An Exact and Efficient First Passage Time Algorithm for Reaction-Diffusion Processes on a 2D-Lattice

Andri Bezzola[a,*], Benjamin B. Bales[a], Richard C. Alkire[b], Linda R. Petzold[c]

[a] *Mechanical Engineering Department, University of California, Santa Barbara, CA 93106, United States*
[b] *Department of Chemical Engineering, University of Illinois, Urbana, IL 61801, United States*
[c] *Mechanical Engineering Department and Computer Science Department, University of California, Santa Barbara, CA 93106, United States*

Abstract

We present an exact and efficient algorithm for reaction-diffusion-nucleation processes on a 2D lattice. The algorithm makes use of first-passage time (FPT) to replace the computationally intensive simulation of diffusion hops in KMC by larger jumps when particles are far away from step-edges or other particles. Our approach computes exact probability distributions of jump times and target locations in a closed-form formula, based on the eigenvectors and eigenvalues of the corresponding 1D transition matrix, maintaining atomic-scale resolution of resulting shapes of deposit islands. We have applied our method to three different test cases of electrodepositon: pure diffusional aggregation for large ranges of diffusivity rates and for simulation domain sizes of up to 4096x4096 sites, the effect of diffusivity on island shapes and sizes in combination with a KMC edge diffusion, and the calculation of an exclusion zone in front of a step-edge, confirming statistical equivalence to standard KMC simulations. The algorithm achieves significant speedup compared to standard KMC for cases where particles diffuse over long distances before nucleating with other particles or being captured by larger islands.

1. Introduction

Systems that evolve as a result of random stochastic movement and interaction among particles are typically modelled by Kinetic Monte Carlo (KMC) methods owing to their simple and robust computational framework that can provide essentially exact results [1]. However, significant limitations are known to arise when time-scales are disparate, such as when it takes many random steps for particles to find each other and interact.

KMC simulations of reaction-diffusion-nucleation systems based on the algorithm by Bortz, Kalos and Lebowitz (BKL) [1] have been widely used with

[*] Corresponding Author

success. However, an unfavorable situation may arise whenever the walker density is low and walkers have long distances to travel before they can nucleate or attach to an edge; in such cases, most of the computational effort will be spent on the independent random walks in order to bring two particles close enough to each other for an event to happen. This situation arises when the surface diffusion rate is large compared to the deposition rate [15]. In such cases, poor computational efficiency can render traditional KMC methods infeasible.

Coarse-grain Monte Carlo methods (CGKMC) superimpose a coarse grid over the particles and use local mean-field values or quasi-chemical approximations to characterize their properties. Transition rates are then based on the coarse cell size, generally allowing for larger diffusion hops at lower rates. The main challenges in using these methods are the loss of atomic scale resolution, as well as the need for assumptions on interactions that occur within the coarse cells. (An overview of KMC and CGKMC methods is available in [4] and the references therein). Hybrid methods are available [6, 29, 12] in which continuum regions far from growth sites are coupled with a KMC method close to the growth sites.

Island dynamics (ID) [14, 3, 5, 25] has also been used to simulate surface evolution during growth. This method has been shown to be applicable for complex systems including electrodeposition with additives [30, 31]. The accuracy and efficiency of the original method have been enhanced by using adaptive grids for the level set function [20], or by combining the island dynamics method with KMC simulations at the edge of islands [33]. However, ID still requires a priori information on the nucleation rates for the simulation of epitaxial growth, as the particles are not tracked individually. Additionally, ID does not assume a physical grid for the movement of particles. Instead, particles move according to the continuum diffusion equation and, as such, ID is not well suited to replicate the small-scale geometric shapes of actual deposits on an atomic lattice.

Although approximate methods such as CGKMC and ID have had success in achieving improved computational speed, these too can fail when accuracy is critically important for events occurring at scales below that of the applied coarse grid [6]. In this work we address an application for which KMC is too time-consuming and approximate methods are unable to capture all of the relevant details of the phenomenon of interest: the atomic-scale nucleation and growth of thin metal deposits by electrodeposition at the very low rates associated with epitaxial and/or single crystal growth.

Precise control of nucleation and growth dynamics on metal and semiconductor surfaces is essential for manufacturing processes associated with solar cells, catalysts, new battery materials, and superlattices having unique optical and electronic properties. Many of these applications have the potential to be deployed in large-scale manufacturing processes. The need for fast algorithms arises for a number of reasons associated with the design of well-engineered materials. For example, although the fundamental science is relatively well-advanced for pristine systems, the surface chemistry and dynamics are often less clear and subject to modeling trial-and-error. For the case investigated in this work, nucleation densities are extremely low, and thus require large simulation domains to obtain accurate statistics for comparison with experimental measurements of nucleation densities and island size distributions. In addition, highly efficient algorithms are needed for the massively iterative calculations required for estimating parameters from experimental data, for identifying uncertainties and sensitivities, and for optimization and control [8, 9]. For all these reasons it is crucial to have a fast and efficient simulation framework that can still capture the relevant physics.

We are specifically interested in simulation of the early stages of nucleation on a pristine flat surface on which mobile atomic-scale adatoms (particles), formed by electrodeposition, diffuse about the surface by hopping from one lattice site to another. The adatoms diffuse until they encounter either another mobile adatom to form an immobile two-atom nucleus, or the edge of a previously-formed nucleus that grows by accretion of adatoms to form an island (as described in Section 2). The exact location of the diffusing particles is of less interest than the time and location of the interaction events associated with nucleation and growth. The use of various computational methods in the modeling of electrochemical systems has been reviewed recently [2].

Under these conditions, the density of mobile atomic-scale particles on the surface is so low that it is difficult for atomic-scale particles to find each other and interact to form a nucleus. Moreover, accurate simulation of nucleation events is critically important since long-term growth patterns are influenced by atomic-scale events. We report here a variation of the first-passage-time-KMC approach, tailored to 2D on-lattice surface processes citeopplestrup2006, oppelstrup2009.

In the first-passage-time (FPT) approach, unoccupied zones are established around each diffusing adatom on the surface, and one calculates the probability of when and where the adatom will exit its zone for the first time. The accuracy and performance of the FPT algorithm relies on the fast and accurate calculation of the exit time and exit location probability distributions.

A similar approach to FPT has been presented by Van Zon et al. with the Green's Function Reaction Dynamics (GFRD) method [36]. Both approaches are based on the finding that particles diffuse independently as long as they are far enough from each other. The GFRD method has its origins in chemistry where it has been applied to simulate stochastic systems with bimolecular reactions. Similarly to FPT, the particles in GFRD methods are assumed to be free of a physical lattice and the motion of particles can be described by Brownian motion. Both FPT and GFRD methods allow particles to take jumps much larger than the atomic length at appropriately reduced rates, as long as the assumption of non-interaction is valid. This can be accomplished by relying on analytic solutions of the diffusion equation to sample the motion of diffusive particles. The division of systems into sub-parts with master equations that can be analytically solved, has also proven very beneficial for chemical reaction systems [16, 10] and reaction-diffusion systems [24].

DeVita et al. [7] proposed another method that involves the rescaling of jump rates and allows for larger jumps. Their method scaled the time of the next diffusive jump with the inverse of the square of the jump distance. For each jump, a search is performed to decide whether that jump violates the no-interaction criteria. This approach is somewhat different than FPT and GFRD as it does not rely on the construction of protective zones. Instead, a large jump is rejected *a posteriori* if the no-interaction criteria are found to be violated.

We assume a discrete on-lattice random walk for the diffusive motion of particles and a continuous-time master equation due to the nature of surface diffusion on metal surfaces. Exit time and exit location probability distributions must be formulated accordingly. On a lattice, the first passage time for a random walk inside a domain can be computed analytically and exactly, as has been shown for the one dimensional First Passage Time KMC (FPKMC) in [21] and [34]. To the best of our knowledge, there are no publications that address the surface diffusion on a two dimensional physical lattice, incorporating nucleation and island growth based on an exact FPKMC algorithm. We present an efficient exact-lattice-FPT (ELFPT) method that gives exact probability distributions for the exit times and exit locations of protective zones on a 2D lattice. Our algorithm is exact: it solves the same master equation and produces the same statistics on the entire domain as KMC simulations on a lattice. The benefit of the proposed method over KMC lies in the combination of many diffusive lattice-hops into larger jumps to the edge of the protective zones, while leaving the other parts of the KMC event queue untouched. We investigate the conditions under which this approach may be advantageous by comparison with the KMC method, which calculates all the diffusion hops. The ELFPT method makes use of exact formulas that allow for an extremely efficient calculation of FPT's on a lattice in two or more dimensions.

