Reaching extended length scales and time scales in atomistic simulations via spatially parallel temperature-accelerated dynamics

Yunsic Shim* and Jacques G. Amar†
Department of Physics & Astronomy, University of Toledo, Toledo, Ohio 43606, USA

B. P. Uberuaga‡ and A. F. Voter§
Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA
(Received 26 June 2007; published 29 November 2007)

We present a method for performing parallel temperature-accelerated dynamics (TAD) simulations over extended length scales. In our method, a two-dimensional spatial decomposition is used along with the recently proposed semirigorous synchronous sublattice algorithm of Shim and Amar [Phys. Rev. B **71**, 125432 (2005)]. The scaling behavior of the simulation time as a function of system size is studied and compared with serial TAD in simulations of the early stages of Cu/Cu(100) growth as well as for a simple case of surface relaxation. In contrast to the corresponding serial TAD simulations, for which the simulation time t_{ser} increases as a power of the system size N ($t_{ser} \sim N^x$) with an exponent x that can be as large as three, in our parallel simulations the simulation time increases only logarithmically with system size. As a result, even for relatively small system sizes our parallel TAD simulations are significantly faster than the corresponding serial TAD simulations. The significantly improved scaling behavior of our parallel TAD simulations over the corresponding serial simulations indicates that our parallel TAD method may be useful in performing simulations over significantly larger length scales than serial TAD, while preserving all the atomistic details provided by the TAD method.

DOI: 10.1103/PhysRevB.76.205439 PACS number(s): 81.10.Aj, 82.20.Db, 66.30.Fq, 82.20.Wt

I. INTRODUCTION

One of the challenges in carrying out realistic nonequilibrium simulations of materials is that many important processes occur on a time scale which is much longer than the typical atomic vibrational time scale. For example, because of the small time step required to integrate the equations of motion, the standard method for dynamical simulations of materials at the atomistic level, molecular dynamics (MD), is generally limited to nanoseconds. However, the relevant thermally activated processes typically take place over time scales of microseconds or even longer.

One approach which has been widely used to overcome the time-scale limitations of molecular dynamics is the kinetic Monte Carlo (KMC) method. While this is an extremely efficient method for the simulation of processes involving activated events, in order to carry out a realistic KMC simulation, one needs to know in advance all the relevant atomic processes along with their corresponding rates. However, precalculating and storing the transition rates for all possible processes is in general extremely difficult. In addition, since many important processes may involve complicated concerted motions, they may be impossible to "guess" *a priori*, and thus will not be included in the KMC rate table. As a result, in KMC simulations only a relatively limited catalog of atomic-scale processes is typically included (along with their corresponding rates), thus limiting the accuracy of the simulations.

In order to overcome the time-scale limitations of MD while preserving the accuracy of the simulation, a variety of accelerated dynamics algorithms have been proposed, including hyperdynamics[1,2] and its variants,[3,4] parallel replica dynamics,[5] and temperature-accelerated dynamics[6,7] (for a review see Ref. 8). In addition, a variety of other methods such as adaptive kinetic Monte Carlo[9,10] and self-learning kinetic Monte Carlo[11,12] have also been proposed. One particularly successful approach is the recently proposed temperature-accelerated molecular dynamics (TAD) method[6,7] which relies on the assumption of harmonic transition-state theory.[13–17] The basic idea behind TAD is that instead of directly carrying out a MD simulation at the desired temperature (T_{low}), a basin-constrained simulation at a significantly higher temperature (T_{high}) is performed. The information obtained from the high-temperature simulation is then used, along with the assumption of harmonic transition state theory, to determine the evolution at the desired low-temperature. The TAD method has recently been applied to the study of a variety of nonequilibrium processes including simulations of the early stages of low-temperature homoepitaxial[18] and heteroepitaxial[19] growth at experimental deposition rates. It has also been used to study the mobility of surface atoms on the flat Ag(100) surface at room temperature and above,[20] as well as to the study of defects and defect diffusion in bulk MgO (Ref. 21) and the dissolution of boron interstitial clusters in Si.[22]

While the TAD method has proven to be very successful in extending the time scale of nonequilibrium simulations, the size of the system that a TAD simulation can reasonably handle is quite small, of the order of a few hundred to a thousand atoms, due to the fact that the time for the standard serial TAD simulation typically increases rapidly with system size. In particular (see the end of Sec. II A) we expect this time to scale approximately as $N^{3+1/3-\gamma}$ where $\gamma = T_{low}/T_{high}$ and N is the number of atoms in the system. However, in many cases of interest, such as multilayer epitaxial growth, the relevant feature sizes involve significantly larger length scales and numbers of atoms, and so simulations of larger systems are clearly required.

 ©2007 The American Physical Society

In this paper we present a method for extending TAD simulations to longer length scales, based on the use of spatial parallelization. In our parallel TAD simulations, we use the recently proposed synchronous sublattice (SL) algorithm[23] which has been shown to be quite efficient in KMC simulations of Cu growth[24] and of the coarsening of Ag islands on the Ag (111) surface.[25] The organization of this paper is as follows. In Sec. II, we briefly review the TAD method and discuss how we have incorporated the SL algorithm in order to develop a parallel version of TAD. In Sec. III, we present results for the scaling of parallel TAD as a function of system size along with a comparison with serial TAD. We also discuss several factors affecting the performance of a parallel TAD simulation, including the choice of an optimal high temperature as well as the existence of low-energy repetitive events. The effects of the sublattice structure on strain relaxation are also discussed. Finally, in Sec. IV we summarize our results.

II. SIMULATIONS

A. Review of TAD method

While the TAD method has been discussed and applied in a number of publications, in order to fix our notation, we first briefly review the method. As already noted, the TAD method is based on the harmonic approximation to transition state theory, which states that the rate ν_i of an activated event i with energy barrier E_i may be described using the Arrhenius law,

$$\nu_i = \nu_{0,i} \exp[-E_i/k_B T], \tag{1}$$

where $\nu_{0,i}$ is the frequency prefactor, k_B is Boltzmann's constant, and T is the temperature. A TAD simulation then involves performing a basin-constrained MD simulation at a temperature T_{high} which is higher than the desired temperature T_{low} at which one wishes to study the time evolution of the system. At some point in time $t_{i,high}$ during the high-temperature MD, the system escapes from an initial basin state A and moves to another basin state B. The activation energy E_i for this transition is then typically determined using the nudged elastic band (NEB) algorithm,[26,27] before returning the system back to the original basin and continuing the high-temperature MD simulation. The activation energy and high-temperature time $t_{i,high}$ are then used to determine the time $t_{i,low}$ this transition would have occurred in an equivalent realization of the dynamics at low temperature,

$$t_{i,low} = t_{i,high} e^{E_i(\beta_{low}-\beta_{high})}, \tag{2}$$

where $\beta = 1/k_B T$.

