Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 217 (2006) 519–529
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Multiscale diffusion Monte Carlo simulation of epitaxial growth

Chuan-Chih Chou, Michael L. Falk *

Department of Materials Science and Engineering, University of Michigan, 2300 Hayward Street, Ann Arbor, MI 48109-2136, USA

Received 28 June 2005; received in revised form 1 December 2005; accepted 7 January 2006
Available online 24 February 2006

Abstract

We present an accelerated kinetic Monte Carlo (KMC) simulation algorithm for molecular beam epitaxial growth during step flow. The acceleration is achieved by allowing adatoms far from the step edges to execute larger jumps with correspondingly reduced rates. The computational complexities of a number of different algorithmic implementations are analyzed and compared. The scaling of the accelerated algorithms are verified by test runs performed on a one-dimensional model. Both performance and accuracy are evaluated. One to two orders of magnitude increases of efficiency are achieved while preserving physical accuracy.

Keywords: Kinetic Monte Carlo; Multiscale methods; Epitaxy; Deposition; Step flow; Computational complexity; Diffusion

1. Introduction

This paper describes an accelerated kinetic Monte Carlo (KMC) simulation algorithm for molecular beam epitaxial (MBE) growth in the step flow regime. KMC simulation has been used extensively for the modeling of epitaxial growth [1–6], and has been shown to reproduce quantitatively accurate results including fluctuations. Capturing these important atomic scale details is beyond the reach of continuous Burton–Cabrera–Frank models [7] and limits their ability to model the detailed processes that control surface evolution during MBE. However, the atomistic nature of KMC simulation and the high diffusion rate typical in these systems impose significant computational cost, which severely restricts the spatial and temporal scale of the simulation. Recently some new approaches have been developed to overcome these limitations. Some of these methods couple stochastic KMC simulation to continuous BCF models in order to expand the scale of simulation [8–10]. Others introduce fluctuations into continuum methods [11]. Still others attempt to derive fully coarse grained equations from atomistic model by treating probability densities [12–14]. These methods have shown some considerable success in grappling with the complexity of coupling these two extremely different simulation schemes. However, these complications often limit the degree of acceleration achieved and reduce the

* Corresponding author. Tel.: +1 734 615 8086; fax: +1 734 763 4788.
E-mail address: mfalk@umich.edu (M.L. Falk).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.01.012

accuracy of simulated fluctuations. In most cases the computational intensity of these algorithms has only been reported anecdotally and the scaling of the algorithms has not been analyzed.

The algorithm we present is motivated by a method developed in the context of dendritic solidification by Plapp and Karma [15] who were in turn inspired by techniques applied in quantum Monte Carlo simulations. In that algorithm an ensemble of off-lattice random walkers are used to simulate a diffusion field. In our algorithm, adatoms remain on-lattice and represent actual adsorbed species, but, as in the Plapp and Karma work, we take advantage of the fact that the farther these random walkers reside from the boundary of interest (in our case the step edge, in theirs the solidification front) the less temporal resolution is required in simulating their motion. In this respect the method is similar to a method developed simultaneously by DeVita et al. which also treats the motion of ad atoms by a similar approximation [16]. This results in major gains in computational efficiency because the adatoms of primary physical interest, those adjacent to the step edge, only comprise a small fraction of the total adatoms in the system. The fluctuations of these atoms are the most critical for accurately modeling the step flow process. By requiring that adatoms far away from the step edges take larger jumps with rates proportional to the inverse square of the step size, we are able to greatly accelerate KMC simulation of epitaxial growth while retaining the accuracy of physical results, including stochastic fluctuations in the region of primary physical interest.

In Section 2, we review the classic KMC algorithm for epitaxial growth, and analyze its computational complexity. In Section 3, we describe a variety of implementation schemes of our accelerated algorithms and their respective computational complexities. In Section 4, we verify our accelerated algorithms and complexity analysis by comparing the physical results and performances of accelerated algorithms with continuum models and classical KMC simulation.

2. The standard rejection free KMC method for simulating epitaxial growth

For simplicity, we choose the epitaxial growth on one-dimensional surface as our model system and assume there is no direct interaction between atoms. This allows us to compare our results to an analytic continuum model to verify that the average behavior is accurately simulated. By considering this simplified system we are also able to perform a detailed analysis of the computational complexity of the algorithm. This restricted model is meant to serve as a ''proof of concept'' of the algorithm. In the conclusions we will discuss ways in which this algorithm can potentially be extended to more realistic systems, i.e., two-dimensional surfaces and systems in which the effects of nucleation are not negligible.

In the system we consider there are four possible events:

1. Hopping of adatoms on the surface.
2. Attachment of adatoms to the edge from both the upper and lower terraces.
3. Detachment of atoms from the edge. (In our algorithm we assume an effectively infinitely asymmetric Ehrlich–Schwoebel barrier [17–19] and the ejected adatom always occupies the lower terrace. This is an arbitrary choice which is not essential to the model or the analysis.)
4. Flux of atoms to the surface from the molecular beam.