The paper is organized as follows: In Section 2 we describe the reaction-diffusion-nucleation problem and present the ELFPT method. This entails the derivation of the exact FPT distributions for random walks on a lattice in one, two and higher dimensions, as well as a method for efficiently sampling an exit location in two or more dimensions. In Section 3 we demonstrate the efficiency of our algorithm compared to standard KMC. We conclude in Section 4 with a discussion.

2. The Exact-Lattice-First-Passage-time (ELFPT) method

The reaction-diffusion-nucleation problem selected for investigation here consists of the the physical phenomena shown in Figure 1. Reactions generate a flux F of new particles depositing onto the surface, which constitutes the simulation domain (1). Those particles can then move around the domain according to the laws of random walks (2). Particles can collide and form a nucleus (3), which is assumed to be a stable aggregate of two adjacent particles. When more particles collide with the aggregate they attach to it (4), leading to aggregate growth. Particles with one bond to an aggregate can diffuse along the aggregate edge (5) until they encounter a kink site (6), i.e. a site with two or more bonds to the aggregate, at which point the particles become immobile. In this work we always consider a two-particle nucleus to be immobile, even if the particles have only one lateral bond to other particles. Additional effects at island edges such as detachment from edges or dimer diffusion, while possible, are not considered in this work. While these assumptions provide a pristine model system suitable for algorithm development, there are many additional physical phenomena that may influence the complicated behavior of actual systems [11].

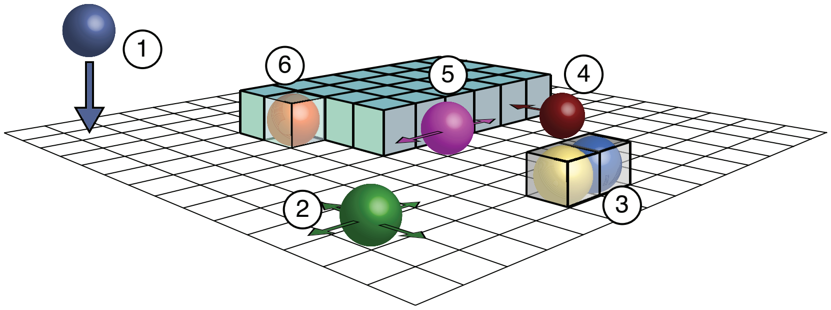

Figure 1: Schematic drawing of possible state transitions for a particle:(1) flux, (2) random walk on surface, (3) nucleation, (4) attachment to an edge, (5) edge diffusion, (6) kink site.

As mentioned previously, standard KMC algorithms struggle when the particles diffuse a long way before they encounter other particles or islands, because they must resolve all lattice hops of the diffusing particles. The main interest for reaction-diffusion-nucleation systems is the final outcome of aggregate size and shape. As long as the algorithm accurately predicts where particles nucleate and aggregate, it is not necessary to resolve the exact path a particle has taken before becoming part of an aggregate. ELFPT greatly reduces the computational cost for the simulation of particles diffusing long distances until nucleation or attachment to islands. The average inter-island distance depends strongly on the ratio D/F, where D is the surface diffusion rate and F is the rate of influx of particles per atomic site.

Before proceeding with the description of the algorithm, it is beneficial to define two types of event classes: A KMC-type event is defined as an event that has an exponentially distributed probability of happening within the next time increment δt [13]. These are called KMC-type events because they can be simulated using standard KMC methods based on the algorithm of Bortz, Kalos and Lebowitz [1] (BKL). Examples of KMC events include influx of particles onto the surface, external reactions, and other events represented by reactions. An FPT event is defined as a diffusion step of a particle. If the diffusion step is taken over more than one lattice site, then the FPT probability is not distributed exponentially because it combines a series of atomic lattice hops into one large FPT event. Only hops to direct neighbors ($L_s = 1$) are exponentially distributed (see Equation (6)). The lack of an exponential probability distribution disqualifies FPT diffusion steps to be used in the classical KMC algorithm.

The ELFPT algorithm is based on a chronological event queue and proceeds as follows:

1. Draw non-overlapping square protection zones centered around each particle (protection zones must not touch or include deposit sites).
2. Sample FPT's $\tau_1, \ldots, \tau_N$ for all particles, based on the size of their respective protection zone.
3. Add τ_i to ordered event queue.
4. Compute the time τ_{KMC} of the next KMC event (edge attachment, lattice site hops etc.) Add τ_{KMC} to event queue.
5. Execute event with smallest τ_{min}.
   (a) If next event is a FPT event:
      i. Sample exit position for particle with lowest FPT and move particle.
      ii. If new position violates zones of other particles, update position of those particles inside their zone (no-passage propagator) and remove their FPT's from the event queue.
      iii. Construct a new protective zone for each particle that moved in steps i. and ii. above.
      iv. Sample new FPT's for each of these particles and put the FPT's in the event queue.
   (b) If next event is a KMC event:
      i. Perform event.
      ii. Check for nucleation, edge attachment or other events that change topology.
      iii. Check all zones in neighborhood for violations due to new KMC event.
      iv. If violations are found, remove FPT's of those particles from event queue, update particles' locations inside their zones (no-passage propagator), calculate new FPT's for each of those particles and add FPT's to event queue.
6. If zone violations occurred in the previous step, check for violations again and keep updating the locations and zones of particles whose zone is violated until all zones adhere to the rules in Step 1.
7. Update system time to τ_{min}.
8. Continue with step 3 until termination criterion has been met.

Figure 2 shows a snapshot of a 2D simulation with particles at the center of their zones and some previously formed aggregate islands.

We note that the above algorithm requires that zones be constructed symmetrically around the particles. This requirement could be relaxed to non-symmetric zones with different side-lengths in each direction. At first glance, asymmetric zones of different side lengths might appear more promising, because such protective zones can cover the simulation domain more completely. However, the benefit of larger zones will be dampened and possibly annihilated by the overhead due to the construction and maintenance of the asymmetrical zones as well as the much more costly sampling of exit times and locations. In the next section we show how FPT's for symmetric protection zones in two and

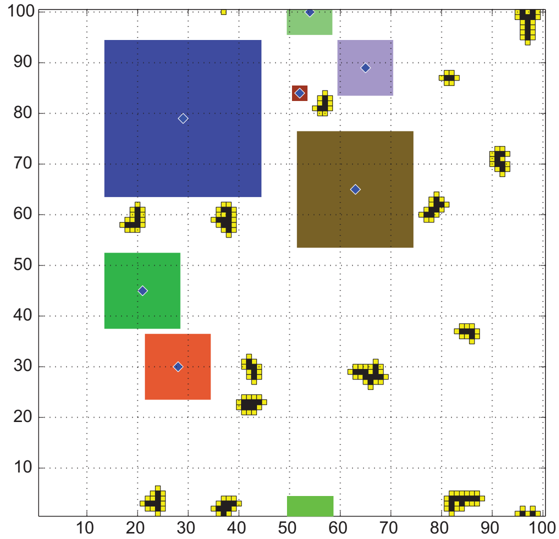

Figure 2: A top view of a 2D simulation with moving particles (blue diamonds), their zones at different sizes (different colors), and islands (black) with absorbing edge sites (yellow).

more dimensions can be very calculated extremely efficiently by reduction to FPT's in one dimension.

The algorithm requires the computation of exit times and exit locations, as well as updates of positions inside an existing zone. In the next few subsections we derive an exact and efficient method to calculate those distributions. We also note that the update at point (5) eventually reduces the movement of particles to standard KMC once they are close to each other or close to an island edge. Therefore, nucleation is detected when two particles get close enough to each other. In all of our simulations, the interaction radius of particles is one atomic lattice space. Thus particles nucleate when they are in adjacent lattice sites. The assumption of no interaction for particles that are more than one lattice spacing apart also justifies the fact that only particles who see their zones violated need to be updated [22]. It is therefore computationally favorable to maintain a chronological event queue.