This process continues until one is confident, with confidence level $1-\delta$ (where δ is a user-supplied parameter), that no further transitions will be found with extrapolated low-temperature event times shorter than the shortest extrapolated low-temperature event time found so far ($t_{low,short}$). In particular, if one assumes that there is a lower bound ($\nu_{\min}$) on the frequency prefactors $\{\nu_{0,i}\}$ for all possible transitions, then the high temperature MD run can be terminated once the high-temperature simulation time $t_{i,high}$ reaches the value

$$t_{high,stop} = \frac{\ln(1/\delta)}{\nu_{\min}} \left(\frac{\nu_{\min} t_{low,short}}{\ln(1/\delta)} \right)^{T_{low}/T_{high}}. \tag{3}$$

The event at time $t_{low,short}$ is then accepted and the whole process is repeated starting in the new basin state. We note that a variation of the TAD method[7] has been developed which permits an additional time acceleration if one knows the minimum energy barrier E_{min} to leave a given state. In this case, the expression for the high-temperature MD time required before accepting an event may be written,[7]

$$t_{high,stop} = t_{low,short} e^{-E_{min}(\beta_{low}-\beta_{high})}. \tag{4}$$

We now briefly outline the expected scaling behavior of the serial TAD algorithm as a function of system size N. As discussed in Ref. 8, the high-temperature MD computational time scales as $N^{2-\gamma}$ where $\gamma = T_{low}/T_{high}$. Using Eq. (3) one can show[7] that for a system of N atoms, the number of high-temperature (attempted) activated events per unit of low-temperature simulated time also scales as $N^{2-\gamma}$. If we assume that asymptotically, the simulation time is dominated by the cost of saddle-point searches which must be made for each attempted event, then the total serial TAD time scales approximately as $N^{2-\gamma}$ times the time t_{srch} to determine a saddle point. In the standard serial TAD algorithm, such a search is carried out globally and as a result, t_{srch} increases approximately as $N^{4/3}$, since the work per iteration scales as the system size N, while the number of iterations also increases due to the presence of long-wavelength modes.[28] Thus in the absence of localized saddle-point searches, the total time for a standard serial TAD simulation scales approximately as $N^{3+1/3-\gamma}$. While in principle this scaling can be reduced to $N^{2-\gamma}$ if saddle searches are localized to the region of the transition, we have not implemented this approach for the simulations in the present study. We also note that such a localization must be applied carefully, as processes involving defects with long-range strain fields require a much greater localization length to achieve the same accuracy in the activation energy E_i.

B. Synthetic mode

In the case of highly repetitive low-barrier events, our TAD simulations were executed in "synthetic" mode, so that the acceptance time, or required high-temperature simulation time to accept an event, becomes shorter and shorter for each revisit to the state. This synthetic mode, which we have implemented somewhat differently from the original TAD simulations,[6] works as follows. Assume we have a state that is revisited a number of times, and a low-barrier, often-accepted escape pathway i out of that state. Once a certain number of attempted escapes n_{att} have occurred along pathway i, we can use this information along with the barrier height E_i to calculate the total escape rate and prefactor for this pathway, as well as the rate constant $k_{i,low}$ at the low temperature. We then generate "synthetic" low-temperature escape times by drawing times from the appropriate exponential distribution, $p(t) = k_{i,low} \exp(-k_{i,low} t)$. For this pathway i, these times are henceforth used instead of times extrapolated from high-T escape attempts [see Eq. (2)]. We

note that in nonsynthetic "standard" TAD, when a state is revisited, the high-temperature time is reset to zero and a fresh set of time extrapolations are generated, from which one determines the first escape pathway and the stop time $t_{high,stop}$ [see Eqs. (2) and (3)]. However, in synthetic mode, upon revisiting the state, it is safe to continue advancing the high-temperature time until reaching the stop time for the next synthetic event at low temperature, still using the stop time definition in Eq. (3). Multiple pathways out of the state can be converted to synthetic mode if desired. As a result, it is safe to continue advancing the time in the state upon each revisit, provided that every event that has been accepted since the last reset was a synthetic event. In this way, the low-temperature event times increase with each new visit, and hence the stop time from Eq. (3) corresponds to an increasing boost factor $t_{low,short}/t_{stop,high}$. In the present simulations, we have used $n_{att}=10$.

While in the present implementation, the high temperature time is advanced on each revisit only enough to reach the stop time for the next accepted event on the low-temperature time line, in the original implementation,[6] once a pathway was promoted to synthetic mode, the high-temperature time was advanced until the next accepted *nonsynthetic* event time at low-temperature was reached, typically passing many synthetic event times along the way. Successive revisits to the state then required no new high-temperature MD until after the nonsynthetic event was accepted. The advantage of the present approach is that time is saved if the state is revisited fewer times than required to reach the next nonsynthetic event.

Finally, we note that for revisits that occurred before the high temperature rate constant had been determined for the fast escape pathway(s) (i.e., before n_{att} attempts had been observed), we nonetheless obtained some increase in the boost factor by utilizing the method explained in detail in Ref. 7. In this method, the stop time is replaced by a shorter stop time, which arises from knowing a lower-bound on the minimum barrier for escape from the state due to time previously spent in the state. As the state is revisited, this lower bound increases and the stop time decreases.

C. Sublattice algorithm for parallel TAD

In order to carry out parallel TAD simulations, we have used a quasi-two-dimensional (2D) domain decomposition (see Fig. 1) along with the recently developed semirigorous synchronous sublattice (SL) algorithm.[23] This algorithm has been shown to be relatively efficient in parallel KMC simulations of a variety of models of multilayer epitaxial growth[23,24] and island coarsening.[25,29] In addition, due to the fact that it requires only local communication with nearest neighbors it has been shown to scale linearly, i.e., the parallel efficiency is independent of the number of processors for a large number of processors. While the SL algorithm is not exact, in parallel KMC simulations of a variety of models[23–25,29] it was found that unless the processor size is extremely small (smaller than a "diffusion length") or the cycle time is too large, there is essentially perfect agreement between the results of parallel simulations using the SL algorithm and serial simulations.

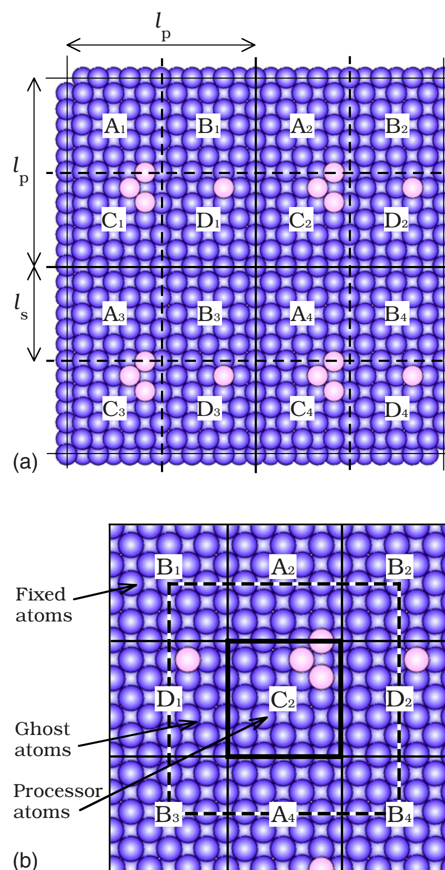

FIG. 1. (Color online) (a) Sublattice decomposition for $N_p=4$ where A, B, C, or D denotes a sublattice and the subscript represents processor number. (b) Simulation region for sublattice C_2, showing sublattice (central region), ghost (surrounded by dashed line), and surrounding fixed atoms.