Since we assume no interaction between atoms, there are no nucleation events. Again for simplicity, we assume that attachment always follows when an adatom reaches the edge, and the event takes negligible time. The steps can occupy the same location, effectively step bunching, and they may move through each other with no interaction between steps other than the effective interaction via the adatoms. Periodic boundaries are implemented to allow continuous step flow. If we specify the rates of the other three kinds of events, then the standard BKL algorithm [20] for KMC simulation proceeds by:

1. Computing the total rate of the system from the rates of all the possible transitions:

$$R_{\text{tot}} = DN_{\text{a}} + R_{\text{d}}N_{\text{e}} + FN, \tag{1}$$

where D is the hopping rate; N_{a} is the number of adatoms on the surface; R_{d} is the detachment rate; N_{e} is the number of edges, F is the flux rate, and N is the total number of sites on the surface.

2. Advancing the time by $-\ln(r)/R_{tot}$, where the r is a random number in the range from zero to one.
3. Choosing an event with probability proportional to the rate and execute the event.
4. Repeat from step 1.

The speed of the KMC simulation depends on two factors: the total rate, which is proportional to the number of simulated events required for a certain period of time, and the average time required for the simulation of one event. For this system, it takes constant time to locate and execute the event since there are only three kinds of events with fixed rate to simulate each, and all events only induce strictly local changes.

We would like to know the total number of simulated events required to deposit one layer of atoms. Note that at equilibrium, the rate at which the adatoms are incorporated into the surface should equal the flux rate. So it takes $1/F$ time to deposit one layer. During this period, there will be $N_e R_d/F$ detachment events, $N_e R_d/F + N$ attachment events, and $N_a D/F$ diffusion events on average. We can approximate N_a by treating the edges as moving sinks and sources and solving for the density function in the moving reference frame traveling along with the terrace. The continuum diffusion equation is then formulated as follows:

$$\frac{\partial c}{\partial t} = \frac{D}{2}\frac{\partial^2 c}{\partial x^2} + FL\frac{\partial c}{\partial x} + F. \tag{2}$$

Here $L = N/N_e$ is the average terrace length between the edges. The first term on the right-hand side of Eq. (2) models the adatom diffusion. The second term models the convection due to the fact that we are in the reference frame of the terrace moving with velocity FL, and the third term models the flux of adatoms to the surface. The boundary conditions constrain the adatom flux at the edges:

$$f_x(0) = -D\left.\frac{\partial c}{\partial x}\right|_{x=0} = R_d - \left(\frac{D}{2} + FL\right)c(0), \tag{3}$$

$$f_x(L) = -D\left.\frac{\partial c}{\partial x}\right|_{x=L} = \left(\frac{D}{2} - FL\right)c(L), \tag{4}$$

where f_x denotes the flux of adatoms in the x direction. Solving these equations in steady state we can obtain an exact, albeit unwieldy, solution for the adatom density in the continuum limit. We find that by keeping only the dominant terms the average density derived from this exact solution can be approximated by

$$\bar{\rho} = \frac{N_a}{N} \approx \frac{R_d}{D} + \left(1 - \frac{2R_d}{D}\right)\left(\frac{1}{2}\coth\left(\frac{FL^2}{D}\right) - \left(\frac{2F}{D}L^2\right)^{-1}\right) \tag{5}$$

as long as we assume that $FL/D \ll 1$. Consequently, the total number of diffusion events required to deposit one layer is

$$\frac{DN_a}{F} = \frac{DN}{F}\bar{\rho} \approx \frac{R_d N}{F} + \frac{1}{2}\left(\frac{DN}{F} - \frac{2R_d N}{F}\right)\left[\coth\left(\frac{FL^2}{D}\right) - \frac{D}{FL^2}\right]. \tag{6}$$

When deposition is sufficiently slow that $FL^2/D \ll 1$ and the detachment rate is small compared to the diffusivity such that $R_d/D \ll 1$ this expression can be further simplified and it can be shown that the adatom density scales as

$$\frac{N_a}{N} = \bar{\rho} \approx \frac{R_d}{D} + \frac{FL^2}{6D} \tag{7}$$

and number of diffusion events scales as

$$O\left(\frac{R_d N}{F} + \frac{NL^2}{6}\right). \tag{8}$$

Fig. 1 shows the adatom density as a function of the terrace length as calculated by KMC and continuum diffusion. When the density reaches ~0.3 the assumption that $FL^2/D \ll 1$ no longer holds and the scaling in Eq. (7) breaks down. However the description given by Eq. (5) holds to higher adatom densities, and hence larger terrace lengths. In what follows we will refer to instances in which $FL^2/R_d \ll 1$ as the detachment dom-