2.1. FPT Distribution in One Dimension

Step (2) in the above algorithm consists of sampling a FPT for a particle starting at the center of its zone at time $t = 0$. The FPT is a random sample of the probability distribution that the particle leaves its protective zone for the first time at time t. The cumulative distribution function (CDF) of the FPT distribution can then be interpreted as the probability that the particle has left the zone at or before time t. The FPT CDF has a closed form analytical solution for lattice systems. For a square pattern of surface sites, (e.g., the <100>plane of a cubic lattice), the FPT CDF in higher dimensions can be obtained from the

FPT CDF in 1D, effectively reducing the computation. Therefore it is sufficient to know the FPT CDF in 1D.

Figure 3 illustrates a random walker and its 1D protective zone of arbitrary length L_s. For illustrative purposes, L_s is set to 7 here. The FPT is the time that the particle lands outside its protective zone for the first time, i.e. the first time the particle reaches either site $L = 4$ or $L = -4$. (Note: in general $L = (L_s + 1)/2$). The particle moves with an average hopping rate of ω:

$$\omega^{(1D)} = 2D/a^2, \tag{1}$$

where D is the diffusion constant and a the lattice spacing.

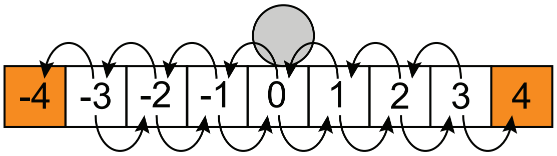

Figure 3: A particle on a 1D lattice starts at the center of its protective zone and performs a random walk until it lands outside its protective zone. The length of the protective zone $L_s = 7$ and its effective half length $L = 4$.

The particle starts at site $i = 0$ and in the next infinitesimal time step dt, it jumps to one of its neighboring sites with probability ωdt. This problem is essentially a continuous time discrete space Markov process. The distribution is thus governed by the Master Equation

$$\frac{dP_i(t)}{dt} = \frac{\omega}{2}[P_{i-1}(t) - 2P_i(t) + P_{i+1}(t)], \tag{2}$$

with initial probability distribution

$$P_i(0) = \delta_{i,0}, \tag{3}$$

where $P_i(t)$ denotes the probability that the particle is at site i at time t and sites $-L$ and L are absorbing sites for the probability of the random walker.

As shown in more detail in the Appendix, solving Equation (2) becomes equivalent to solving the discrete Laplacian equation with absorbing boundary conditions. This problem can be solved directly by means of the eigenvalues λ_i of the discrete Laplacian with pure Dirichlet boundary conditions. The calculation of the FPT CDF on a discrete lattice can thus be solved using the λ_i which are readily derived [35] and given in numerous textbooks cf.[32],

$$\lambda_j = -4\sin^2\left(\frac{\pi j}{4L}\right), \; j = 1, 2, \cdots, L_s. \tag{4}$$

$F(t)$, the FPT CDF, is the probability that the particle is at state $i = L$ or

$i = -L$ at time t. It follows that $F(t) = P_{-L}(t) + P_L(t)$. $F(t)$ is given by

$$F(t) = 2\left(1 + \sum_{i=1}^{L} \frac{\exp\left(\lambda_{2i-1}\frac{\omega}{2}t\right) + 1}{\lambda_{2i-1}\prod\limits_{j=1, j\neq i}^{L}(\lambda_{2i-1} - \lambda_{2j-1})}\right). \tag{5}$$

The derivation of this formula is presented in the Appendix.

It is important to note that $F(t)$ is a closed form solution for the FPT CDF and does not require any approximation in terms of truncatio. As such, the proposed method produces the same diffusion and nucleation statistics as a standard KMC method, because it is solving the same master equation (2) with no approxiamtion. Also note that in the extreme case of $L_s = 1$, Equation (5) reduces to the exponential function for a single lattice hop.

$$F(t)|_{L_s=1} = 1 - \exp(-\omega t). \tag{6}$$

Figure 4 shows the validity of Equation (5) by comparing FPT CDF's with simulated examples that were generated by 10,000 KMC samples for several values of L. The hopping rate has been chosen as $\omega = 5$. Since both approaches solve the same master equation, the KMC results should approach $F(t)$ for the limit of an infinite number of runs.

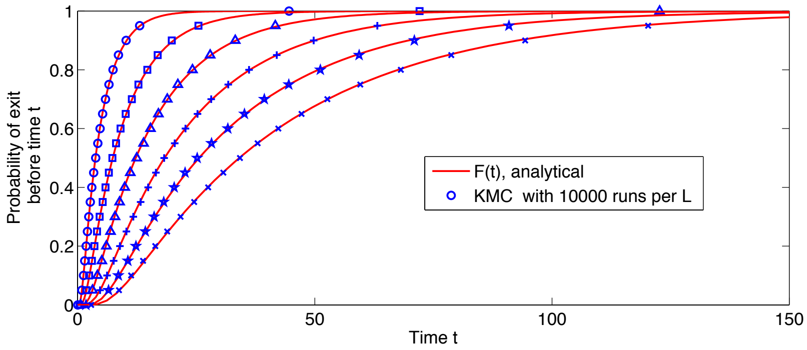

Figure 4: FPT CDF's for zones of sizes $L = 5$ (◯), 7 (□), 9 (△), 11 (+), 13 (★), and 15 (×)

2.2. *FPT in 2D*

The FPT CDF for a random walk in two dimensions can be found in a similar way as explained above, using the eigenvalues of the 2D discrete Laplacian. This is efficiently done by separation of variables, since the probability $P_{i,j}(t)$ in a random walk to find a particle at site i, j at time t can be written as

$$P_{i,j}(t) = P_i(t)P_j(t), \tag{7}$$

as the random walk in the x-direction is independent of the random walk in the y-direction. However, we are not primarily interested in the probabilities $P_{i,j}(t)$, instead we focus on the exit time probability $F(t)$ and its complement, the survival probability $S(t)$.

Consider the survival probability CDF $S(t)$, i.e. the probability that a particle has never exited its protective zone by time t. In one dimension it is simply

$$S(t) = 1 - F(t). \tag{8}$$

The probability that a particle survives a two dimensional random walk inside a rectangle is equal to the probability that it survives two independent random walks in one dimension, one in the x-direction and one in the y-direction. Thus we obtain

$$S^{(2D)}(t) = S_x^{(1D)} S_y^{(1D)}(t) = \left(1 - F_x^{(1D)}(t)\right)\left(1 - F_y^{(1D)}(t)\right). \tag{9}$$

Converting back to the FPT CDF and using Equations (8) and (9), we obtain the 2D FPT expressed in terms of the 1D FPT as

$$F^{(2D)}(t) = 1 - S^{(2D)}(t) = 1 - \left(1 - F_x^{(1D)}(t)\right)\left(1 - F_y^{(1D)}(t)\right). \tag{10}$$

The same argument can be made for a random walk inside a hyperrectangle of an n dimensional Cartesian lattice. The FPT CDF in n dimensions can be computed as:

$$F^{(nD)}(t) = 1 - \prod_{i=1}^{n}\left(1 - F_i^{(1D)}(t)\right). \tag{11}$$

If the hypercube has different side lengths in all directions, then it is necessary to compute n 1-dimensional exit time CDF's $F_i^{(1D)}(t)$ to determine the exit time CDF for the entire hypercube $F^{(nD)}$. By choosing an equilateral protection zone, i.e. a square in 2D or a cube in 3D, (11) is reduced to a function of a single 1-dimensional CDF

$$F^{(nD)}(t) = 1 - \left(1 - F^{(1D)}(t)\right)^n. \tag{12}$$

The FPT simulation requires random samples of $F(t)$ to determine a first passage time for a particle and a given zone size. In general it is not possible to invert Equation (5) to determine an FPT sample, given a uniform random number r in $[0, 1]$. However, $F(t)$ is a CDF and as such a monotonically increasing function. This observation allows the use of the method of bisection to very quickly and efficiently determine the exit time sample t^* such that $F(t^*) = r$. Using equilateral protection zones replaces the cost of up to n evaluations of $F_i^{(1D)}(t^*)$ to that of a single evaluation, but it bears the small price of an exponentiation.

2.3. Exit Location

Since we chose for the particles in 1D to always begin their random walk at the center of their zone, they are equally likely to exit the zone through either the left or the right boundary. In two or more dimensions, the question of exit location becomes a bit more intricate. First, an exit edge must be determined, and second, the exact location along the zone edge must be chosen. The former problem can be solved easily when the zones are constructed equilaterally and symmetrically around the particle. The particle is then equally likely to exit through any of its zone's boundaries. For the remainder of this work, we thus use exclusively symmetric and equilateral zones.

