# A Rejection Scheme for Off-Lattice Kinetic Monte Carlo Simulation

Hamza M. Ruzayqat* and Tim P. Schulze*

*Department of Mathematics, University of Tennessee, Knoxville*

E-mail: ruzayqat@math.utk.edu; schulze@math.utk.edu

**Abstract**

We introduce a new kinetic Monte Carlo (KMC) algorithm for off-lattice simulation. In off-lattice KMC one needs to calculate the rates for all possible moves from the current state by searching the energy landscape for index-1 saddle points surrounding the current basin of attraction. We introduce a rejection scheme where the true rates are replaced by rate estimates. This is done by first associating each saddle point with the atom that would move the most if that transition were to take place, then constructing an estimate for the total rate associated with each atom by using a nearest-neighbor bond count. These estimates allow one to select a set of possible transitions, one of which is accepted or rejected based on a localized saddle point search focused on a particular atom. In principle, this allows a performance boost that scales with the number of particles in the system. We test the method on a growing two-species nano-cluster, and find we can reduce computation time by ninety percent for clusters that contain around fifty-five particles, and ninety-six percent for clusters that contain around sixty-five particles.

# 1 Introduction

While most Kinetic Monte Carlo (KMC) simulations are lattice based, many important technological applications involve multi-component systems where lattice mismatch leads to elastic strain and crystal defects,[1] neither of which can be accurately modeled with a lattice based approach. Off-lattice Kinetic Monte Carlo (OLKMC), initially developed by Henkelman and Jónsson,[2] is aimed at overcoming these limitations. Fully general off-lattice simulations make use of either an empirical potential or an even more costly density functional theory calculation, seeking to exhaustively calculate the transition path to all of the neighboring states within the multi-particle configuration space.

A fully implemented off-lattice simulation is an enormously complex task when compared to lattice based simulations, where rates can be precomputed and stored. So much so that KMC simulation loses much of its utility and applications of these methods are limited to systems with only a few hundred atoms, simulated for much shorter times, and at much greater computational cost. Even sophisticated approaches that maintain catalogs of previously seen environments[3–5] suffer from these limitations when compared to the simple bond counting schemes.

The severe limitations of OLKMC has lead to the development of simplified approaches that combine some aspects of lattice- and off-lattice models.[6–9] The rigid lattice of traditional KMC is replaced by a network of linear springs that are allowed to deform so as to minimize the system's potential energy. An important technique for accelerating these weakly off-lattice simulations was the development of a rejection-based algorithm that makes use of rate approximations that are similar to the models used in lattice-based simulation.[10] These weakly off-lattice approaches have been used to study the effect of lattice mismatch during heteroepitaxial film growth.[11–13] However, this particular approach cannot capture effects due to large displacements, like the formation of dislocations and other lattice defects, nor effects due to the concerted movement of multiple atoms. These are limitations that we seek to overcome by implementing a similar strategy for OLKMC.

The main idea is to partition the set of all transitions out of the current state into $N_p$ categories, where $N_p$ is the number of particles in the system, and to do this in a way that allows the rates to be found using localized searches centered on a particular particle. This is accomplished by first associating each transition with the particle that moves the largest distance when the system is moved from the initial state to the transition state. This retains the essential simplification of lattice-based KMC models based on single-particle moves while allowing for more complicated, multi-particle moves. Rate estimates for all moves associated with a given atom are then constructed based on local environments in a way that also mimics the bond-counting approach in a typical lattice-based model. These rate estimates reflect the intuition that loosely coordinated atoms are much more likely to reconfigure than fully coordinated atoms in the interior of a crystal. Together, the partitioning and rate estimates allow one to select a candidate event without doing the costly saddle point search for each atom in the system. In principle, this allows for an $O(N_p)$ improvement in performance.