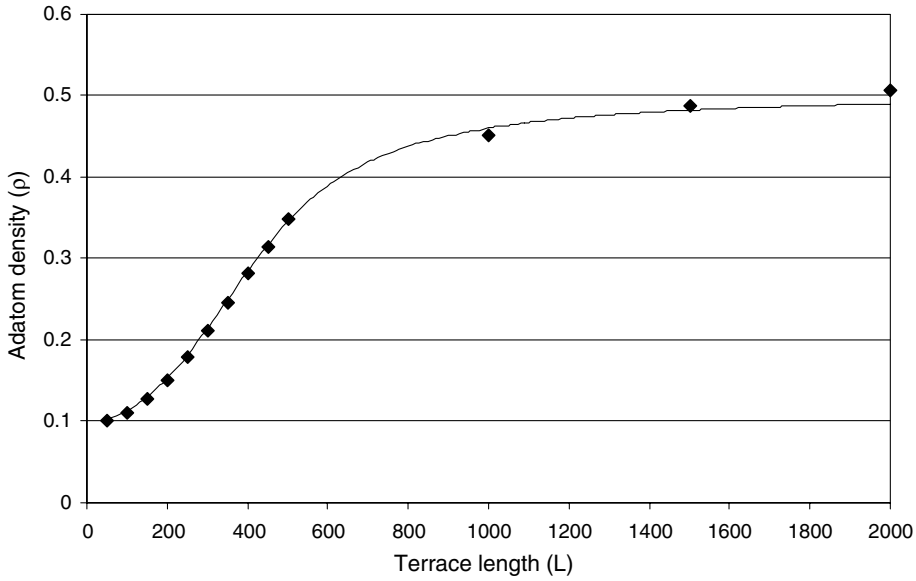

Fig. 1. Simple KMC simulation results of adatom density as a function of terrace length (L). The line shows the continuum approximation from Eq. (5) for comparison.

inated regime and the opposite limit as the deposition dominated regime. In both regimes, the computational cost of the standard rejection free algorithm is dominated by the diffusion of adatoms, and the standard rejection free simulation can be very slow even for systems with only a few hundred adatoms on the surface on average.

3. Analysis of the computational complexity of the accelerated algorithms

We develop our accelerated algorithm from the idea that the detailed hopping of adatoms far away from the edges are not important to the overall behavior of the system, especially in this model system where the adatoms are coupled to each other only through the movements of the edges. So we should be able to approximate the system with acceptable errors while reducing the total rate significantly by making adatoms far away from the edges jump less frequently but in larger steps. Since diffusion theory dictates that the root-mean-square displacement of the atom is proportional to the square root of the number of steps taken, we modify the system such that adatoms far away from the edges make jumps of size j instead of 1, with rate D/j^2 instead of D. To make sure that these coarse-grained diffusion events are physical, we must make sure that $j < d_{\min}$, the distance to the closest edge.

The simplest way to set up the step sizes is to choose jump size such that j is equal to the larger of $(d_{\min} - 1)$ or 1. We will refer to this method as "linear zoning" (LZ). In this case, almost all adatoms have different rates, so it takes $O(\log N_a)$ time to choose the adatom to diffuse if we arrange the rates of the adatoms into a binary tree and apply a standard divide-and-conquer algorithm. On the other hand, the total computational time for diffusion is greatly reduced. We can estimate the computational cost of the LZ method by integrating the density function derived from Eqs. (2)–(4) with the weight function:

$$w(x) = \begin{cases} d_{\min} > 1 & d_{\min}^{-2}, \\ d_{\min} \leqslant 1 & 1. \end{cases} \tag{9}$$

By doing so, we find that the total number of diffusion events required to deposit one layer is on order

$$O\left(\frac{R_d N_e}{F} + N \log L\right). \tag{10}$$

In the deposition dominated regime this implies that for a given terrace length, L, the LZ algorithm scales as $O(N \log N)$. However, when the total rate of the flux between the edges, FL, is significantly less than the detachment rate, R_d, the number of diffusion events will remain roughly constant independent of N, and the cost of diffusion is dominated by the cost of choosing adatoms and scales as $O(\log N)$.

These contributions must be weighed against the cost of attachment and detachment. For each attachment and detachment event the rates and step sizes of almost all atoms in the region of the moved edge must be modified. Thus attachment and detachment events take time on order $N_a/N_e \sim \bar{\rho}L$. The total number of attachment and detachment events per deposited layer is always unchanged

$$N + \frac{2N_e R_d}{F} \tag{11}$$

no matter what accelerated diffusion algorithm we use. Overall, when N is held fixed and L is varied, in the detachment dominated regime the cost of edge movements dominates the computational complexity of LZ, and multiplying Eq. (11) by N_a/N_e we find that it takes

$$O\left(N_a\left(L + \frac{2R_d}{F}\right)\right) \tag{12}$$

time to deposit one layer. Therefore in the detachment dominated regime the cost of the algorithm scales as $O(N_a)$ which grows linearly with the system size, N. In the deposition dominated regime the computation time scales with the system size multiplied by the terrace length.