In our application of the SL algorithm to parallel TAD, a square decomposition is used. In particular, each square region is assigned to a different processor, and in order to avoid conflicts between processors due to the synchronous nature of the algorithm, each processor's domain is further divided into four square sublattices (A,B,C, or D). Figure 1(a) shows a schematic diagram of square decomposition along with the further division into four sublattices for the case of four processors ($N_p=4$) corresponding to a system size (in units of the processor size) $N_S=4$. The lateral sizes of the processor and sublattice domains are $l_p=6a$ and $l_s=3a$, respectively, where $a=3.615$ Å is the lattice constant of Cu. We note that this gives a sublattice size l_s which is about 2.2 times larger than the range of the potential cutoff ($r_{cut}=4.961$ Å).

In order for each processor to carry out TAD independently for a given sublattice, the configuration in neighboring processors must be known at least as far as the range of the potential cutoff. As a result, in addition to containing the configuration information for its own domain, each processor's array also contains a "ghost-region" of moving atoms as well as a more distant "fixed-region" of fixed atoms which include the relevant information about the neighboring processor's configuration beyond the processor's boundary. Fig-

ure 1(b) shows the detailed structure of the simulation region for sublattice C as an example. As can be seen, the sublattice region is surrounded by the ghost region and further by a fixed region in which the ghost region is "embedded." In our simulations, the width of the ghost and fixed regions was set to $d_{\text{ghost}}=d_{\text{fixed}}=l_s/2$, as shown in Fig. 1(b). As a result, the total area of all atoms in the TAD simulation for a given selected sublattice and processor including processor, ghost, and fixed atoms is equal to $9l_s^2$. We note that while the finite-size of the sublattice clearly limits the maximum size of concerted events which may be included in our parallel TAD simulations, we expect that it is large enough to include the relevant events during low-temperature Cu(100) growth. In particular, it is large enough to include concerted events such as the "collapse" of a (111) facet recently observed by Voter *et al.*[8] [see Fig. 9(d) in Ref. 8]. In addition, if the ghost region is included, then the sublattice size is approximately $6a\times 6a$ which is somewhat larger than the largest sizes previously used in serial TAD simulations of thin-film growth.

We note that for the EAM potential used in the present study, while the energy of an atom is affected by neighboring atoms only as far away as the cutoff distance r_{cut}, the *force* on an atom can be affected by atoms as far away as $2r_{cut}$. This is a consequence of the many-body nature of the potential: the force on atom i depends in part on the electron density $\bar{\rho}_j$ at each atom j within its cutoff range, while $\bar{\rho}_j$ depends in turn on all atoms within one cutoff distance of atom j. Thus the use of a fixed region of width $0.5l_s$, or $1.1r_{cut}$, introduces errors in the forces on some of the atoms in the ghost region. Nevertheless, we believe that these boundary conditions are acceptable for the present study since the force errors are reasonably small (on the order of 5×10^{-5} eV/Å) and only affect the ghost region, and also because our main focus is on the computational scaling. In this connection, we note that the missing fixed atoms can be included without significantly increasing the computational cost, since their contributions to the electron densities can be computed once at the beginning of a subcycle (see below) and then reused during the remainder of the subcycle. However, in some cases (e.g., in the absence of deposition) there may be a small change in the overall scaling since accepted transitions in a given processor can affect the fixed-atom positions of a greater number of neighboring sublattices in the next subcycle.

A complete synchronous cycle corresponding to a time interval τ then proceeds as follows. Since there are four sublattices, each cycle is divided into four subcycles. In each subcycle one of the sublattices (A, B, C, or D) is selected (either randomly or sequentially) and is the same for all processors. Each processor then independently carries out TAD in the selected sublattice until the low-temperature time t_{low} has increased by an amount equal to τ. During the TAD simulation, atoms in the ghost region are also allowed to move, and may even participate in an attempted transition. However, any attempted transition that is centered in the ghost region is rejected because such an attempted event belongs to another sublattice. In order to determine in general whether or not an attempted transition belongs to a given processor's selected sublattice region, and thus avoid "double-counting," after every attempted transition the center of displacement of the move was calculated using the expression $\mathbf{r}_{CD}=\Sigma_j\delta_j\mathbf{r}_j^i/\Sigma_j\delta_j$, where $\delta_j=|\mathbf{r}_j^f-\mathbf{r}_j^i|$ is the magnitude of the displacement of atom j and $\mathbf{r}_j^i$ and $\mathbf{r}_j^f$ are the initial and final positions of atom j, and the sum is carried out over atoms for which δ_j is greater than a threshold value (here this threshold was set to 0.1 Å). If $\mathbf{r}_{CD}$ resides within the boundaries of the currently active sublattice, the attempted transition is recorded as belonging to this processor; otherwise it is rejected.

At the end of each subcycle, each processor communicates any necessary changes (boundary events) with its neighboring processors. Due to the presence of extended ghost and fixed regions, each processor may typically need to communicate with 1–3 neighboring processors at the end of each subcycle, and with up to six nearest neighbors for large events which "straddle" all four boundaries. In our previous applications of the SL algorithm to kinetic Monte Carlo simulations[23–25,29] we used local (nearest-neighbor) communication in order to minimize the communication overhead and thus maximize the parallel efficiency. However, since the communications overhead are negligibly small compared to the TAD calculation time, for convenience in most of our parallel TAD simulations we have used global communication such that at the end of every subcycle every processor has information about the whole system. Since the number of nearest neighbors in our parallel TAD simulations is relatively large, we have found that at least for small and intermediate system sizes ($N_S\leq 36$) there is no difference between the parallel efficiency of parallel TAD runs using local communication and those using global communication. However, for very large system sizes we would expect that the use of local communication would lead to a significantly higher parallel efficiency, since in this case each processor is only slowed down by local fluctuations. Finally, we note that during each subcycle there may be small relaxation displacements in the minimum-energy positions of atoms that did not participate in any transitions. While displacements larger than a critical threshold may be important, in order to enhance the speed of the TAD simulations we have imposed a cutoff ϵ such that at the end of a subcycle any atomic displacements in the minimum-energy basin smaller than ϵ are ignored (i.e., the original positions at the beginning of the subcycle are preserved) while displacements larger than ϵ are accepted. In the present simulations, we typically chose a value $\epsilon=0.2$ Å. The use of such a cutoff can enhance the simulation speed since, in the absence of a nearby deposition or activated event in the previous cycle, it allows the TAD simulation to pick up where it left off because it recognizes that the state has not changed.

In previous parallel KMC simulations of a variety of systems[23,24,29] it was found that as long as the time interval τ is not too long, then there is excellent agreement between the parallel KMC results and the corresponding serial results. In particular, in general the cycle time should not be longer than the inverse of the fastest rate for "propagating" events such as monomer diffusion. The reason for this is that such a limitation on the cycle time ensures that there is on average *at most* one event of each type in each subcycle (for example, if monomer hopping is the fastest *possible* rate then each monomer in a given sublattice can make on average at

most one hop in a given cycle). However, in order to maximize the efficiency of the TAD simulations in each sublattice (i.e., the average number of actual events per processor per subcycle) and minimize the communication time overhead, one would like to have the largest possible value of τ that does not "corrupt" the time evolution. In the simulations presented here, with $T_{low} \leq 100$ K, we typically set $\tau \leq 10^{-4}$ s. We note that the time scale for a typical activated process for Cu(100) at 100 K (edge diffusion) is about 10^{-2} s (the corresponding energy barrier is about 0.2 eV) which is much longer than our cycle time, while the time scale for monomer diffusion on Cu(100) at this temperature is effectively infinite. In contrast, the energy barrier for monomer diffusion on a (111) facet is about 0.05 eV which corresponds to a hopping time which is significantly smaller than our cycle time. However, at low temperatures the size of such (111) facets is quite small. As a result we find that in our simulations the average number of actual diffusion events per cycle (not including low-barrier repetitive events which do not propagate) is relatively small, thus justifying the relatively long cycle time used. For example, in our simulations of deposition at 77 K with a deposition rate of 500 ML/s, the average number of accepted activated events per subcycle per processor (not including low-barrier repetitive events) was approximately equal to 0.05 which corresponds to significantly less than one event per processor per subcycle. In contrast, since our cycle time is equal to the inverse of the deposition rate (see Sec. II D below), there is on average one deposition event per subcycle.