In 2D, suppose that a particle leaves its square protective zone in the positive x-direction. Then the location along the y-direction can be found by a conditioned 1D random walk for the y-direction, the condition being that the random walk never touched the sites $-L$ or L. The y-location just after the exit is the same as the y-location just before the exit. Thus we must first sample a y-location at the exit time. Combined with the coin flip for the x-location ($x_0 + L$ or $x_0 - L$), we then have the coordinates of the exit location. This problem can be written formally in terms of the evolution of the probability density function $P_i(t)$ already given by the diffusion master equation (2). The initial conditions are given by

$$P_i(0) = \delta_{i,0}, \tag{13}$$

and the governing equations are (compare to Figure 3):

$$\dot{P}_i(t) = \frac{\omega}{2} \begin{cases} P_{i-1}(t) - 2P_i(t) + P_{i+1}(t), & |i| < L-2 \\ P_{i-1}(t) - 2P_i(t), & i = L-1 \\ -2P_i(t) + P_{i+1}(t), & i = -L+1 \end{cases} \tag{14}$$

which can be written in matrix form as

$$\dot{\vec{P}}(t) = \frac{\omega}{2}\mathbf{A}\vec{P}(t), \tag{15}$$

where $\mathbf{A}$ is again the discrete Laplacian for a regular grid with absorbing boundaries at $-L$ and L. This system does not conserve probability because of the absorbing sites, so in order to sample the exit location we need to renormalize $\vec{P}(t)$ with the total probability of finding the particle inside the zone $\sum_i P_i(t)$ at time t. This procedure is the discrete equivalent to the "No-Passage" propagator presented in [23].

Equation (15) with initial conditions (13) has the analytic solution

$$\vec{P}(t) = \exp\{\frac{\omega}{2}\mathbf{A}t\}\vec{P}(0). \tag{16}$$

Numerically, Equation (16) can be evaluated using the eigen-decomposition of $\mathbf{A}$,

$$\vec{P}(t) = \mathbf{V}\exp\{\frac{\omega}{2}\mathbf{\Lambda}t\}\mathbf{V^T}\vec{P}(0), \tag{17}$$

where $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues λ_j of $\mathbf{A}$ on the diagonal (see Equation (4)), and $\mathbf{V}$ is the corresponding matrix of eigenvectors $\vec{v}_j$ of $\mathbf{A}$. The eigenvectors $\vec{v}_j$ also have an exact closed form solution [35]

$$v_{k,j} = \sqrt{\frac{1}{L}} \sin\left(\frac{kj\pi}{2L}\right),\ k = 1, 2, \cdots, L_s, \tag{18}$$

where the subscripts k, j denote the k-th entry of the eigenvector $\vec{v}_j$ corresponding to the eigenvalue λ_j.

With equations (4),(18), and (17) it is now possible to determine the conditional distribution for the y-location of the particle, given that it exited its zone in the x-direction at time t:

$$\tilde{P}_i(t) = \frac{\vec{P}_i(t)}{S(t)}. \tag{19}$$

$S(t)$ normalizes $\vec{P}(t)$ such that $\sum_i \tilde{P}_i(t) = 1$ due to the fact that $S(t) = \sum_i P_i(t)$. It is technically not needed for sampling the exit location, but reported here for mathematical completeness. If the problem is in n dimensions, then $n - 1$ samples of $\tilde{P}_i(t)$ need to be drawn, one for every dimension except the exit dimension.

2.4. No-Passage Propagator

Particles whose zones become violated (steps 5.a.ii and 5.b.iii) during the execution of the algorithm need to update their position inside their zone before a new zone can be constructed around the particle. This event is referred to as No-Passage (NP) event in [23]. The NP propagator for a particle starting at the center of a symmetric equilateral domain is straightforward and relies on the same calculations that are used for the exit location. Equation (19) can be used to sample n coordinates inside the particle's zone, with t being the time that the particle has resided in the zone until resampling of its new position.

2.5. Nucleation and Island Boundaries

By choosing zones to be non-overlapping and exclusive to each particle, the algorithm reduces to a standard random walk KMC when two or more particles are close to each other. Once particles are three or fewer sites apart from each other, their zone sizes reduce to their current site. The FPT then reduces to movement of single lattice hops with an exponentially distributed time between hops with mean hopping rate ω, as mentioned previously in Equation (6) - essentially a KMC movement of particles.

The same holds for the edges of larger islands. Once a particle gets close enough to an island, its movement gradually reduces to a KMC random walk until it either attaches to the island or it moves further away again where it can have a larger zone. The algorithm can thus be seamlessly integrated with any physical phenomena that occur at the island edges. In particular, edge diffusion and detachment kinetics can be incorporated in the same manner as in standard

KMC methods. It is also possible to incorporate a 1D FPKMC method for the edge diffusion as presented in [21]. Additional phenomena may be included, as long as the events do not destroy the integrity of the zones of the particles. Possible phenomena are detachment from the edge or Schwöbel-Ehrlich barriers in the case of multi-layer deposition[17]. There will be no overhead in compute time by incorporating these additional features, compared to implementing them for a KMC method. It is also possible to implement absorbing and reflecting boundaries using this feature of zone size reduction to KMC hops, but these boundary conditions will artificially slow down the ELFPT simulation somewhat, due to the greater resolution of the diffusive motion of particles at domain boundaries.

Based on these observations it is evident that the algorithm works best in situations with few particles and a low edge density on the surface. The speedup is generated from situations where particles are far away from each other. In particular, it is extremely efficient for the determination of island size distributions in cases of epitaxial growth where the mean-free-path length of particles is high. In such situations the surface diffusivity D is high compared to the influx rate F of particles. Furthermore, it is exact in the sense that it fully replicates the statistical results from standard KMC simulations. The ELFPT method is efficient where many other simulation methods for epitaxial growth struggle: the determination of island size distributions without prior knowledge of nucleation rates while maintaining atomistic resolution of the island shapes. We have performed a few studies on island size distribution with and without edge diffusion, and we report on the performance of the new algorithm compared to a standard KMC algorithm.

3. Computational Results and Discussion

The following experiments have been performed on a 2.5GHz Quad Core Intel desktop computer with 4GB RAM memory running Ubuntu 11.10. The codes are written in C++. The codes ran on one core only for both the ELFPT and the KMC simulations, so no speed-up from multi-threading or parallelization is involved in these results.

3.1. Diffusion and Nucleation for Different D/F

The setup for the first example is the following: The surface is seeded with an initial deposit site at the center. Particles subsequently appear at random locations on the surface at a mean rate of $F = 1$. The particles perform a random walk with diffusion rate D on the surface. It is assumed that particles that land at a site adjacent to a deposit will be turned into deposit immediately and irreversibly, i.e. they cannot detach in the future. In this first example, it is further assumed that there is no diffusion of particles along island edges. The simulation is terminated when 10% of the surface is covered with deposit. Runtimes for ELFPT and KMC are reported in Table 1. Both of the methods ELFPT and KMC were run with the same code. It was written such that if

L_{max} is set to 1 by the user, the code reverts to a standard KMC method and avoids neighborhood searches and list updates. As such, the KMC version of the code contains a few unnecessary *if* statements, but we estimate this overhead to be negligible compared to the generation of random numbers and searches through lists and arrays.

Numerical simulations are typically compared with experimental observations of early stages of nucleation and growth such as may be obtained by atomic force microscopy (AFM). Such measurements are carried out on regions that are $1 - 2\mu m^2$, which corresponds to about 2000x2000 to 4000x4000 lattice sites for metal surfaces with lattice spacings on the order of $250nm$ to $500nm$. Rates range from $(D/F) < 10^2$ [28] to $(D/F) >> 10^8$, above which standard KMC methods become prohibitively expensive [38] . For this experiment we varied the surface diffusion rate of particles from $D/F = 10^6$ to $D/F = 10^{15}$. Preliminary runs suggested that the optimal speedup was obtained by limiting the maximum allowable size of the protection zones to a value, L_{max}, that was found heuristically, and which depended on simulation domain size and D/F ratio. At present, the value of L_{max} which yields optimal speedup must be found heuristically, and depends on simulation domain size as well as the D/F ratio. We varied L_{max} from 8 to 64 for all cases, which correspond to maximal zone sizes of 15x15, 31x31, 63x63, and 127x127. The runtimes of the different cases are reported in Table 1. The reason why largest possible zone sizes are not always optimal lies in the overhead that is spent while searching the neighborhood of particles for edge sites or other particles. We performed the same simulation with a standard KMC by setting the L_{max} value to 1, reducing the FPT approach to a standard KMC simulation where the neighborhood search for edges and other particles is reduced to direct neighbors. This experiment serves as a simple test case to compare the performance improvement in 2D diffusion for ELFPT compared to a standard KMC approach. As expected, the optimal speedup is achieved for the largest domains and highest values of D/F.