In the next section we review the basic OLKMC procedures. In section 3, we introduce the rejection scheme. In section 4, we demonstrate the method by simulating the growth of two-species nanoclusters with core shell structures. This is a rather challenging system compared to that typically studied using OLKMC. Indeed, the fully implemented algorithm that is used for comparison purposes has to be abandoned once the system contains around seventy particles. In the final section we give some brief concluding remarks and point toward further improvements that may be possible in the future.

## 2 Off Lattice KMC

Before discussing the rejection algorithm in the next section, we briefly review the components of a fully implemented, rejection-free OLKMC. KMC simulation of crystal growth is motivated by observations of molecular dynamics simulations, relying on transition state theory (TST) to provide an approximate model.[2,13–15] The essential observation is that the

system spends most of its time randomly oscillating within the $N_p$-particle, $dN_p$-dimensional configuration space about a local minimizer $\mathbf{X}_i \in \mathbb{R}^{dN_p}$ of the system potential energy, $U(\mathbf{X})$, with rare transitions between basins of attraction. For the system to move from basin $i$ to basin $j$, it has to overcome a minimum energy barrier $\Delta U_{ij}$. If $k_B T$ is the energy scale defined by the temperature of the system, then the harmonic approximation to TST [16,17] estimates the rate $R_{ij}$ at which the transition occurs as

$$R_{ij} = K \exp(-\Delta U_{ij}/k_B T), \tag{1}$$

where $K$ is a prefactor,[18] that we take to be a constant scaled to one.

These observations give rise to an alternative model where the Newtonian dynamics is replaced by a Markov-chain, with the system making relatively rare, random transitions between states, represented by local minimizers $\mathbf{X}_i$ in the system's configuration space, at rates $R_{ij}$ calculated from (1). More specifically the energy barrier

$$\Delta U_{ij} = U(\mathbf{X}_{ij}) - U(\mathbf{X}_i), \tag{2}$$

requires locating both the initial local minimum, $\mathbf{X}_i$, and the index-1 saddle point, $\mathbf{X}_{ij}$ (where $\nabla U = 0$ and all but one of the principal curvatures are positive), separating the basins of attraction. Note that these local minima and saddle points are, in principle, determined by the motion of all of the particles simultaneously within the configuration space.

After enumerating the full set of transition rates and relabeling them using a single index: $\{r_n \equiv R_{ij}^n\}$, a single iteration of an OLKMC simulation is described by:

**Algorithm 1**

1. Calculate rates $r_i$ for each transition accessible from the current configuration.
2. Calculate partial sums $P_n = \sum_{i=1}^{n} r_i$, $n = 1, 2, \ldots, N$.
3. Generate a uniformly distributed random number $r \in [0, P_N)$.

4. Locate interval $I$ such that $P_{I-1} \leq r < P_I$.

5. Update the physical time $t \leftarrow t + \Delta t$ with $\Delta t = -\ln r'/P_N$, where $r'$ is a uniformly distributed random number in $(0, 1]$.

6. Move the system to the designated transition state, perturb away from the current local minimum, and relax to the new configuration.

Note that in Step 5 since the average value of $\ln x$ over the interval $(0, 1]$ is $\int_0^1 \ln x \, dx = -1$, the same average time scale can be obtained with $\Delta t = 1/P_N$. Steps 1 and 6 are tremendously costly compared to lattice-based simulations. So much so that KMC loses much of its utility in that the system size and the number of iterations that can be simulated are greatly reduced. This is particularly so with the first step, as the number of saddle points grows rapidly with the number of particles in the system. Saddle point searches are typically done with some sort of eigenvector climbing algorithm, with a computational cost similar to minimization using nonlinear conjugate gradient. In this paper we implement a version of the Dimer method introduced by Henkelman and Jónsson.[19–22]