The consequence of the computationally expensive attachment and detachment events of LZ algorithm is a continued linear scaling of the algorithm on system size or worse than linear scaling if the terrace length grows with system size. This motivates us to pursue a more efficient algorithm. The LZ algorithm requires at least linear time to move the edge because adatoms at different distances from the edge are all given different rates and step sizes. An obvious option is to divide the surface into larger rate zones so that whenever edge moves, only the rates and step sizes of the adatoms at the margins of the rate zones need to be modified. We choose to divide the surface into rate zones using a geometric series, namely 2^n, resulting in approximately $N_e \log L$ zones. Under the constraint $j < d_{min}$, we set $j = 1, 2, 6, 14, 30, \ldots = \max(1, 2^n - 2)$. We will refer to this method as "geometric zoning" (GZ). By integrating the density derived from Eqs. (2)–(4) with the rates of the zones as the weight function, we find that the total number of diffusion events required to deposit one layer scales identically to that calculated for the LZ algorithm in Eq. (10). The number of diffusion events still remains roughly constant with respect to N unless the system is in the deposition dominated regime.

In our implementation of the GZ algorithm we retain the binary rate tree of the LZ algorithm, so it still takes $O(\log N_a) \sim O(\log N)$ time to choose the adatom to diffuse. Therefore the time for diffusion in this scheme is $O((N_e R_d/F + N \log L) \log N)$. Using this scheme we only have to update the rates and the step sizes of the adatoms at the margins of the rate zones for attachment and detachment events. Algorithmically, two operations are required for these updates:

1. Check the margins of the rate zones in the region of the moved edge. If there are adatoms present, then
2. Update the rates and the step sizes of these adatoms.

The first clearly takes $O(\log L)$ time. We can estimate the total number of adatoms on the margins of the rate zones in the region of the moved edge and therefore the computational complexity of the second operation by taking the finite summation $2\sum \rho(x_n)$ over $x_n < L/2$ where ρ is the density function derived from Eqs. (2)–(4) and $x_n = 2(2^n - 1)$. Since the adatom density is bounded by approximately 0.5 as terrance length increases in Eq. (5) (see Fig. 1), we can approximate ρ by a constant, $\bar{\rho}$. This gives us the expression $\bar{\rho} \log L$. Since we have to update the binary tree from the leaf (adatom) all the way to the root with this implementation, it takes $O(\log N_a)$ time to update an adatom. Therefore, the second operation takes $O(\bar{\rho} \log N_a \log L)$ time. Since the total number of attachment and detachment events is $N_e R_d/F + N$, the total cost of all edge movements scales as $O((N_e R_d/F + N) \log N \log L)$. This is the dominant contribution to the computational complexity. Notice that the term that is superlinear in N will not show up unless the system is in the deposition dominated regime. Therefore the GZ algorithm is anticipated to always have superior scaling with respect to the LZ algo-

Table 1
Computational complexities of algorithms described in text

	Standard rejection free KMC	Linear zoning (LZ)	Geometric zoning (GZ)
Diffusion events	$O(N_a D/F)$	$O(N_e R_d/F + N \log L)$	$O(N_e R_d/F + N \log L)$
Cost per diffusion event	$O(1)$	$O(\log N)$	$O(\log N)$
Attachment/detachment events	$O(2N_e R_d/F + N)$	$O(2N_e R_d/F + N)$	$O(2N_e R_d/F + N)$
Cost per attachment/detachment	$O(1)$	$O(N_a/N_e)$	$O((N_a/N)\log N_a \log L)$
Computational complexity	$O(N_a D/F)$	$O(N_a(L + R_d/F))$	$O((N + N_e R_d/F)\log N \log L)$

rithm in the detachment dominated regime, and will have better scaling in the deposition dominated regime when the terrace length scales with system size. It has somewhat worse scaling in the deposition limited regime when the terrace length is independent of system size.

To test out how the accelerated algorithm using geometric zoning (GZ) affects the physical result, we also ran this implementation with a modification which retains the sizes of the rate zones of the original but modifies the step sizes to be the size of the previous rate zone. In other words, the sizes of the rate zones are still described by a geometric series 2^n, but the step sizes correspond to the series 2^{n-1}. This is chosen in order to force every newly added adatom to pass through each rate zone before it attaches to the edge. We believe the added detail results in a more accurate description of diffusion to the step edge. Again by integrating the density with the rates of the zones as the weight function, we find that the total number diffusion events scales as in Eq. (10). Therefore, the computational complexity of this variant, which we will call GZ2 is the same as that of the original, which we will refer to as GZ1. Table 1 summarizes the computational complexities of different diffusion algorithms discussed in this section.

4. Verification of acceleration and scaling of the algorithms

In order to justify the approximations and the computational complexity analysis we tested our algorithms against the standard rejection free KMC simulation and compared the performances and equilibrium statistics. We used the parameters $N = 1000$, $R_d/D = 0.1$, $N_e = 2$ (two edges placed at equal distances to each other), while we varied D/F. The simulation runs until a total of 500 layers are deposited. In order to make sure that the statistics accurately reflect the equilibrium state instead of the transient phase, the statistics were