The results in Table 1 show that if L_{max} is kept constant for ELFPT and the domain size is varied, then runtimes for both algorithms scale approximately linearly with the size of the domain (number of sites) for any given D/F. However, when comparing scaling over D/F, ELFPT shows much better scaling than KMC. In fact, for higher values of L_{max}, ELFPT scales constant with D/F for the tested domain sizes, whereas KMC ($L_{max} = 1$) scales approximately with $(D/F)^{1/3}$ for the large domain sizes. This corresponds well with the findings from nucleation theory that predict an island density (N) scaling of $N \propto (D/F)^{-1/3}$ [37, 26].

Runtimes for KMC level off for small domain sizes and high D/F values. This stems from the fact that for high D/F there is typically only one particle in the domain at a time. This particle diffuses very fast and attaches to the deposit before another particle appears in the domain. The total number of diffusive KMC hops during a simulation remains independent of D/F for those cases; only the simulation time increments change with D/F. The chance of having two particles diffusing at the same time is very limited in such cases, and consequentially the chance for nucleation is minimal. This leads to growth modes where only one island will grow in the simulation domain.

As may be seen in Table 1, the choice of the ideal maximum protection zone size depends on the simulation parameters. Figure 5 shows the speedup that FPT obtains versus KMC. The colors of the bars correspond to the optimal L_{max} value. The best speedup was found for high values of D/F and large domain sizes. An observable trend predicts that larger L_{max} are more favorable for simulations of larger domains and high D/F ratios. Further investigation will be needed to determine to optimal value of L_{max} *a priori*.

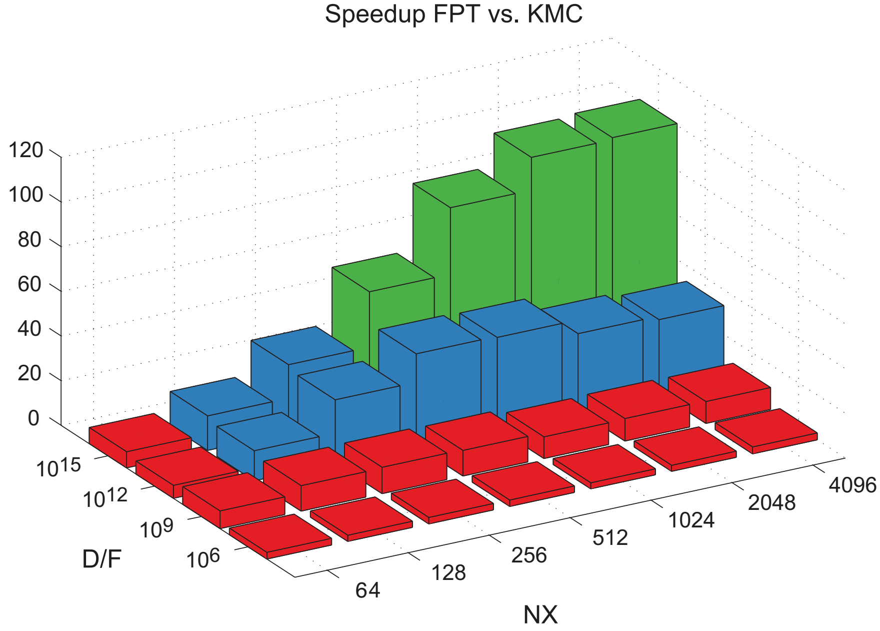

Figure 5: Speedup factors of FPT algorithm vs. KMC. The colors correspond to the optimal L_{max} value for the given case: Red: $L_{max} = 8$; Blue: $L_{max} = 16$; Green: $L_{max} = 32$.

3.2. Edge Diffusion

In order to be able to simulate the physically more realistic situation where particles can diffuse along island edges, we have included edge diffusion at different rates. The mechanism is the following: particles that are attached to islands with one lateral bond are allowed to move along the edge of the island until they encounter either another edge-walker or they land in a kink site where they have two or more lateral bonds. In both cases the particles become immobile and so part of the island. Since particles moving along edges do not influence reaction and diffusion propensities, edge diffusion can be executed synchronously after every flux or diffusion event. Our method consisted of letting particles with one bond to islands hop with a mean rate of $\omega_{edge} = D_{edge}/2a^2$. After every flux or surface diffusion step, all edge particles are moved in a KMC fashion until they land in a kink site, collide with another edge particle or the time of the flux or surface diffusion event is reached.

Situations arise during simulation with high D_{edge}/F where islands are perfectly rectangular and have no kinks. In such cases it is possible that the edge particles circle around the island many times until the time of the next flux or surface diffusion event is reached. We therefore terminated edge diffusion of individual particles after they performed 1000 edge diffusion steps. This can introduce some error, but we estimate the error to be small since after 1000 hops, the probability distribution of an edge-particle can be assumed to be nearly uniform around the island. Furthermore, those edge walkers will continue moving along the edge after the next flux or surface diffusion event. This procedure illustrates how additional phenomena can be seamlessly introduced by means of standard KMC. The added features can reduce the efficiency of ELFPT, however their addition is not likely to render ELFPT less efficient than KMC. Figure 6 shows how the island sizes and shapes change with different D/F and D_{edge}/F ratios. The snapshots in the top row of Figure 6 all show an island at the center of the image. These islands are the islands that grew from the initial seed. All other islands grew from 'wild' nuclei, which resulted from diffusing particles.

3.3. Exclusion Zone in Front of Step-Edge

As a third example, we investigate the experimentally observable effect of an exclusion zone in front of step-edges [18]. Theoretical results and experimental observations [19] predict that nucleation is less likely to occur close to step-edges. Far away from a feature like a step-edge, all sites are equally likely to be covered with deposit, that is if the overall coverage is 20%, then there is a 20% chance that a site is covered with deposit. Close to an edge, however, a particle is more likely to hit the edge and become part of the growing front than it is to find another particle for nucleation. Sites very close to the initial location of the edge are therefore very likely to be covered with deposit at the end. There exists a zone between the edge front and the far field where particles are unlikely to nucleate, because the edge acts as a sink for nearby particles. This zone between the edge front and the far field is called the exclusion zone. It is of practical interest to know how far the influence of the edge extends into the domain. Stephens et al. predicted regimes at which particles are very unlikely to nucleate between step-edges [30]. We are interested in statistical results for those predictions to quantify the probability of particles nucleating at a given distance from the advancing step-edge.

For the numerical experiment, we seeded the left border of our domain with an entire row of deposit, acting like a step-edge. The edge diffusion rate D_{edge} was kept equal to the surface diffusion rate D for this experiment. The system was then evolved with the same rules as in the first example, until 20% of the surface was covered with deposit. Snapshots of the simulations with different D/F values are shown in Figure 7. To obtain the statistical data, we ran 1000 independent runs at different D/F ratios. It was therefore crucial to have an

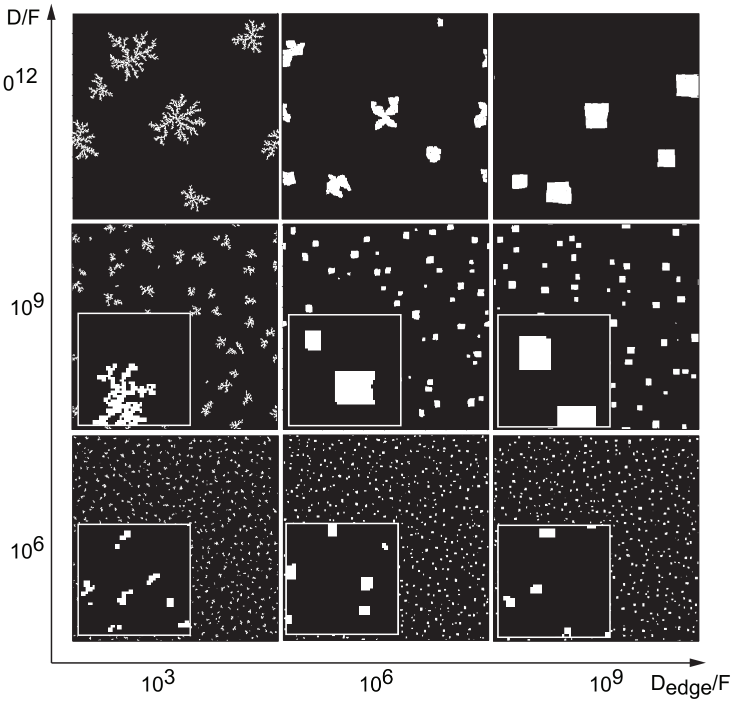

Figure 6: Snapshots of deposit (white) for different values of D/F and D_{edge}/F after deposition of 0.1 mono-layers. Nucleation density decreases with increasing D/F, and island shapes are more compact for high D_{edge}/F. All domain sizes are 512x512 lattice sites. The inserts in the bottom two rows are magnified 10 times to show more detail.

efficient algorithm that can handle all D/F ratios as well as the sizes necessary to fully resolve the exclusion zone without effects from the domain boundary on the opposite site of the edge. We recorded the locations of deposit for each D/F over 1000 runs and calculated the probability of the domain being covered with deposit at a given distance from the edge. The probability distributions are shown in Figure 8.