In principle, the model requires locating all saddle points connected to the current basin of attraction. In practice, there is no way of knowing for certain when this has been achieved. This error is, in some sense, controllable, in that one can increase the number of attempts at finding new saddle points until a point of diminishing returns is acheived. Some recent work[23] seeks to make a more exhaustive exploration of the local potential energy surface, but, as with so much of the work on OLKMC, seems to be limited to very small systems for the time being. Here, we follow a practice similar to that of Henkelman and Jónsson[2] and initiate a large number of searches by randomly perturbing the system about the current state. More specifically, let $\mathbf{X} = \{\boldsymbol{x}_i \in \mathbb{R}^3\}_{i=1}^{N_p}$. For each atom $j$, we perturb the entire system with a magnitude that decays with increasing distance:

$$\boldsymbol{x}_i^k = \boldsymbol{x}_i + \frac{k\sigma}{N_g} \hat{\boldsymbol{n}}_i^k \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2\right) \qquad \text{for all } i = 1, \cdots, N_p, \tag{3}$$

where $N_g$ is the number of initial guesses, $k = 1, \cdots, N_g$ is the guess number, $\hat{\boldsymbol{n}}_i^k$ is a random unit vector in $\mathbb{R}^3$ and $\sigma$ is a length scale parameter that will be discussed later. This reflects the fact that most configuration changes are localized about a single particle or a small group of particles.

This procedure will find some saddle points that do not connect to the system's initial basin of attraction. Thus, upon finding a saddle point, one must requench the system starting from the newly found saddle point to ensure that the resulting saddle point is *connected*. One must also scan the list of previously acquired connected saddle points to prevent duplicates.

For the final step, we initialize a nonlinear conjugate-gradient minimization scheme near the chosen saddle point. We perturb this initial condition slightly in the direction away from the initial configuration and monitor the progress with a strict descent requirement and maximum displacement threshold, with the aim of guiding the system into the neighboring basin of attraction.

Another difficulty with general OLKMC procedures is what is known as the "small barrier problem".[24] Occasionally the system will reside in a basin of attraction with one or more extremely shallow minima, the crossing of which has little impact on the configuration. The small barrier means that the rate will be extremely high, and it is highly probable for the events associated with shallow barriers to be selected. When the reverse process also has a shallow barrier, this can lead to many wasted iterations as the system makes insignificant oscillations before a transition that fundamentally changes the configuration finally occurs. For this reason, we implement our OLKMC with a minimum barrier size chosen to reflect barriers that are typical for the surface motion of a single, loosely bonded atom, *e.g.* what one would refer to as an "adatom" in the context of epitaxial growth.

# 3 Rejection Algorithm

In a rejection algorithm, one samples a rate distribution formed from upper bounds on the actual rates, $\hat{r}_n \geq r_n$, and rejects the selected move with the appropriate frequency so that a stochastic process with the original rate distribution is formed. The simplest examples of this use a single global upper bound $\hat{r} \geq r_n, \forall n$. One can then select a candidate event using a randomly selected integer $1 \leq n \leq N$, where $N$ is the total number of events. This avoids all but one of the rate calculations in Step 1 of Algorithm 1, but at the expense of additional random number generation when events are rejected.

The overall efficiency of a rejection scheme,

$$\mathcal{E} = \frac{\text{accepted trials}}{\text{attempts}},$$

can be very low when the rate distribution has a wide range, as is often the case for KMC simulations due to the exponential dependence of rates on the energy barrier $\Delta U$. For lattice-based simulations, rejection-free KMC is therefore often superior because the cost of random number generation is high compared to calculating rates. The expense of rate calculations in OLKMC suggests that even an inefficient rejection scheme may be superior to the rejection-free scheme outlined in Algorithm 1. There is, however, a fundamental difficulty in implementing rejection for OLKMC in that determining the number and description of the events to be sampled relies on the same costly saddle-point searches required for the rejection-free algorithm. Below, we introduce a means of circumventing this need by partitioning the set of possible transitions so that each transition is associated with a uniquely defined *key* atom. In addition to making a rejection scheme viable, we will see that this also makes it more efficient by tailoring rate estimates to local environments. To this end, we will borrow the notion of a bond-counting formula from lattice based simulation, with an eye toward using this as a rate estimate rather than a rate model. This same strategy was used effectively in the weakly off-lattice models for strain mentioned earlier.[10–12]

