilistic meaning is as follows: first select a walker m and the associated $\tilde{t} = \tilde{t}_m$ with the correct probability, and then sample positions for all other walkers from (the normalized

forms of) single particle distributions g_l's. It is shown in Ref. [9] that m and the associated $\tilde{t}$ can be sampled by simply choosing the minimum among independently sampled exit times, one time for each walker.

Numerical efficiency of the FPKMC algorithm, however, derives from an important additional observation not used in Ref. [9]: it is not necessary to sample g_l for $N-1$ walkers that have not exited their protections. One can of course sample $N-1$ new positions $\mathbf{r}_l \in \mathcal{P}_l$ at time $\tilde{t}$, and then sample $N-1$ new exit times from $g_l(\mathbf{r}, t-\tilde{t}; \mathbf{r}_l)$. However, as discussed in Section II B, the new exit times and locations have exactly the same distributions as those sampled originally. Therefore, those particles that have not exited their protections can be left in the queue or updated to the current point in time at will. For computational efficiency, the first-passage predictions for walkers are updated only when this becomes necessary, namely, for the walker m with the shortest exit time in the queue and, perhaps, for one or few other walkers whose protective regions need to be modified.

---

[1] M. Strobel, K.-H. Heinig, and W. Möller. Three-dimensional domain growth on the size scale of the capillary length: Effective growth exponent and comparative atomistic and mean-field simulations. *Phys. Rev. B*, 64(24):245422, Dec 2001.

[2] J. S. Reese, S. Raimondeau, and D. G. Vlachos. Monte Carlo Algorithms for Complex Surface Reaction Mechanisms: Efficiency and Accuracy. *J. Comp. Phys.*, 173(1):302–321, 2001.

[3] M. Biehl. Lattice gas models and Kinetic Monte Carlo simulations of epitaxial growth. In A. Voigt, editor, *Int. Series of Numerical Mathematics 149*, pages 3–18. Birkhaeuser, 2005.

[4] S. K. Theiss, M.-J. Caturla, M. D. Johnson, J. Zhu, T. J. Lenosky, B. Sadigh, and T. Diaz de la Rubia. Atomic scale models of ion implantation and dopant diffusion in silico. *Thin Solid Films*, 365:219–230, 2000.

[5] C. Domain, C. S. Becquart, and L. Malerba. Simulation of radiation damage in Fe alloys: an object kinetic Monte Carlo approach. *Journal of Nuclear Materials*, 335(1):121–145, 2004.

[6] D. P. Tolle and N. Le Novere. Particle-based stochastic simulation in systems biology. *Current Bioinformatics*, 1(3):315–320, August 2006.

[7] J. S. van Zon and P. R. ten Wolde. Green's-function reaction dynamics: A particle-based approach for simulating biochemical networks in time and space. *J. Chem. Phys.*, 123(23):234910,

2005.

[8] S. J. Plimpton and A. Slepoy. Microbial cell modeling via reacting diffusive particles. *J. Phys.: Conf. Ser.*, 16:305–309, 2005.

[9] M. H. Kalos, D. Levesque, and L. Verlet. Helium at zero temperature with hard-sphere and other forces. *Phys. Rev. A*, 9(5):2178–2195, 1974.

[10] J. Dalla Torre, J.-L. Bocquet, N. V. Doan, E. Adam, and A. Barbu. JERK, an event-based Kinetic Monte Carlo model to predict microstructure evolution of materials under irradiation. *Philosophical Magazine*, 85:549–558, 2005.

[11] A. Donev. Asynchronous event-driven particle algorithms. *SIMULATION: Transactions of The Society for Modeling and Simulation International*, 85(4):229–242, 2008.

[12] S. Redner. *A guide to first-passage processes*. Cambridge University Press, Cambridge, 2001.

[13] Daniel ben Avraham. Computer simulation methods for diffusion-controlled reactions. *J. Chem. Phys.*, 88(2):941–948, 1988.

[14] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for Monte Carlo simulation of Ising spin systems. *J. Comput. Phys.*, 17(1):10–18, 1975.

[15] S. S.Andrews and D. Bray. Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Physical biology*, 1(3):137–151, 2004.

[16] F. Leyvraz and S. Redner. Spatial organization in the two-species annihilation reaction $A + B \to 0$. *Phys. Rev. Lett.*, 66(16):2168–2171, 1991.

[17] Y. Shafrir and D. ben Avraham. Large scale simulations of two-species annihilation, $A+B \to 0$, with drift. *Phys. Lett. A*, 278(4):184–190, 2001.

[18] M. Smith and T. Matsoukas. Constant-number Monte Carlo simulation of population balances. *Chemical Engineering Science*, 53(9):1777–1786, 1998.

[19] D. Zhong, R. Dawkins, and D. ben Avraham. Large-scale simulations of diffusion-limited n-species annihilation. *Phys. Rev. E*, 67(4):040101, 2003.

[20] H. Kim and K. J. Shin. Exact Solution of the Reversible Diffusion-Influenced Reaction for an Isolated Pair in Three Dimensions. *Phys. Rev. Lett.*, 82(7):1578–1581, 1999.