The exclusion zone is not very apparent near the edge (left side of image) for a value of D/F of 10^3 in the snapshots of Figure 7. For higher surface diffusion rates, one can clearly see that nucleation is less likely to happen in the vicinity of the step-edge. Figure 8 shows the resulting probabilities from our ensembles of simulations. For a lower ratio of D/F the exclusion zone does not extend far into the domain, and it is less pronounced than for larger values of D/F. For large D/F ratios, the edge is more likely to grow into the domain, i.e. particles are more likely to attach to the edge, which is in agreement with the larger islands seen in the first experiment. Nonetheless, all ratios of D/F exhibit an exclusion zone, confirming physical observations [18, 19] and previous calculations [30].

To demonstrate that ELFPT is statistically equivalent to KMC, we have also run the same experiment with our KMC code and plotted the results in Figure 8. In this case, ELFPT was up to 12x faster than KMC. ELFPT speedup was compromised somewhat by our standard KMC treatment of the edge diffusion. It is planned to adapt a FPT approach for edge diffusion in a future version of the code to overcome this limitation. As Nandipati et al. have shown [21], edge diffusion is well suited for an FPT approach and can yield considerable speedup.

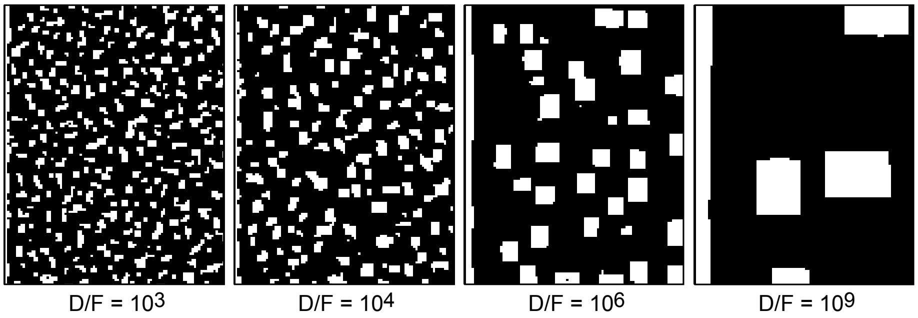

Figure 7: Snapshots of deposits (white) at 20% coverage. Shown here are 100x128 windows of the simulation which was 256x256 sites. Initially, the simulation domain is seeded with deposits along the left boundary of the domain.

4. Conclusions

We have presented a highly efficient, exact method to simulate and study reaction-diffusion-nucleation systems on a lattice. Examples have been given

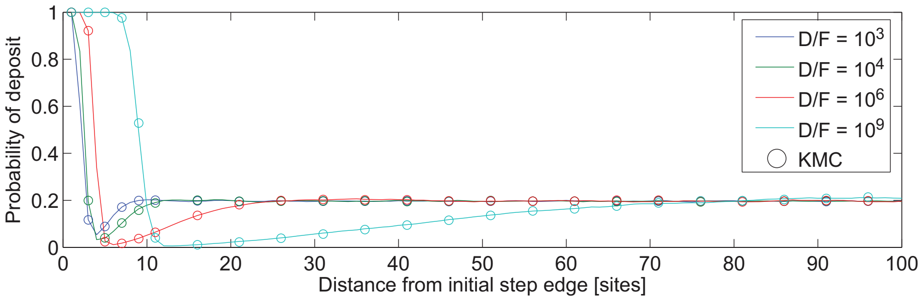

Figure 8: Probability of deposit being located at distance x from a step-edge. The exclusion zone is larger for larger D/F. (Dip in probability of deposit in front of growing edge.)

for physical problems in two dimensions. We outlined how the algorithm can also be efficiently applied to systems in higher dimensions. Our method requires no *a priori* knowledge of nucleation rates or attachment rates to edges. The new method produces the same statistical results as KMC simulations but with a very substantial speedup when nucleation densities are low and mean free paths of particles are high. Fast and accurate simulations of this system allow for the estimation of reaction rate parameters involved in thin film depositions. The new ELFPT method yields substantial speedup compared to conventional KMC methods and it scales well with increasing ratios of D/F. The reduction of the FPKMC to standard KMC near nucleation sites and edges allows for the incorporation of a multitude of additional physical effects. As an example we have shown that edge diffusion can be seamlessly incorporated. Our approach is not limited to square lattices and sub-monolayer systems. We will report results for the hexagonal lattice as well as multi-layer growth in a future publication.

Acknowledgments

This research was supported through grant DE-FG02-12ER26085 from the U.S. Department of Energy.

[1] 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, Jan. 1975.

[2] R. D. Braatz, E. G. Seebauer, and R. C. Alkire. *Multiscale Modeling and Design of Electrochemical Systems*, pages 289–334. Wiley-VCH Verlag GmbH & Co. KGaA, 2008.

[3] R. E. Caflisch, M. F. Gyure, B. Merriman, S. Osher, C. Ratsch, D. D. Vvedensky, and J. Zinck. Island dynamics and the level set method for epitaxial growth. *Appl. Math. Lett.*, 12(4):13–22(10), May 1999.

[4] A. Chatterjee and D. Vlachos. An overview of spatial microscopic and accelerated kinetic Monte Carlo methods. *J. Comput.-Aided Mater. Des.*, 14(2):253–308, July 2007.

[5] S. Chen, B. Merriman, M. Kang, R. E. Caflisch, C. Ratsch, L.-T. Cheng, M. Gyure, R. P. Fedkiw, C. Anderson, and S. Osher. A level set method for thin film epitaxial growth. *J. Comput. Phys.*, 167(2):475 – 500, 2001.

[6] C.-C. Chou and M. L. Falk. Multiscale diffusion Monte Carlo simulation of epitaxial growth. *J. Comput. Phys.*, 217(2):519–529, Sept. 2006.

[7] J. P. DeVita, L. M. Sander, and P. Smereka. Multiscale kinetic Monte Carlo algorithm for simulating epitaxial growth. *Phys. Rev. B*, 72(20):205421–, Nov. 2005.

[8] T. O. Drews, R. D. Braatz, and R. C. Alkire. Parameter sensitivity analysis of monte carlo simulations of copper electrodeposition with multiple additives. *J. Electrochem. Soc.*, 150(11):C807–C812, 2003.

[9] T. O. Drews, J. C. Ganley, and R. C. Alkire. Evolution of surface roughness during copper electrodeposition in the presence of additives. *J. Electrochem. Soc.*, 150(5):C325–C334, May 2003.

[10] J. Fu, S. Wu, and L. Petzold. Time dependent solution for acceleration of tau-leaping. *Journal of Computational Physics*, 2012. To appear.

[11] M. Giesen. Step and island dynamics at solid/vacuum and solid/liquid interfaces. *Progress in Surface Science*, 68(1-3):1–154, Sept. 2001.

[12] S. Gill, P. Spencer, and A. Cocks. A hybrid continuum/kinetic Monte Carlo model for surface diffusion. *Mater. Sci. Eng., A*, 365(1-2):66–72, Jan. 2004.

[13] D. T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22(4):403–434, Dec. 1976.