Unlike lattice-based models with predefined event catalogs, neither the number nor the nature of transitions at a given time step is known *a priori* in OLKMC. Acquiring this information requires an exhaustive saddle point search like that in Step 1 of Algorithm 1, a calculation that would defeat the purpose of the rejection algorithm. In order to extend the rejection scheme to OLKMC, we partition the set of connected saddle points into localized subsets by associating moves with their key atom. We define this as the atom whose position changes by the greatest magnitude when it is moved from the configuration of the current local minimum to the saddle point configuration, and refer to the associated set of saddle points as *connected key* saddles. For a system with $N_p$ particles this has the effect of partitioning the entire set of transitions into $N_p$ subsets, each of which represents all of the moves for which one particular atom is the key atom. For moves that are essentially single atom hops, this will correctly associate the event with the hopping atom, while providing a natural generalization for more complicated, multi-atom moves. One now needs to over-estimate the sum of the rates for all configuration changes associated with a given key atom. If this can be done reliably, one can then choose a candidate event based on the estimated rates. After a candidate is chosen, one then calculates the total transition rate for that particular atom, accepting or rejecting the move with probability:

$$P(\text{acceptance}) = \frac{\text{true rate}}{\text{approximate rate}}.$$

This avoids the tremendously more costly need to calculate all of the rates for the entire system before selecting a move, and it does so with zero error as long as the estimates are upper bounds for the true rates.

The rate estimates we use attempt to model a lower bound for the smallest significant barrier (i.e. attempting to avoid the small barrier problem discussed above) $\check{U}$ and an upper bound $\hat{N}$ for the number of transitions that have this energy scale. To allow for greater flexibility in protecting against the estimate being lower than the true rate, we also include

an additive constant $\hat{C}$ in the estimate:

$$\hat{r}_j = K\hat{N}_j \exp\left(-\check{U}_j/k_BT\right) + \hat{C}, \qquad j = 1, \cdots, N_p. \tag{4}$$

The energy barrier bound is based on a generalized notion of a nearest neighbor. For our off-lattice model, we define a nearest neighbor as an atom lying within a distance, $d_1$, slightly larger than the lattice spacing of a perfect crystal. Similarly, we define second and third neighbor distances $d_2$ and $d_3$. We use an estimate where $\check{U}_j$ is linear in the number of first, second and third nearest neighbors for each species.

Next, we provide an outline of the rejection algorithm. We assume that the system is initialized at time $t$ to an arbitrary local minimum.

**Algorithm 2**

1. Calculate rate estimates $\hat{r}_j$, $j = 1, \cdots, N_p$, using Eq. (4)
2. Calculate partial sums $P_n = \sum_{j=1}^{n} \hat{r}_j$, $n = 1, 2, \ldots, N_p$.
3. Generate a uniformly distributed random number $r \in [0, P_{N_p})$.
4. Locate interval $J$ such that $P_{J-1} \leq r < P_J$.
5. Update the physical time $t \leftarrow t + \Delta t$ with $\Delta t = -\ln r'/P_{N_p}$, where $r'$ is a uniformly distributed random number in $(0, 1]$.
6. Perform *local* saddle point searches centered on atom $J$ as follows:
   (a) For $k = 1, \cdots, N_g$ initiate dimer searches with initial guesses as in Eq. (3).
   (b) Sort through the resulting saddle points so that $\{\mathbf{X}_J^1, \mathbf{X}_J^2, \cdots, \mathbf{X}_J^M\}$ is the set of distinct connected key saddles.
7. Calculate the true rates $r_J^i = K \exp\left(-\Delta U_J^i/k_BT\right)$, $i = 1, 2, \ldots, M$ for moves in which atom $J$ is the key atom.