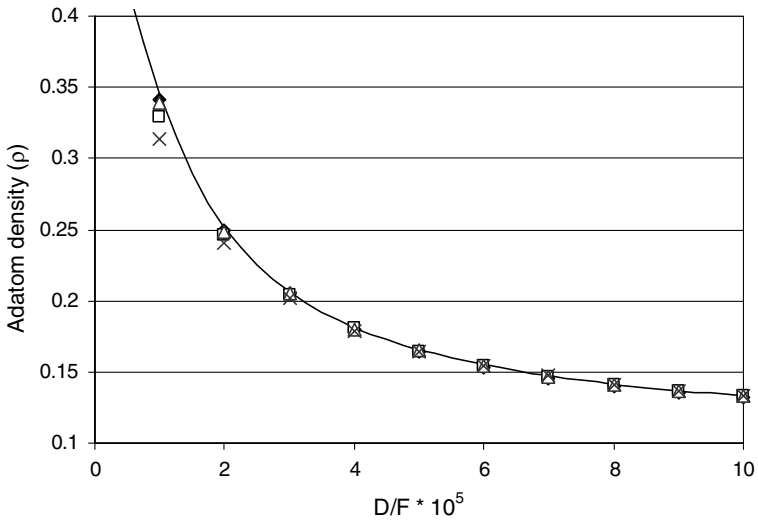

Fig. 2. Average adatom density vs. the ratio of the hopping rate to the deposition rate (D/F). Results are given for the standard rejection free KMC results (◆), LZ (×), GZ1 (□) and GZ2 (△) algorithms. For these simulations $N = 1000$, $R_d/D = 0.1$, $N_e = 2$ and 500 layers were deposited. The results are compared to the continuum approximation given in Eq. (5).

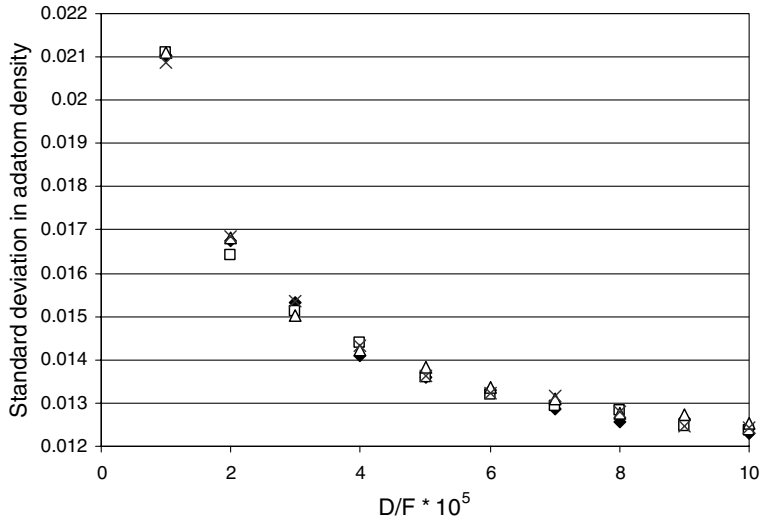

Fig. 3. Standard deviation in adatom density vs. the ratio of the hopping rate to the deposition rate (D/F). Results are given for the standard rejection free KMC results (◆), LZ (×), GZ1 (□) and GZ2 (△) algorithms. All parameters are identical to those for Fig. 2.

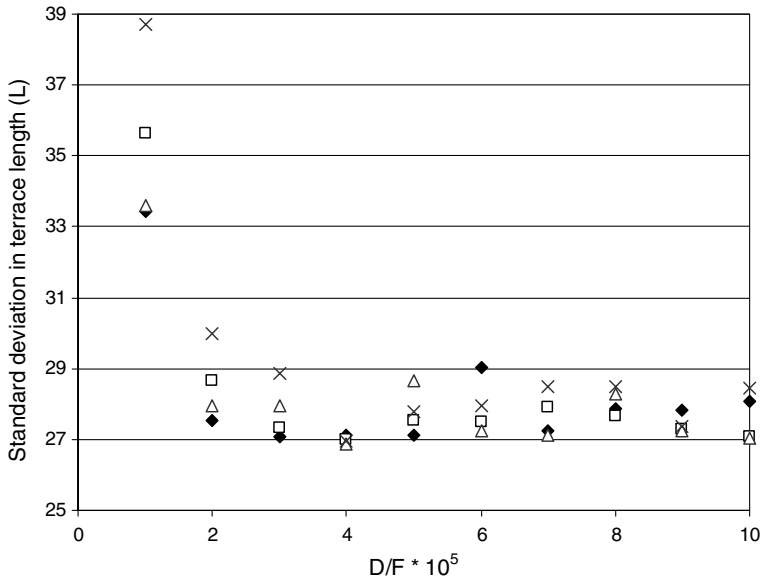

Fig. 4. Standard deviation in terrace length (L) vs. the ratio of the hopping rate to the flux rate (D/F). Results are given for the standard rejection free KMC results (◆), LZ (×), GZ1 (□) and GZ2 (△) algorithms. All parameters are identical to those for Fig. 2.

taken only after the first layer had deposited. Figs. 2–4 represent the physical statistics of the simulation while Figs. 5–7 represent the performance of the algorithms.