[14] M. F. Gyure, C. Ratsch, B. Merriman, R. E. Caflisch, S. Osher, J. J. Zinck, and D. D. Vvedensky. Level-set methods for the simulation of epitaxial phenomena. *Phys. Rev. E*, 58(6):R6927–R6930, Dec. 1998.

[15] B. Harald. Microscopic view of epitaxial metal growth: nucleation and aggregation. *Surf. Sci. Rep.*, 31(4–6):125–229, 1998.

[16] T. Jahnke and D. Altıntan. Efficient simulation of discrete stochastic reaction systems with a splitting method. *BIT Numerical Mathematics*, 50:797–822, 2010.

[17] S. J. Liu, H. Huang, and C. H. Woo. Schwoebel-Ehrlich barrier: from two to three dimensions. *Appl. Phys. Lett.*, 80(18):3295–3297, Mar. 2002.

[18] A. Milchev. *Electrocrystallization: Fundamentals of Nucleation and Growth.* p.195, Kluwer Academic Publishers, 2002.

[19] A. Milchev. Electrocrystallization: Nucleation and growth of nano-clusters on solid surfaces. *Russ. J. Electrochem.*, 44:619–645, 2008. 10.1134/S1023193508060025.

[20] C. Min and F. Gibou. A second order accurate level set method on non-graded adaptive cartesian grids. *J. Comput. Phys.*, 225(1):300–321, July 2007.

[21] G. Nandipati, Y. Shim, and J. G. Amar. First-passage time approach to kinetic Monte Carlo simulations of metal (100) growth. *Phys. Rev. B*, 81(23):235415–, June 2010.

[22] T. Oppelstrup, V. V. Bulatov, A. Donev, M. H. Kalos, G. H. Gilmer, and B. Sadigh. First-passage kinetic Monte Carlo method. *Phys. Rev. E*, 80(6):066701–, Dec. 2009.

[23] T. Opplestrup, V. V. Bulatov, G. H. Gilmer, M. H. Kalos, and B. Sadigh. First-passage Monte Carlo algorithm: Diffusion without all the hops. *Phys. Rev. Lett.*, 97(23):230602–, Dec. 2006.

[24] R. Ramaswamy and I. F. Sbalzarini. Exact on-lattice stochastic reaction-diffusion simulations using partial-propensity methods. *The Journal of Chemical Physics*, 135(24):244103, 2011.

[25] C. Ratsch, M. F. Guyre, R. E. Caflisch, F. G. Gibou, M. Petersen, M. Kang, J. Garcia, and D. D. Vvedensky. Level-set method for island dynamics in epitaxial growth. *Phys. Rev. B*, 65:195403, 2002.

[26] C. Ratsch, A. Zangwill, P. Šmilauer, and D. D. Vvedensky. Saturation and scaling of epitaxial island densities. *Phys. Rev. Lett.*, 72:3194–3197, May 1994.

[27] S. Redner. *A Guide to First-Passage Processes.* Cambridge University Press, 2001.

[28] M. Schroeder and D. E. Wolf. Magic Islands and Submonolayer Scaling in Molecular Beam Epitaxy. *Phys. Rev. Lett.*, 74:2062–2065, Mar. 1995.

[29] T. P. Schulze, P. Smereka, and W. E. Coupling kinetic monte-carlo and continuum models with application to epitaxial growth. *J. Comput. Phys.*, 189(1):197–211, July 2003.

[30] R. M. Stephens and R. C. Alkire. Simulation of kinetically limited nucleation and growth at monatomic step edges. *J. Electrochem. Soc.*, 154(8):D418–D426, Aug. 2007.

[31] R. M. Stephens and R. C. Alkire. Island dynamics algorithm for kinetically limited electrochemical nucleation of copper with additives onto a foreign substrate. *J. Electrochem. Soc.*, 156(1):D28–D35, Jan. 2009.

[32] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* SIAM, second edition, 2004.

[33] Y. Sun, R. Caflisch, and B. Engquist. A multiscale method for epitaxial growth. *Multiscale Model. Simul.*, 9(1):335–354, Jan. 2011.

[34] V. I. Tokar and H. Dreyssé. Accelerated kinetic Monte Carlo algorithm for diffusion-limited kinetics. *Phys. Rev. E*, 77:066705, June 2008.

[35] C. Van Den Broeck, W. Horsthemke, and M. Malek-Mansour. On the diffusion operator of the multivariate master equation. *Physica A*, 89(2):339–352, Nov. 1977.

[36] J. S. van Zon and P. R. ten Wolde. Simulating biochemical networks at the particle level and in time and space: Green's function reaction dynamics. *Phys. Rev. Lett.*, 94(12):128103–, Apr. 2005.

[37] J. A. Venables, G. D. T. Spiller, and M. Hanbucken. Nucleation and growth of thin films. *Rep. Prog. Phys.*, 47(4):399, 1984.

[38] D. Vvedensky. Atomistic modeling of epitaxial growth: comparisons between lattice models and experiment. *Comput. Mat. Sci.*, 6(2):182 – 187, 1996.

Appendix: Derivation of Equation (5)

Note: Equations (20) to (30) are taken from [27, p. 72].

Consider a random walker starting at site 0 and hopping either left or right with a hopping rate ω. When the particle reaches either site L or $-L$ it will be absorbed by that site, or in other words, sites L and $-L$ are traps. The total length of the domain is thus $2L+1$. Figure 9 illustrates the problem for $L = 4$.

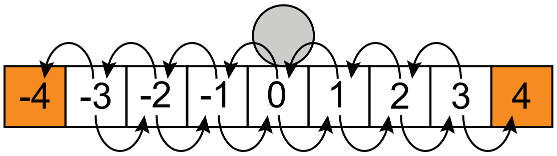

Figure 9: A domain with 9 sites, $L = 4$. The arrows indicate transition possibilities.

For a domain with traps at sites -L and L we can write the following master equations:

$$
\begin{aligned}
\dot{P}_0(t) &= \frac{\omega}{2}[P_{-1}(t) - 2P_0(t) + P_1(t)], && (20)\\
\dot{P}_1(t) &= \frac{\omega}{2}[P_0(t) - 2P_1(t) + P_2(t)], && (21)\\
&\vdots && (22)\\
\dot{P}_{L-1}(t) &= \frac{\omega}{2}[P_{L-2}(t) - 2P_{L-1}(t)], && (23)\\
\dot{P}_L(t) &= \frac{\omega}{2}[P_{L-1}(t)]. && (24)
\end{aligned}
$$

The factor of 1/2 stems from the fact that half of the time the particle jumps to the left, and half the time it jumps to the right. Due to this symmetry we have $P_{-i}(t) = P_i(t)$. Setting the hopping rate to 1 by the time rescaling $\bar{t} \to \frac{\omega}{2}t$ and taking the Laplace transform of above equations we obtain a set of algebraic equations:

$$
\begin{aligned}
sP_0(s) - P_0(\bar{t}=0) &= P_{-1}(s) - 2P_0(s) + P_1(s), && (25)\\
sP_1(s) - P_1(\bar{t}=0) &= P_0(s) - 2P_1(s) + P_2(s), && (26)\\
&\vdots && (27)\\
sP_{L-1}(s) - P_{L-1}(\bar{t}=0) &= P_{L-2}(s) - 2P_{L-1}(s), && (28)\\
sP_L(s) - P_L(\bar{t}=0) &= P_{L-1}(s). && (29)
\end{aligned}
$$

The initial conditions are $P_i(\bar{t}=0) = \delta_{i,0}$. Combining that with the fact that

$P_{-i} = P_i$ we can write the above equations in matrix form as

$$\begin{bmatrix} s+2 & -2 & 0 & \cdots & 0 \\ -1 & s+2 & -1 & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & -1 & s+2 & 0 \\ 0 & 0 & \cdots & -1 & s \end{bmatrix} \begin{bmatrix} P_0(s) \\ P_1(s) \\ \vdots \\ P_{L-1}(s) \\ P_L(s) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{30}$$

We are interested in solving Equations (30) for the absorbing state $P_L(s)$. Denote the big system matrix in Equation (30) by $\mathbf{A}$. $\mathbf{A}$ is a square matrix of size $n = L+1$. In order to solve for $P_L(s)$ we need only the $(n,1)$ entry of $\mathbf{A}^{-1}$, due to the structure of the right hand side. $\mathbf{A}^{-1}_{n,1}$ can be obtained by finding the adjugate of $\mathbf{A}$:

$$\mathbf{A}^{-1} = \frac{1}{det(\mathbf{A})} adj(\mathbf{A}). \tag{31}$$

The transpose of the adjugate of $\mathbf{A}$ is the matrix $\mathbf{C}$ of cofactors of $\mathbf{A}$. Thus we need to find $C_{1,n}$. We are now going to show that $C_{1,n}$ is always equal to 1 for the system (30).