8. Calculate partial sums $p_i = \sum_{n=1}^{i} r_J^n$, $i = 1, 2, \ldots, M$.

9. If $r - P_{J-1} > p_M$, reject the event; set $\hat{r}_J = p_M$; return to Step 2.

10. Otherwise, locate the interval $I$ such that $p_{I-1} \leq r - P_{J-1} < p_I$.

11. Move the system to saddle-point configuration $\mathbf{X}_J^I$, perturb away from the current local minimum, and relax to the new local minimum.

The success of the method hinges on Steps 1 and 6. If the rate estimates in Step 1 are lower than the actual sum of rates, the algorithm is no longer equivalent to a fully implemented OLKMC, as the corresponding events will be undersampled. While this is undesirable, it has an effect similar to other sources of error inherent to OLKMC. This undersampling error can be monitored and controlled to some extent in that it will be detected a certain fraction of the time. If $\hat{R}_E$ is the sum of the rate estimates for the atoms with rate estimates that are too small, then the probability that one of these atoms is selected for a saddle search is $\frac{\hat{R}_E}{P_{N_p}}$. These instances can be counted and used as a metric for upwardly adjusting rate estimates. Note, however, that the selection of an undersampled event is not an error—the error is a failure to select such events sufficiently often. The error can be more accurately measured by calculating

$$E = \sum_{i=1}^{N_p} \max(0, r_i - \hat{r}_i).$$

When events are undersampled we introduce an error with probability $\frac{E}{P_{N_p}+E}$ relative to a simulation with the minimal correction applied to the errant rates, *i.e.* $\hat{r}_i \leftarrow r_i$ whenever $\hat{r}_i < r_i$. The error cannot be computed on every iteration without losing the advantages of the rejection scheme, as one has to loop through the atom list and calculate the true rate for each atom and its rate estimate. However, it can be monitored for some small fraction of the time steps. While undersampling can be reduced or even eliminated by using sufficiently generous rate estimates, this comes at the expense of increased rejection due to oversampling. Specifically, we will reject an event with probability $1 - \mathcal{E} = 1 - \frac{R}{P_{N_p}}$, where $R$ is the sum

of the actual rates. In our simulations, we monitor both our undersampling error and the efficiency.

In Step 6, we wish to calculate the sum of all the rates for which atom $J$ is the key atom. We do this by perturbing the system about atom $J$ as in Eq. (3) to create a list of initial configurations for the dimer search. It is possible that this will miss some moves which will introduce an error similar to that discussed above in the context of the rejection-free scheme. This error can be decreased by creating another list of initial configurations by perturbing the system about a neighboring atom. This will reduce the efficiency: As we make our search for connected key saddle points more exhaustive, we will produce costly duplicates and saddles associated with atoms other than the candidate atom.

In Step 9, $p_M$ is the sum of the true transition rates for which the candidate atom is the key atom. When we reject a move where this atom is key we can set the estimated rate $\hat{r}_J$ to this sum, so that transitions associated with this atom will not be rejected on the next trial. This normally does not save a significant amount of computation, as it is likely that a different key atom will be selected on the next iteration, but it will occur more often when there is a single loosely-bonded atom at the surface. In the cases where the same key atom is chosen on a subsequent trial, we can also re-use saddle point information obtained for the rejected trial. Similarly, on any rejected trial, we can retain the information about saddle points associated with atoms other than the candidate atom, although this will only have a slight impact on performance.

When parallel resources are available, we suggest performing both the full OLKMC and the rejection scheme by distributing the $N_g$ dimer searches for each atom over the available processors, keeping in mind that one has to remove the duplicates before calculating the true rates. However, the boost in performance will still be $O(N_p)$. Suppose the computational cost to perform the rejection scheme on one processor is $O(N_g)$, then it is $O(N_p N_g)$ for the full scheme. When $n$ processors are available, the cost for the rejection scheme is $O(N_g/n)$ and $O(N_p N_g/n)$ for the full scheme.