We can see from Figs. 2–4 that our accelerated algorithms accurately reproduced the physical statistics of the standard rejection free KMC algorithm, especially in the region of high D/F ratio. As expected, GZ2 is more accurate than GZ1, which is in turn more accurate than LZ, since LZ has the largest step sizes on average while GZ2 has the smallest step sizes among the accelerated algorithms. Nevertheless, even LZ deviates no more than 10% in average adatom density in Fig. 2 and no more than 16% in the standard deviation of the terrace length in Fig. 4. The accelerated algorithms exhibit a tendency to underestimate adatom density while overestimating the standard deviation of the terrace length when D/F is small. The continuum approximation

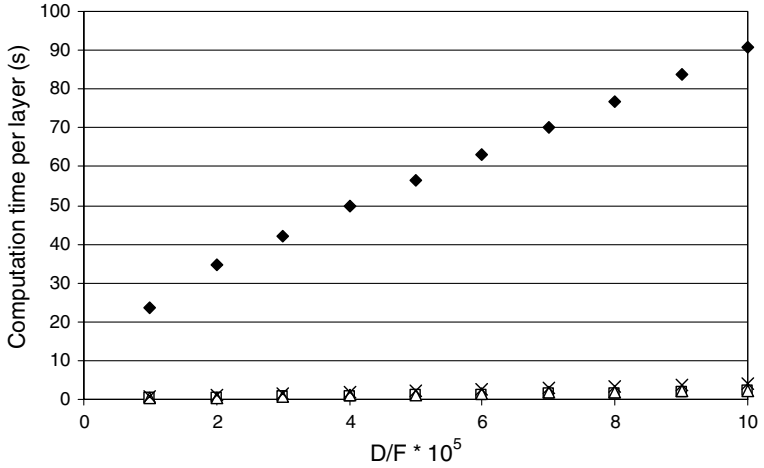

Fig. 5. Time scaling of all algorithms with respect to the ratio of the hopping rate to the deposition rate (D/F). Results are given for the standard rejection free KMC results (◆), LZ (×), GZ1 (□) and GZ2 (△) algorithms. All parameters are identical to those for Fig. 2.

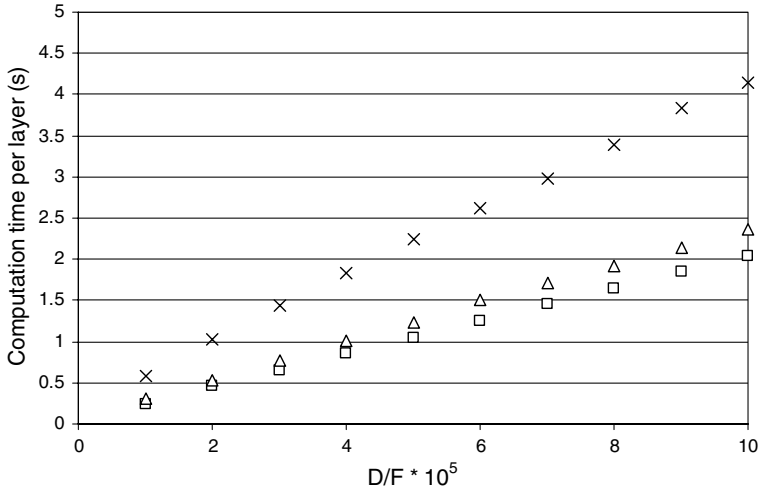

Fig. 6. Time scaling of the accelerated algorithms with respect to the ratio of the hopping rate to the deposition rate (D/F). Results are given for the LZ (×), GZ1 (□) and GZ2 (△) algorithms. All parameters are identical to those for Fig. 2.

obtained in Eq. (5) from solving diffusion equation with moving sinks and sources at the boundary is also shown in Figs. 1 and 2 for comparison with the simulated data.

As we can see from Figs. 5 and 6, the simulation time required grows linearly with D/F, the de-dimensionalized diffusion rate. This is expected since we fixed the value of R_d/F to be $0.1D/F$, and the computational complexities of algorithms all roughly depend linearly on R_d/F. A 40- to 20-fold acceleration is achieved with LZ, while the acceleration of GZ1 reaches 100- to 40-fold. From Fig. 7, we can see that the ratio of acceleration with respect to the standard rejection free KMC simulations for all three accelerated algorithms is proportional to N_a.

In addition to simulation time, we also recorded the total number of events simulated in order to compare the average time required to simulate an event for each algorithm. We found that the simulation speed in terms of events/s remains quite constant for standard rejection free KMC algorithm, GZ2, and GZ1, at approximately 1.45, 0.65, and 0.59 million events/s on our test platforms, respectively. As expected, although the standard rejection free KMC algorithm is the fastest in terms of events/s, it is nevertheless the slowest in terms of simulation time due to the staggering total number of simulated events required. GZ2 runs slightly