D. Simulation details

In our simulations, the Cu-Cu interactions were modeled using a Voter-Chen embedded atom method (EAM) potential,[30,31] whose parameters are given in Ref. 5. The equations of motion were integrated using a Langevin velocity-Verlet algorithm[32] with time step $\Delta t = 4$ fs. The minimum prefactor was set to $\nu_{min} = 10^{11}$ s^{-1} and a confidence level of 95% ($\delta = 0.05$) was used.[33] The substrate slab was composed of five (100)-oriented layers in which the upper two layers were allowed to move, and as in a number of previous serial TAD simulations[18–20] the bottom three layers were fixed. We note that strictly, to eliminate all force errors, there should be enough fixed layers (five in this case) to include all atoms within $2r_{cut}$ of any moving atoms. However, as already discussed in Sec. II B, the errors introduced in the present study are minimal, and can be eliminated if desired (e.g., in a future study focused more on the physics) without changing the computational cost. The substrate was kept at constant temperature using a Langevin thermostat[32] with coupling constant $\alpha_L = 10^{12}$ s^{-1}. All simulations were carried out using the 1.2 GHz AMD and1.3 GHz Itanium 2 clusters at the Ohio Supercomputing Center (OSC).

Simulations were carried out both with and without deposition. In the case of deposition, the deposition events were simulated using ordinary molecular dynamics at temperature T_{low}, while an initial kinetic energy $K_i = 0.2$ eV (corresponding to $2k_BT_m$ where $T_m = 1356$ K is the melting temperature of Cu) was used for the depositing atom.[34] We note that the effects of steering due to short-range attraction[34,35] are naturally taken into account during the deposition process. We also note that in some cases, a particle deposited initially within a sublattice may end up in its ghost region due to interactions with neighboring particles, but this is still accepted as its final position.

In the case of deposition, the average time interval τ_{dep} between deposition events in each sublattice is given by the inverse of the deposition rate, i.e., $\tau_{dep} = (FA_s)^{-1}$ where F is the deposition flux in units of monolayers per second, $A_s = 18$ is the number of atoms per monolayer in each sublattice, and FA_s is the total deposition rate in each sublattice. However, since deposition is a random (Poisson) process, the distribution of time intervals Δt_{dep} between deposition events in a given sublattice is given by the expression $P(\Delta t_{dep}) = \tau_{dep}^{-1} \exp(-\Delta t_{dep}/\tau_{dep})$. Accordingly, the correct time interval between deposition events can be generated (see Ref. 23) using the expression $\Delta t_{dep} = -\ln(\xi)\tau_{dep}$, where ξ is a uniform random number between 0 and 1. In principle, a deposition event that occurs during a given subcycle should occur in the middle of the corresponding TAD simulation. However, since deposition in a given processor leads to a "new state" for that processor's TAD simulation, rather than interrupt a TAD subcycle by a deposition we instead deposit a particle or particles at the beginning of each subcycle during which deposition occurs. In order to ensure that on average no more than one deposition occurs at the beginning of each subycle, we have used cycle times $\tau < \tau_{dep}$.

III. RESULTS

A. Scaling behavior in the absence of deposition

In order to study the scaling behavior of our parallel TAD simulations as a function of system size, we first considered the simple case without deposition in which each processor contains one trimer in sublattice C and one monomer in sublattice D [see Fig. 1(b)]. Typical diffusion events found in this system are edge diffusion (energy barrier $E_e = 0.20$ eV) and monomer diffusion ($E_m = 0.53$ eV), although at 100 K, the latter is not active. Parallel TAD simulations were carried out for different numbers of processors $N_p = 1, 4, 9, 16$, and 36 corresponding to systems of size $6\sqrt{N_p}a \times 6\sqrt{N_p}a$. In units of the basic $l_p \times l_p$ processor size, this corresponds to systems of size $N_S = N_p$. In all cases, the total simulated time was 0.01 s, corresponding to ten cycles with cycle time $\tau = 10^{-3}$, while averages were typically taken over five or more runs. Simulations were carried out both with an assumed minimum energy barrier $E_{min} = 0.19$ eV [see Eq. (4)] as well as without ($E_{min} = 0$). For comparison, serial TAD simulations over the same time interval were also performed using systems of size $N_S = 1$ and 4 both with and without E_{min}. In all cases, periodic boundary conditions parallel to the (100) plane were assumed. For the case of a minimum energy barrier E_{min}, serial simulations were also carried out for a system size $N_S = 9$.

Figure 2 shows the corresponding results for the simulation time as a function of system size N_S, where N_S is the number of $l_p \times l_p$ units in the substrate. As expected, for a

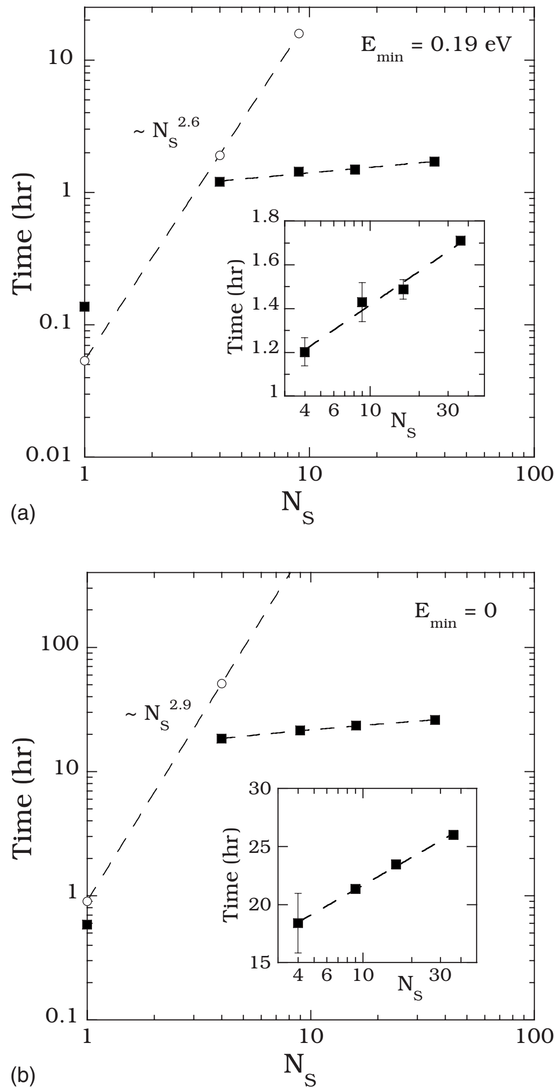

FIG. 2. Timing comparison between serial (open symbols) and parallel (closed symbols) runs for the case of surface relaxation as a function of system size N_S. (a) E_{min}=0.19 eV. (b) E_{min}=0. Insets are semilog plots demonstrating logarithmic scaling for the parallel case. Here, T_{low}=100 K, T_{high}=600 K, and the simulated time corresponds to 0.01 s.