# 4 Results

In this section we demonstrate the new algorithm by simulating the growth and evolution of a two-species cluster with an emerging core-shell structure. For this, we use a Lennard-Jones potential, modified for two interacting species, type A and type B.[25–27] The total energy of a system of $N_p$ particles interacting by the Lennard-Jones potential is given by

$$U(\mathbf{X}) = \sum_{i<j}^{N_p} \phi(r_{ij}), \quad \text{where } \phi(r_{ij}) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right],$$

$\mathbf{X} \in \mathbb{R}^{3N_p}$ is the current configuration in three dimensions, $r_{ij}$ is the distance between atoms $i$ and $j$, $\phi(r_{ij})$ is the interaction potential, $\sigma_{ij}$ is the distance at which $\phi(r_{ij})$ is zero and $\epsilon_{ij}$ is the chemical bond energy. We will use the Lorentz-Berthelot mixing rules[25,26] and take

$$\sigma_{ij} = \begin{cases} \sigma_A & \\ \sigma_B & \\ \frac{1}{2}(\sigma_A + \sigma_B), & \end{cases} \qquad \epsilon_{ij} = \begin{cases} \epsilon_A & \text{if both atoms are type A,} \\ \epsilon_B & \text{if both atoms are type B,} \\ \sqrt{\epsilon_A \epsilon_B}, & \text{if the atoms are different.} \end{cases}$$

We omit any truncation of the potential, but for larger systems we could employ the standard practice of introducing a cutoff radius that is often chosen to be three or four times a typical bond-spacing. To remove translational and rotational degrees of freedom, one particle is constrained to the origin, a second to a line passing through the origin, and a third to a plane containing this line. This is equivalent to adjusting the frame of reference to satisfy these constraints.

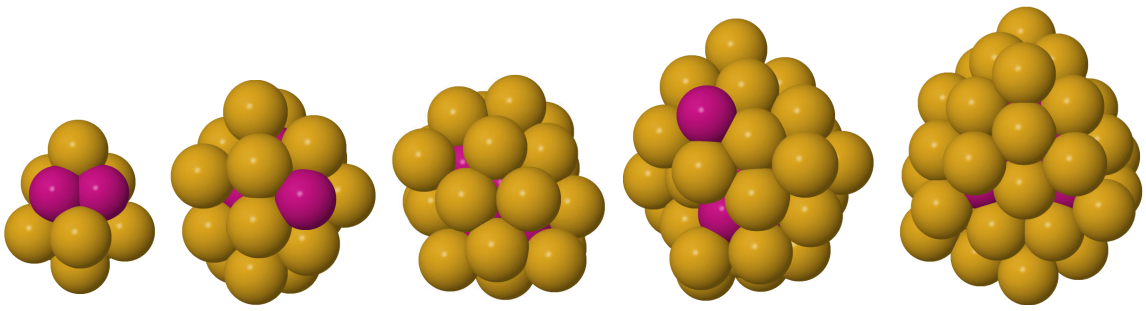
For a particle of type $A$ or $B$, we denote its nearest neighbors of species $\alpha$ by $n_k^{(\alpha)}$ where $k = 1, 2, 3$ correspond to the first, second and third nearest neighbors, respectively. In Eq. (4), we take $\hat{N}_i = 4\sum_{i=1}^{3}\sum_{\alpha\in\{A,B\}} n_i^{(\alpha)}$, $\hat{C} = 1.955$ and the energy barrier bound