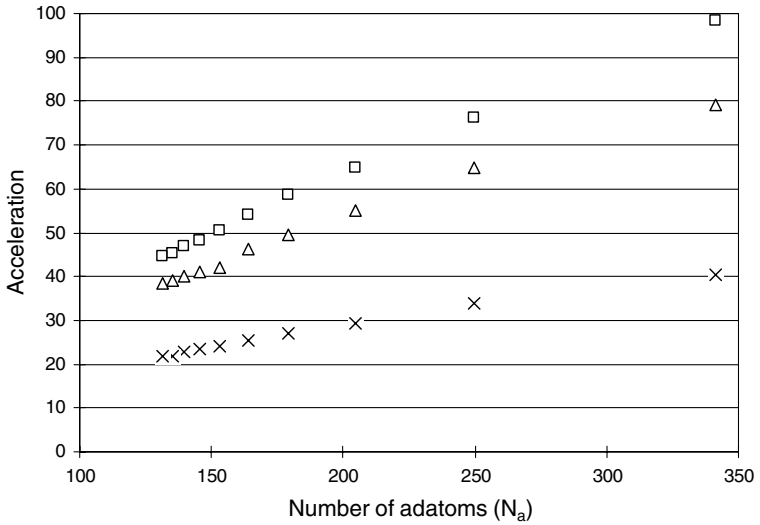

Fig. 7. Acceleration ratios of the algorithms as compared to the standard rejection free KMC algorithm as a function of the average number of adatoms in the system. Results are given for the LZ (×), GZ1 (□) and GZ2 (△) algorithms. All parameters are identical to those for Fig. 2.

faster than GZ1 in terms of events/second, due to the increased number of computationally cheaper diffusion events. LZ is the slowest in terms of events/s, and the actual speed ranges from 0.17 to 0.23 million events/s as D/F goes from 10^5 to 10^6. This is not surprising since the computational cost of LZ is dominated by attachment and detachment events, which happen proportionally less frequently when D/F is higher.

From Fig. 8 we can see the scaling of these algorithms with respect to size of the systems. We can make direct comparisons to our predictions for the scaling as given in Table 1. As expected, the standard rejection free algorithm scales with $N_a \sim \bar{\rho}N$, and the LZ algorithm scales with N_a $(N + 2N_e R_d/F)$. Hence both scale linearly or worse with respect to size due to increasing number of surface atoms. GZ1 and GZ2 scale as

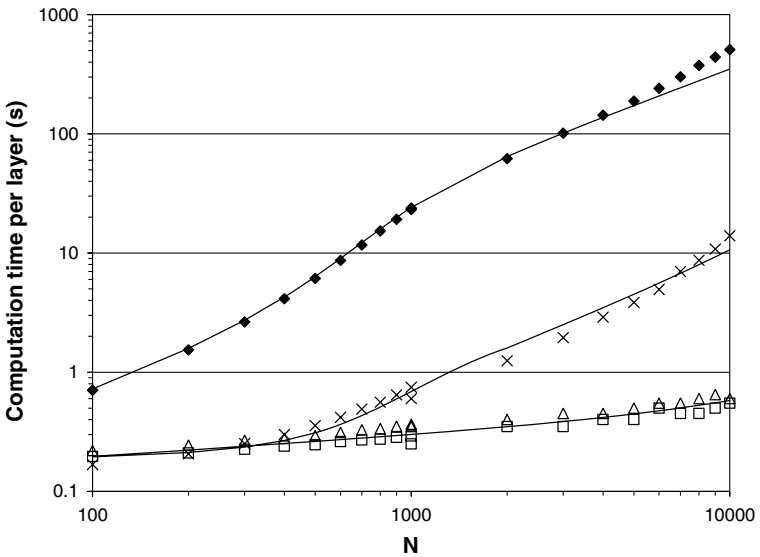

Fig. 8. Log–log graph of the time scaling with respect to N. In each simulation the number of edges was kept constant; $N_e = 2$. For $N \leqslant 1000$, 500 layers were deposited, while for $N > 1000$, 20 layers were deposited. Results are given for the LZ (×), GZ1 (□) and GZ2 (△) algorithms as well as the standard rejection free algorithm (♦). The lines provide comparisons to the predictions given in the text and Table 1.

$(N + 2N_e R_d/F)\log N \log L$, essentially logarithmically with size, as expected in the detachment dominated regime. By comparing these predictions we can obtain an estimate for when the accelerated algorithms will be useful. At system sizes smaller than approximately 30 lattice spacings, i.e., terrace lengths smaller than 15 lattice spacings, the acceleration schemes are no longer advantageous.

5. Conclusions

By balancing the cost of diffusion events and edge movements with careful assignment of step sizes and rates according to the relative distance to the closest step edge, we have achieved an accelerated KMC algorithm (GZ1 and GZ2) for epitaxial growth orders of magnitudes faster than the classical KMC algorithm and superior in terms of computational complexity. We have compared the algorithmic scaling to predictions based on analysis of the various algorithms., We have also compared the accuracy of the various algorithms. GZ2 is more accurate than GZ1, which is in turn more accurate than LZ; the accelerated algorithms tend to underestimate the adatom density by a small amount. Compared to similar algorithms that employ a coupling of continuum and atomistic methodologies, as in [8], the method appears to provide more dramatic speedup and increased accuracy when compared to the standard rejection free KMC algorithm. Since the acceleration is achieved by modifying classical KMC algorithm rather than coupling between two distinct simulation algorithms, it is easier to implement and does not require multiple representations for the adatoms.