system size N_S=1, the serial simulations are comparable in speed with and/or somewhat faster than the parallel simulations. However, for a system size N_S=4, the parallel simulations are already noticeably faster, especially in the case of regular TAD (E_{min}=0). Figure 2 also indicates that the serial simulation time t_{ser} increases as a power of the system-size N_S, i.e., $t_{ser}(N_S) \sim (N_S)^x$ where $x \simeq 2.6$ in the case of an assumed minimum energy barrier E_{min}, and $x \simeq 2.9$ in the more typical case of no assumed minimum energy barrier (regular TAD). In contrast, for $N_S \geq 4$ the simulation times of our parallel TAD simulations appear to increase only logarithmically with the system size N_S. As a result, for system sizes $N_S \geq 4$ the parallel simulations are significantly faster than the serial simulations. The logarithmic increase rather than saturation in the parallel timing with N_S is most likely due to the existence of global fluctuations in the TAD simulation times over all processors, which are expected to increase logarithmically with system size.[36]

Due to the relative homogeneity of the system being simulated, the fluctuations in the TAD simulation times between different processors for a given subcycle are generally relatively small. As a result, the simulation time for a 36-processor simulation is less than twice that for a 4-processor simulation. However, from N_S=1 to 4 there is a big jump in the parallel simulation time. This is due to the fact that for $N_S \geq 4$ the size of the TAD simulations in each processor (including ghost and fixed atoms) was $9l_s^2$, while for N_S=1 no fixed atoms were needed due to the periodic boundary conditions, and so the system size ($4l_s^2$) was significantly smaller.

B. Parallel TAD simulations of Cu/Cu(100) growth

1. Scaling behavior and optimal high temperature

We now consider the scaling behavior as a function of system size in the early stages of Cu/Cu(100) growth at low temperature. In particular, we first consider growth with substrate temperatures T_{low}=20, 40, and 77 K and deposition rate F=500 ML/s. In this case, to ensure that on average no more than one deposition occurs at the beginning of each subcycle, we have used cycle times $\tau < \tau_{dep}$ where $\tau_{dep} = (A_s F)^{-1} = 1.1 \times 10^{-4}$ s. In particular, two different cycle times $\tau = 4 \times 10^{-5}$ and 10^{-4} s were used. In order to examine the effects of the high temperature T_{high} on the simulation time, three different high temperatures were used corresponding to T_{high}=300, 400, and 600 K. For comparison, we have also carried out serial TAD simulations for the smallest system sizes (N_S=1, 2.25, and 4) with T_{low}=40 K and T_{high}=600 K. In the parallel simulations ($N_p = N_S$) the maximum coverage ranged from 0.5 to 1 ML, while in the serial simulations, due to their time-consuming nature, the maximum coverage for N_S=4 was only 0.37 ML. We note that in all our deposition simulations, no minimum energy barrier E_{min} was assumed.

Figure 3 shows a semilog plot of the corresponding parallel results for the simulation time per monolayer deposited as a function of the system size N_S for three different substrate temperatures (T_{low}=20, 40, and 77 K) with E_{min}=0 and $N_S = N_p$=4, 9, 16, 25, and 36. As can be seen, except for the case of T_{low}=40 K and T_{high}=300 K, for which the simulation time appears to saturate with increasing system size, all the timing results given in Fig. 3 exhibit a logarithmic increase with N_S, just as was found in the absence of deposition. Note that for T_{low}=40 K and T_{high}=600 K, the slope of the timing result with $\tau = 10^{-4}$ s is smaller than that with $\tau = 4 \times 10^{-5}$ s, indicating that as expected, a longer cycle time yields a better performance. The results shown in Fig. 4 also indicate that for T_{low}=40 K there is an optimal high-temperature T_{high}^{opt}=300 K. Accordingly, results for T_{low}=40 K and T_{high}=300 K are also shown in Fig. 3 (open diamonds). In this case, a cutoff $E_{reverse}$=0.01 eV was also included in the simulations (see Sec. III B 2) such that any events with reverse barriers less than $E_{reverse}$ were not accepted. As can be seen, the simulation speed is significantly faster than for the other cases shown.

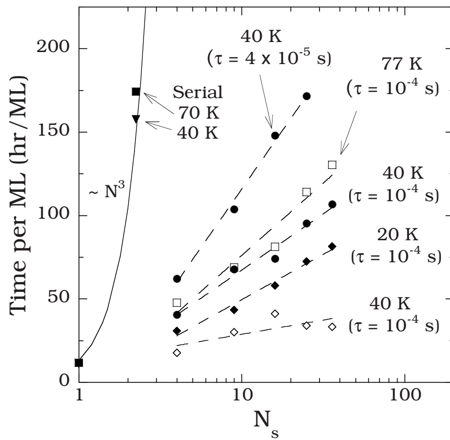

FIG. 3. Simulation time per ML in serial simulations (closed square and triangle, solid curve) and parallel TAD simulations (symbols with dashed line fits) of Cu/Cu(100) growth as function of system size N_S for T_{low}=20–77 K. Here, T_{high}=600 K (closed symbols), 400 K (open squares), and 300 K with $E_{reverse}$=0.01 eV (◇). The flux F=500 ML/s in all cases.

Also shown in Fig. 3 are the corresponding serial TAD results for T_{low}=40 and 70 K with T_{high}=600 K and N_S =1, 2.25, and 4 (closed triangles and closed squares). As for our results without deposition and no assumed minimum energy barrier, in this case the serial simulation time t_{ser} scales as $t_{ser} \sim (N_S)^x$ where $x \simeq 3.0$. We note that for a system size N_S=4, the serial TAD results were so slow that at T_{low} =70 K (40 K) only 0.3 ML (0.37 ML) were deposited. In particular, the time/ML for our serial runs with N_S=4 was approximately 588 h at 40 K and 689 h at 70 K. Thus already for a system size N_S=4 the parallel TAD simulations are significantly faster than the serial TAD simulations.

In order to further investigate the dependence of the optimal high-temperature on substrate temperature and deposition rate we have also carried out parallel TAD simulations with a fixed number of processors (N_p=4) for two different deposition rates (F=50 and 500 ML/s) and for three different deposition temperatures (40, 77, and 100 K) with different values of T_{high}. While the amount deposited ranged from 0.14 ML for T_{low}=100 K and T_{high}=200 K, to 3 ML for T_{low}=77 K and T_{high}=200 K, in most cases approximately 1 to 2 ML were deposited. Figure 4 shows a semilog plot of the corresponding results for the "coverage per unit (wallclock) time" or simulation speed as a function of T_{high}. As expected, the simulation speed decreases with increasing substrate temperature T_{low} as well as with decreasing deposition rate. However, for each substrate temperature, there is a distinct peak which indicates the existence of an optimal high temperature T_{high}^{opt} which increases with increasing T_{low}. For example, $T_{high}^{opt} \simeq 300$ K for T_{low}=40 K while $T_{high}^{opt} \simeq 500$ K for T_{low}=100 K. In contrast, while decreasing the deposition rate also significantly decreases the simulation speed for a fixed high temperature, it appears to have little effect on the optimal high temperature.