First we write $C_{i,j}$ as a function of the corresponding (i,j)-minor of $\mathbf{A}$:

$$C_{i,j} = (-1)^{i+j} M_{i,j}. \tag{32}$$

The (i,j)-minor, denoted by $M_{i,j}$, of an $n \times n$ square matrix $\mathbf{A}$ is defined as the determinant of the $(n-1) \times (n-1)$ matrix formed by removing from $\mathbf{A}$ its i-th row and j-th column. Coming back to system (30), we write $M_{1,n}$ as:

$$M_{1,n} = \begin{vmatrix} -1 & s+2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & s+2 & -1 & \cdots & 0 \\ & & \ddots & & & \\ 0 & 0 & \cdots & 0 & -1 & s+2 \\ 0 & 0 & \cdots & 0 & 0 & -1 \end{vmatrix}, \tag{33}$$

which is the determinant of an upper triangular matrix with only (-1) on the diagonal. The size of the matrix is $n-1$, thus

$$M_{1,n} = (-1)^{n-1}, \tag{34}$$

and Equation (32) becomes

$$C_{1,n} = (-1)^{1+n}(-1)^{n-1} \equiv 1 \tag{35}$$

for all $n > 0$.

Combining this result with Equation (31), we conclude that

$$A^{-1}_{n,1} = \frac{1}{det(\mathbf{A})}. \tag{36}$$

Thus $P_L(s)$ is equivalent to the inverse of the determinant of the system matrix $\mathbf{A}$. $\mathbf{A}$ can be written in the form $\mathbf{A} = (sI - \tilde{\mathbf{A}})$, with

$$\tilde{\mathbf{A}} = \begin{bmatrix} -2 & 2 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ & & \ddots & & \\ 0 & \cdots & 1 & -2 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}. \tag{37}$$

The determinant of $\mathbf{A}$ then becomes the characteristic polynomial of $\tilde{\mathbf{A}}$. The characteristic polynomial of a matrix can also be written in terms of the matrix eigenvalues $\tilde{\lambda}_{\mathbf{i}}$. The last column of $\tilde{\mathbf{A}}$ consists of only zeros, thus one of the eigenvalues of $\tilde{\mathbf{A}}$ is zero and the other L eigenvalues turn out to be non-zero. We can now write

$$A^{-1}_{n,1} = \frac{1}{det(\mathbf{A})} = \frac{1}{s\prod_{i=1}^{L}(s - \tilde{\lambda}_i)}. \tag{38}$$

In a final step, we point out that the eigenvalues of $\tilde{\mathbf{A}}$ are a subset of the eigenvalues of the discrete Laplacian on a regular Cartesian lattice of size $L_s = 2L + 1$. The connection can be made also through the fact that $P_{-i} = P_i$ in the derivation above. The entire system is symmetric around site 0 and therefore only the odd eigenvalues of the discrete Laplacian are needed. It turns out that the odd eigenvalues of the discrete Laplacian λ_i are also the eigenvalues $\tilde{\lambda}$ of $\tilde{\mathbf{A}}$. $\tilde{\lambda}_i$. Thus we can finally write

$$P_L(s) = \frac{1}{s\prod_{i=1}^{L}(s - \lambda_{2i-1})}, \tag{39}$$

now using the eigenvalues λ_i of the discrete Laplacian instead of those of matrix $\tilde{\mathbf{A}}$. The inverse Laplace transform $\mathcal{L}^{-1}[P_L(s)]$ can be computed analytically. First we denote $\lambda_{-1} = 0$ as the zero'th eigenvalue of $\mathbf{A}$ (corresponding to the zero eigenvalue of $\tilde{\mathbf{A}}$). Equation (39) can then be rewritten as

$$P_L(s) = \frac{1}{\prod_{i=0}^{L}(s - \lambda_{2i-1})}. \tag{40}$$

Before proceeding with the inverse Laplace transform $\mathcal{L}^{-1}[P_L(s)]$ it is important to note that all λ_{2i-1} are real and unique for $i = 0, 1, \ldots, L$ (λ_{-1} is zero by definition. For $i = 1, \ldots, L$, the arguments to the sine function in $\lambda_{2i-1} = -4\sin^2\left(\frac{\pi(2i-1)}{4L}\right)$ are unique, real, positive and $< \pi/2$. Therefore, λ_{2i-1} are real and unique for $i = 0, \ldots, L$). It is thus possible to invert Equation (40) by method of partial fractional expansion. We start by writing (40) as

$$P_L(s) = \sum_{i=0}^{L} \frac{c_i}{s - \lambda_{2i-1}} \tag{41}$$

The residues of $P_L(s)$, c_i, are determined via the formula

$$c_i = \left[(s - \lambda_{2i-1})P_L(s)\right]_{s=\lambda_{2i-1}}, \; i = 0, \ldots, L. \tag{42}$$

The calculation of $\mathcal{L}^{-1}[P_L(s)]$ becomes straight forward and we can write

$$P_L(\bar{t}) = \left(\sum_{i=0}^{L} c_i \exp\left(\lambda_{2i-1}\bar{t}\right)\right) u(\bar{t}) \tag{43}$$

where $u(\bar{t})$ is the Heaviside step function.

After some algebraic clean-up and remembering that we used the time rescaling $\bar{t} \to \frac{\omega}{2}t$, we can finally write for $t > 0$

$$P_L(t) = 1 + \sum_{i=1}^{L} \frac{\exp\left(\lambda_{2i-1}\frac{\omega}{2}t\right) + 1}{\lambda_{2i-1} \displaystyle\prod_{j=1, j\neq i}^{L} \left(\lambda_{2i-1} - \lambda_{2j-1}\right)}. \tag{44}$$

The probability of the particle being absorbed at either site $-L$ or L by time t is the sum of $P_{-L}(t)$ and $P_L(t)$ which is simply $2P_L(t)$.

Table 1: Runtimes for different domain sizes and different D/F values. Shaded grey cells highlight KMC runtimes and best ELFPT runtimes.

CPU runtimes [sec]

Domain Size (# sites)	$\mathbf{L_{max}}$	$\mathbf{\frac{D}{F} = 10^{6}}$	$\mathbf{\frac{D}{F} = 10^{9}}$	$\mathbf{\frac{D}{F} = 10^{12}}$	$\mathbf{\frac{D}{F} = 10^{15}}$
64x64 (4096)	KMC	0.023	0.099	0.097	0.101
	8	0.008	0.013	0.017	0.014
	16	0.017	0.017	0.017	0.016
	32	0.062	0.046	0.046	0.049
	64	-	-	-	-
128x128 (16,384)	KMC	0.094	0.791	1.392	1.569
	8	0.034	0.069	0.114	0.112
	16	0.071	0.075	0.104	0.101
	32	0.214	0.193	0.206	0.187
	64	0.691	0.646	0.639	0.652
256x256 (65,536)	KMC	0.406	3.350	16.87	18.49
	8	0.140	0.285	1.395	1.363
	16	0.292	0.321	0.594	0.606
	32	0.805	0.799	0.836	0.841
	64	2.601	2.560	2.563	2.560
512x512 (262,144)	KMC	1.710	13.63	118.4	250.3
	8	0.600	1.185	8.879	16.706
	16	1.239	1.396	2.879	5.407
	32	3.391	3.191	3.768	4.522
	64	11.53	10.35	11.01	11.26
1024x1024 (1,048,576)	KMC	7.241	50.98	514.2	2,562
	8	2.643	5.239	33.85	203.0
	16	5.551	6.081	12.65	63.39
	32	15.15	13.70	15.39	30.10
	64	46.98	45.09	49.29	53.31
2048x2048 (4,194,304)	KMC	31.53	219.7	1,812	15,663
	8	11.49	21.87	135.1	867.8
	16	25.67	25.80	52.55	310.6
	32	60.48	56.59	64.81	156.9
	64	199.6	193.9	206.7	242.8
4096x4096 (16,777,216)	KMC	175.0	925.7	7,174	59,754
	8	56.53	94.42	519.1	3,782
	16	116.2	105.7	218.1	1,223
	32	292.1	246.2	276.4	592.8
	64	838.1	811.0	837.3	986.3