The algorithms described have only been implemented under the simplified assumptions described above. In order to be more generally applicable implementations need to be developed to simulate two-dimensional surfaces and to include nucleation processes. Inclusion of two-dimensional surfaces seems straightforward since domain decomposition techniques have been deployed in the context of a number of other simulation methods including the parallelization of molecular dynamics [21]. While the details would be different, conceptually the problem would be similar. Introducing nucleation in such a model would be more complex, but could be introduced effectively by introducing interaction events that represent the probability of an encounter between two atoms much as in Ref. [12]. This probability would have to depend on their proximity to each other and to other potential sinks. While the integrals needed to compute these probabilities could be complex, they could certainly be numerically approximated. In any case, the methods are specific to the regime of step-flow growth in which terrace lengths are long (15 or more lattice spacings by our estimate). The methods would not provide significant acceleration to layer-by-layer growth dominated by nucleation events.

Acknowledgments

The authors thank Peter Smereka and Tim Schulze for helpful conversations and clarifications and the NSF for support under Grant # CMS 0331016.

References

[1] F.F. Abraham, G.W. White, Computer simulation of vapor deposition on two-dimensional lattices, Journal of Applied Physics 41 (1970) 1841.
[2] G.H. Gilmer, P. Bennema, Simulation of crystal growth with surface diffusion, Journal of Applied Physics 43 (1972) 1347.
[3] J.D. Weeks, G.H. Gilmer, Dynamics of crystal Growth, Advances in Chemical Physics 40 (1979) 157–228.
[4] S.V. Ghaisas, A. Madhukar, Monte-Carlo simulations of MBE growth of III–V semiconductors – the growth-kinetics, mechanism, and consequences for the dynamics of RHEED intensity, Journal of Vacuum Science and Technology B 3 (1985) 540–546.
[5] A. Madhukar, S.V. Ghaisas, The nature of molecular-beam epitaxial-growth examined via computer-simulations, CRC Critical Reviews in Solid State and Materials Sciences 14 (1988) 1–130.
[6] H. Metiu, Y.T. Lu, Z.Y. Zhang, Epitaxial-growth and the art of computer-simulations, Science 255 (1992) 1088–1092.
[7] W.K. Burton, N. Cabrera, F.C. Frank, The growth of crystals and the equilibrium structure of their surfaces, Philos. Trans. Roy. Soc. Lond. 243A (1951) 299.
[8] T.P. Schulze, P. Smereka, E. Weinan, Coupling kinetic Monte-Carlo and continuum models with application to epitaxial growth, Journal of Computational Physics 189 (2003) 197–211.
[9] S.P.A. Gill, P.E. Spencer, A.C.F. Cocks, A hybrid continuum/kinetic Monte Carlo model for surface diffusion, Materials Science and Engineering A – Structural Materials Properties Microstructure and Processing 365 (2004) 66–72.
[10] T.P. Schulze, A hybrid scheme for simulating epitaxial growth, Journal of Crystal Growth 263 (2004) 605–615.

[11] G. Russo, L.M. Sander, P. Smereka, Quasicontinuum Monte Carlo: a method for surface growth simulations, Physical Review B 69 (2004) 121406.
[12] L. Mandreoli, J. Neugebauer, R. Kunert, E. Scholl, Adatom density kinetic Monte Carlo: a hybrid approach to perform epitaxial growth simulations, Physical Review B 68 (2003) 155429.
[13] M. Petersen, C. Ratsch, R.E. Caflisch, A. Zangwill, Level set approach to reversible epitaxial growth, Physical Review E 6406 (2001).
[14] R.E. Caflisch, E. Weinan, M.F. Gyure, B. Merriman, C. Ratsch, Kinetic model for a step edge in epitaxial growth, Physical Review E 59 (1999) 6879–6887.
[15] M. Plapp, A. Karma, Multiscale finite-difference-diffusion Monte-Carlo method for simulating dendritic solidification, Journal of Computational Physics 165 (2000) 592–619.
[16] J.P. DeVita, L.M. Sander, P. Smereka, Multiscale kinetic Monte Carlo algorithm for simulating epitaxial growth, Physical Review B 72 (2005) 205421.
[17] R.L. Schwoebel, E.J. Shipsey, Step motion on crystal surfaces, Journal of Applied Physics 37 (1966) 3682.
[18] K. Kyuno, G. Ehrlich, Step-edge barriers: truths and kinetic consequences (383 (1986) 766), Surface Science 394 (1997) L179–L187.
[19] R.L. Schwoebel, Step motion on crystal surfaces 2, Journal of Applied Physics 40 (1969) 614.
[20] A.B. Bortz, M.H. Kalos, J.L. Lebowitz, A new algorithm for Monte Carlo simulation of Ising spin systems, Journal of Computational Physics 17 (1975) 10–18.
[21] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, Journal of Computational Physics 117 (1995) 1–19.