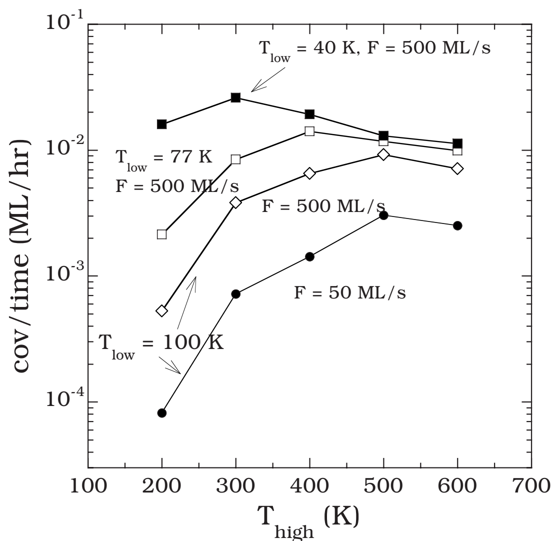

FIG. 4. Deposited coverage (ML) per unit simulation time as a function of T_{high} for three different substrate temperatures (40, 77, and 100 K) with F=50–500 ML/s and N_p=4.

2. Surface diffusion and repetitive events

In order to further understand the activated events which occur during growth, we have examined the number of accepted activated events N_{event} as a function of coverage for different substrate temperatures T_{low} and fixed system size (N_p=4), as well as for the case of a fixed substrate temperature (T_{low}=40 K) as a function of the number of processors. As shown in Fig. 5, the number of accepted events increases with coverage as well as with increasing substrate temperature. However, one particularly noticeable feature is the existence of sudden jumps in the number of accepted events as a function of coverage. Each of these jumps is due to the appearance of a state with low-barrier repetitive events which leads to a large number of events in a very short time, before eventually escaping to a state with higher barriers. As can be seen in Fig. 5, a very large fraction of the events observed in our parallel TAD simulations are due to such repetitive low-barrier transitions.

Figure 5 also indicates that the probability of such low-barrier repetitive events tends to increase with system size N_S as well as coverage; i.e., the more particles deposited, the higher the probability that a low-barrier state will be encountered. As already shown in Fig. 3, this may contribute to the slowing down of parallel TAD simulations with increasing system size. The number of such low-barrier repetitive events also increases significantly with increasing substrate temperature, thus significantly lowering the performance of our parallel TAD simulations. We note that in the scaling results shown in Fig. 3, the probability of such low-barrier repetitive events was somewhat reduced since the runs were restricted to coverages ranging from 0.5 to 1 ML. For the case of T_{low}=40 K and N_p=36 shown in Fig. 5(b) the simulation was terminated at a coverage of approximately 0.75 ML after the same repetitive event had exceeded the limit of 50 000 repetitions, which was selected as a stop parameter for that run. As discussed in Sec. II B, after the first ten repetitions our TAD simulations switch from regular TAD mode to "synthetic" mode, such that the repetitive

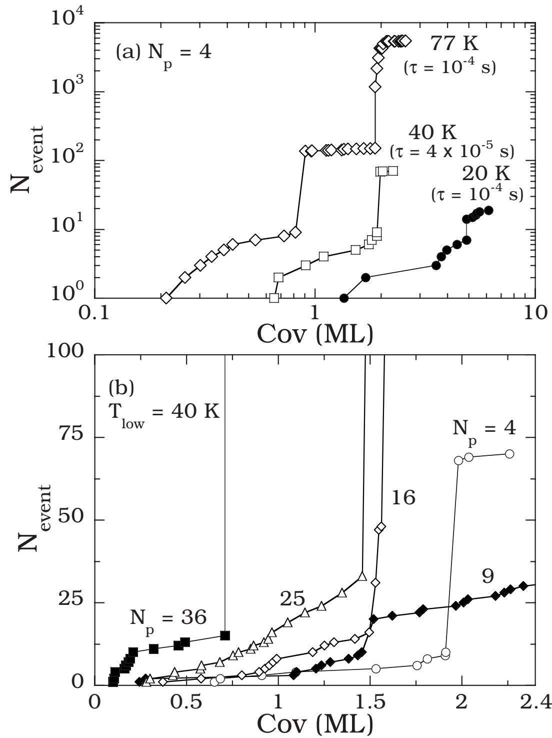

FIG. 5. Number of accepted events as a function of coverage. (a) T_{low}=20–77 K with N_p=4 and two different values of τ. (b) T_{low}=40 K with N_p=4–36 and $\tau=4\times10^{-5}$ s. In both cases, T_{high} =600 K and F=500 ML/s.

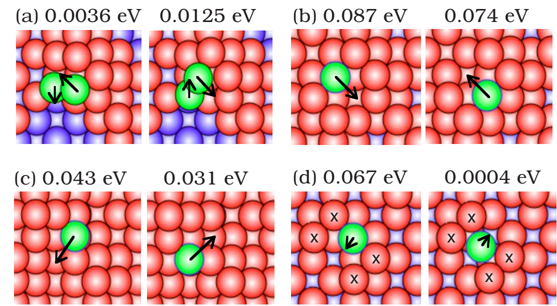

FIG. 6. (Color online) Typical low energy-barrier repetitive events observed at T_{low}=77. In each case the forward and reverse transitions are indicated (left and right pictures) while corresponding energy barriers are above each picture. Arrows represent direction of motion of atoms involved in each transition.

events are treated in a kinetic Monte Carlo fashion.[6] Thus while the time spent on the repetitive events is still significant, it is substantially less than would occur using regular TAD. In particular, once a transition pathway has been switched to synthetic mode, the direct cost of repeating that event a large number of times in synthetic mode is significantly reduced. However, there is still an indirect cost due to the fact that at T_{high} there are many attempted escapes over the low barrier. Even though this number is typically much smaller than the number of synthetic escapes, the system must be brought back into the state and rethermalized for a period of time after each attempted escape.

Figure 6 shows some typical low-barrier repetitive events found in our simulations of growth at 500 ML/s with a substrate temperature of 77 K. The energy barriers for each diffusion move, as well as the corresponding reverse move, are indicated. The configuration shown in Fig. 6(a) is particularly interesting since it represents dimer rotation with a very low energy barrier on a small (111) facet. However, it only repeats a few hundred times before the atoms involved diffuse down to a stable configuration on a lower terrace. In contrast, Fig. 6(b) shows an extremely stable repetitive event in which an atom performs "edge-diffusion" on a (100) terrace along and between two island edges (dimers). The repetitive events shown in Figs. 6(c) and 6(d) correspond to variations of this type of event. In Fig. 6(c) the initial and final sites of the diffusing atom have only three supports from the surface and thus the energy barriers for these moves are slightly smaller than those for the configuration given in Fig. 6(b). The repetitive event in Fig. 6(d) is unusual since it involves edge diffusion of an atom hopping from a fourfold hollow site to a symmetric configuration at a central bridge site surrounded by the atoms labeled with an ×, before either hopping with a very low energy barrier (E=0.0004 eV) back to the original fourfold hollow site, or onward (with the same energy barrier) to the fourfold hollow site on the other side. This particular configuration repeats thousands of times before escaping to a new configuration.

While such low-barrier repetitive moves can reduce the performance of our parallel TAD simulations, we expect that slowing them down or suppressing them, or equivalently "lumping" them together into one state[3] does not significantly alter the surface morphology. As already noted in Sec. III B 1, we have also carried out simulations in which a small cutoff $E_{reverse}\simeq0.01$ eV was included such that events with reverse barriers less than $E_{reverse}$ are not accepted. This effectively lumps the states together. As shown in Fig. 3 for the case of T_{low}=40 K with T_{high}=300 K, by suppressing the very lowest barrier repetitive events, the number of repetitive events was significantly reduced and we have obtained significantly improved parallel performance.

3. Strain relaxation

Since in our parallel TAD simulations each sublattice is surrounded by fixed atoms, and in addition at the end of a subcycle only displacements larger than a cutoff displacement ϵ=0.2 Å are accepted, it is possible that the system may build up some unrelaxed strain. In order to test for the existence of incomplete strain relaxation, we have carried out additional parallel TAD simulations of growth at T_{low} =20, 40, and 77 K with deposition rate F=500 ML/s and with system sizes ranging from N_S=4 to 36. At the end of each parallel TAD simulation, the whole system was relaxed via energy minimization and the change in positions before and after relaxation was measured.

Figure 7 shows the fraction $f_>(d)$ of atoms with relaxation displacements greater than d after strain relaxation for four different system sizes (N_S=4–36) and coverages (θ =1–13 ML) and at three different temperatures T_{low} =20, 40, and 77 K. As can be seen, for a coverage of 1 ML,

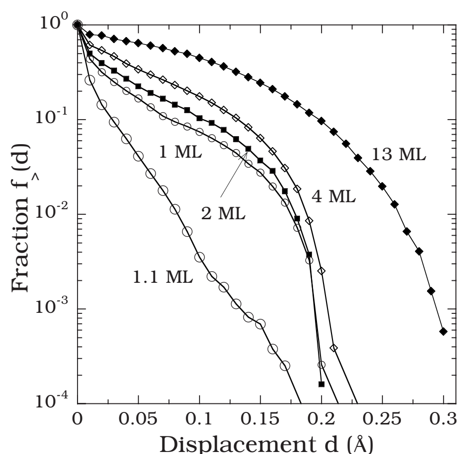

FIG. 7. Fraction $f_>(d)$ of atoms with displacement larger than d resulting from relaxation of multilayer films after growth of 13 ML (N_S=4 and T_{low}=20 K), 4 ML (N_S=16 and T_{low}=40 K), 2 ML (N_S=25 and T_{low}=40 K), and 1 and 1.1 ML (N_S=36 and T_{low}=77 K) as described in text. Here F=500 ML/s.

there is a small but noticeable fraction (approximately 10%) of atoms with displacements greater than 0.1 Å. However, for larger coverages this fraction is significantly larger, although except for the highest coverage of 13 ML, the fraction of atoms with a displacement larger than the cutoff displacement ϵ=0.2 Å is negligible. These results indicate that in order to more properly take such strains into account, it may be desirable to somewhat reduce the displacement cutoff ϵ and/or to perform a global relaxation relatively frequently, i.e., perhaps after every few tenths of a monolayer of growth. We note that including such periodic global relaxations in our parallel TAD simulations is relatively easy and, except for extremely large systems, should not significantly increase the simulation time. In order to test this, we have carried out an additional simulation in which we started out with a relaxed configuration after 1 ML, then grew an additional 0.1 ML with ϵ=0.2 Å, and then relaxed again to determine the displacement distribution (see Fig. 7). As can be seen, the displacement distribution now decays significantly more rapidly. In particular, the fraction of atoms with displacements greater than 0.1 Å is now approximately 0.4% while the maximum displacement (not shown) is now less than ϵ.

IV. CONCLUSIONS

We have presented a method for carrying out temperature-accelerated dynamics (TAD) simulations over extended length scales, thus offering a route to simulations over both large length and time scales simultaneously. In our method, a two-dimensional spatial decomposition is used along with the recently proposed synchronous sublattice (SL) algorithm. In test simulations of relaxation of a simple system configuration with sizes ranging from N_S=1 (corresponding to a system with lateral dimensions of $6a \times 6a$) to N_S=36 (corresponding to a system with lateral dimensions of $36a \times 36a$), we found that the parallel TAD simulation time increases logarithmically with increasing system size. A similar logarithmic scaling was also observed in parallel TAD simulations of the early stages of Cu/Cu(100) multilayer growth at a variety of temperatures. In contrast, the serial simulation time t_{ser} increases with a power law as a function of the system size N_S, i.e., $t_{ser} \sim (N_S)^x$ where $x \simeq 2.6$ or 3.0 depending on whether or not a minimum energy barrier is assumed.[37] Accordingly, for system sizes N_S=4 and larger, our parallel TAD simulations are significantly faster than the serial TAD simulations. In particular, for system sizes greater than N_S=9 (corresponding to a system with lateral dimensions of $18a \times 18a$), the serial simulations become prohibitive compared to the parallel simulations, while the parallel TAD simulation time increases relatively slowly for larger system sizes. This significantly improved scaling behavior is noteworthy because many problems of interest have been out of reach for serial TAD due to their required size.

We have also investigated the dependence of the performance of our parallel TAD simulations on a variety of simulation parameters. For example, in simulations of the early stages of low-temperature Cu/Cu(100) growth we have found that several factors besides the deposition rate and substrate temperature may play an important role. These include the high temperature T_{high}, the cycle-time τ, and the existence of very low energy barrier repetitive events. In particular, due to the relative computation costs of various elements of TAD depending on T_{high}, there is an optimal high temperature yielding the best performance for a given substrate temperature T_{low}. However, while the optimal high temperature increases with increasing T_{low} it does not exhibit a significant dependence on the deposition rate.

Due to the fact that it leads to less "interruption" in the TAD simulations in each processor, our results indicate that in general a longer cycle time improves the performance of our parallel TAD simulations. However, in our parallel TAD simulations with deposition, we have used a maximum cycle time corresponding to the inverse of the deposition rate (since for a longer cycle time there would either be multiple depositions at the beginning of each subcycle, or if the depositions were carried out purely randomly, the TAD would be interrupted by deposition events). In addition, we note that in previous parallel KMC simulations of film growth using the SL algorithm[23,24] it was found that as long as the cycle time is less than the inverse of the rate of the fastest possible activated event, then there is excellent agreement between parallel and serial simulations. Unfortunately, such a choice is not usually practical in TAD simulations because the fastest possible rate is not known *a priori*. However, for the highest temperature TAD simulations of growth presented here (T_{low}=77 K) and for the cycle times used, we have found that there is on average significantly *less* than one nonrepetitive activated event per subcycle. This indicates that the cycle time is sufficiently small and that as a result the accuracy of our parallel TAD simulations should not be affected by the SL algorithm. Nevertheless, in future applications of the SL algorithm to parallel TAD simulations it may be worth carrying out some more extensive tests to ensure that the time interval does not alter the dynamics of the growth process.

We have also found that the existence of repetitive low-barrier events can play an important role in reducing the

overall speed of parallel TAD simulations. Since there are many possible candidates for such events, there is no simple way to avoid or reduce them in advance in a TAD simulation unlike in a typical KMC simulation. However, by ignoring repetitive events with extremely small energy barriers $E_i < 0.01$ eV we have been able to obtain a significant speedup in our parallel TAD simulations. We note that the use of such a cutoff is particularly effective for a large number of processors since the possibility of having such low-barrier repetitive events in the system increases as the system size increases.

Finally, we have examined the possible accumulation of unrelaxed strain due to the use of our sublattice algorithm combined with the use of a displacement cutoff ϵ after each subcycle. Our results indicate that after a monolayer or more a small but non-negligible amount of strain can develop. However, we have also found that by relaxing the system more frequently, e.g., every 0.1 ML, the strain may be significantly reduced without significantly slowing down our parallel TAD simulations. We note that in the case of deposition, it should also be possible to significantly reduce the cutoff displacement ϵ without significantly affecting the parallel efficiency (since on average there is one deposition per cycle) and thus further reduce any strain-related effects. In this connection, we note that in several recent experiments on Cu/Cu(100) and Ag/Ag(100) growth,[38–40] a surprisingly large vacancy density has been observed at room temperature and below. In future work, we plan to perform parallel TAD simulations using optimized parameters and large system sizes in order to understand the detailed mechanisms of vacancy formation and coarsening observed in these experiments.

ACKNOWLEDGMENTS

This research was supported by NSF Grants No. CCF-0428826 and No. DMR-0606307 as well as by a grant of computer time from the Ohio Supercomputer Center. Work at Los Alamos National Laboratory (LANL) was supported by the United States Department of Energy, Office of Basic Energy Sciences, Division of Materials Sciences and Engineering, and by the Laboratory Directed Research and Development (LDRD) program. LANL is operated by Los Alamos National Security, LLC, for the National Nuclear Security Administration of the U.S. Department of Energy under Contract No. DE-AC52-06NA25396.

*yshim@physics.utoledo.edu
†jamar@physics.utoledo.edu
‡blas@lanl.gov
§afv@lanl.gov

[1] A. F. Voter, J. Chem. Phys. **106**, 4665 (1997).
[2] A. F. Voter, Phys. Rev. Lett. **78**, 3908 (1997).
[3] R. A. Miron and K. A. Fichthorn, Phys. Rev. Lett. **93**, 128301 (2004).
[4] R. A. Miron and K. A. Fichthorn, Phys. Rev. B **72**, 035415 (2005).
[5] A. F. Voter, Phys. Rev. B **57**, R13985 (1998).
[6] M. R. Sørensen and A. F. Voter, J. Chem. Phys. **112**, 9599 (2000).
[7] F. Montalenti and A. F. Voter, J. Chem. Phys. **116**, 4819 (2002).
[8] A. F. Voter, F. Montalenti, and T. C. Germann, Annu. Rev. Mater. Res. **32**, 321 (2002).
[9] G. Henkelman and H. Jonsson, J. Chem. Phys. **115**, 9657 (2001).
[10] G. Henkelman and H. Jonsson, Phys. Rev. Lett. **90**, 116101 (2003).
[11] O. Trushin, A. Karim, A. Kara, and T. S. Rahman, Phys. Rev. B **72**, 115401 (2005).
[12] A. Karim, A. N. Al-Rawi, A. Kara, T. S. Rahman, O. Trushin, and T. Ala-Nissila, Phys. Rev. B **73**, 165411 (2006).
[13] R. Marcelin, Ann. Phys. (Paris) **3**, 120 (1915).
[14] E. Wigner, Z. Phys. Chem. Abt. B **19**, 203 (1932).
[15] H. Eyring, J. Chem. Phys. **3**, 107 (1935).
[16] D. G. Truhlar, B. C. Garrett, and S. J. Klippenstein, J. Phys. Chem. **100**, 12771 (1996).
[17] R. Zwanzig, *Nonequilibrium Statistical Mechanics* (Oxford University Press, Oxford, UK, 2001).
[18] F. Montalenti, M. R. Sørensen, and A. F. Voter, Phys. Rev. Lett. **87**, 126101 (2001).
[19] J. A. Sprague, F. Montalenti, B. P. Uberuaga, J. D. Kress, and A. F. Voter, Phys. Rev. B **66**, 205415 (2002).
[20] F. Montalenti, A. F. Voter, and R. Ferrando, Phys. Rev. B **66**, 205404 (2002).
[21] B. P. Uberuaga, R. Smith, A. R. Cleave, F. Montalenti, G. Henkelman, R. W. Grimes, A. F. Voter, and K. E. Sickafus, Phys. Rev. Lett. **92**, 115505 (2004).
[22] M. Cogoni, A. Mattoni, B. P. Uberuaga, A. F. Voter, and L. Colombo, Appl. Phys. Lett. **87**, 191912 (2005).
[23] Y. Shim and J. G. Amar, Phys. Rev. B **71**, 125432 (2005).
[24] Y. Shim and J. G. Amar, Phys. Rev. B **73**, 035423 (2006).
[25] G. Nandipati, Y. Shim, J. G. Amar, A. Karim, A. Kara, T. Rahman, and O. Trushin (unpublished).
[26] G. Henkelman and H. Jonsson, J. Chem. Phys. **113**, 9978 (2000).
[27] G. Henkelman, B. P. Uberuaga, and H. Jonsson, J. Chem. Phys. **113**, 9901 (2000).
[28] S. Goedecker, F. Lançon, and T. Deutsch, Phys. Rev. B **64**, 161102(R) (2001).
[29] F. Shi, Y. Shim, and J. G. Amar, Phys. Rev. E **76**, 031607 (2007).
[30] A. F. Voter and S. P. Chen, Mater. Res. Soc. Symp. Proc. **82**, 175 (1987).
[31] A. F. Voter, in *Intermetallic Compounds: Principles and Practice*, edited by J. H. Westbrook and R. L. Fleischer (Wiley, New York, 1995), Vol. 1, p. 77.
[32] M. P. Allen and D. J. Tildesley, *Computer Simulations of Liquids* (Oxford, New York, 1987), p. 263.
[33] We note that in some cases (see, for example, Ref. 18 where a value of δ ranging from 0.01 to 0.03 was used) one might wish to use a smaller value of δ than the value (δ=0.05) used here. However, since here we are primarily interested in testing the scaling and efficiency of our algorithm, we have used a slightly higher value in order to speed up our simulations.
[34] J. Yu and J. G. Amar, Phys. Rev. Lett. **89**, 286103 (2002).

[35] F. Montalenti and A. F. Voter, Phys. Rev. B **64**, 081401(R) (2001).
[36] Y. Shim and J. G. Amar, Phys. Rev. B **71**, 115436 (2005).
[37] As noted in Sec. III, the serial scaling may be improved if localized saddle searches are employed, but will still be significantly worse than the logarithmic scaling obtained in our parallel TAD simulations.
[38] C. E. Botez, W. C. Elliott, P. F. Miceli, and P. W. Stephens, Phys. Rev. B **66**, 075418 (2002); **66**, 195413 (2002).
[39] C. E. Botez, K. Li, E. D. Lu, W. C. Elliott, P. F. Miceli, E. H. Conrad, and P. W. Stephens, Appl. Phys. Lett. **81**, 4718 (2002).
[40] C. Kim, R. Feng, E. H. Conrad, and P. F. Miceli, Appl. Phys. Lett. **91**, 093131 (2007).