$\check{U}_i = -1.2(B_1 + B_2)$, where

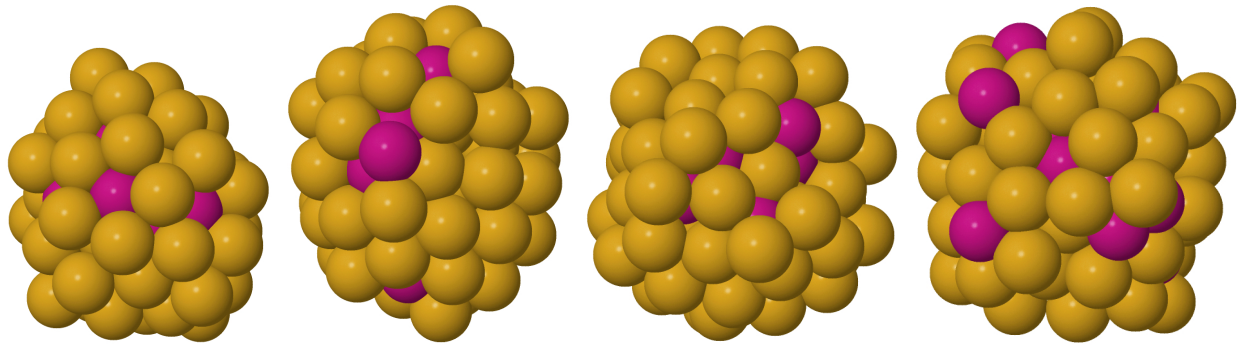
$$B_1 = \epsilon_\alpha(\lambda_1 n_1^{(\alpha)} + \lambda_2 n_2^{(\alpha)} + \lambda_3 n_3^{(\alpha)}),$$
$$B_2 = \sqrt{\epsilon_\alpha \epsilon_\beta}(\lambda_1 n_1^{(\beta)} + \lambda_2 n_2^{(\beta)} + \lambda_3 n_3^{(\beta)} + \lambda_4),$$

with $\alpha = \text{species}(i)$, $\beta \neq \alpha$, $\lambda_1 = \lambda_2 = 0.5$, $\lambda_3 = 0.8$ and $\lambda_4 = -5.9$.

Deposition is modeled by adding an additional rate, $r_{\text{dep}}$, to the rate table in Steps 1 and 2. When a deposition occurs, the species is selected so that the ratio is three A-particles for every B particle, and the appropriate particle is placed at a randomly selected solid angle a distance $d$ from the origin. The coordinates of this particle are relaxed toward a local minimum by steepest descent while constraining the remaining particles. After the constrained minimum is reached, the full system is relaxed by a conjugate gradient search, with care taken to monitor for descent or large moves, so that the particle settles into a local minimum without significantly disturbing the prior configuration.



(a) $A_9B_3$ (b) $A_{18}B_6$ (c) $A_{27}B_9$ (d) $A_{36}B_{12}$ (e) $A_{45}B_{15}$

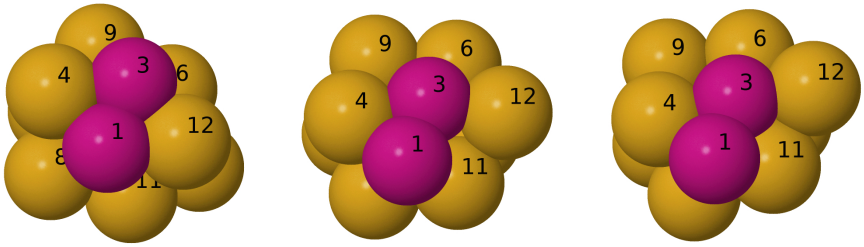


(f) $A_{54}B_{18}$ (g) $A_{63}B_{21}$ (h) $A_{72}B_{24}$ (i) $A_{75}B_{25}$

Figure 1: Snapshots at different times during the growth of a two species cluster, $A_{75}B_{25}$, with random deposition at rate = 0.85. The Lennard-Jones parameters are taken to be $\epsilon_A = \epsilon_B = 0.25$ and $\sigma_A = 1.3$ & $\sigma_B = 1$. Both the dimer and the conjugate gradient algorithms are terminated once the $L^2$ norm of the gradient is less than $10^{-3}$ or the maximum number of iterations is achieved. The view in these snapshots is chosen so that $B$ atoms appear clearly.

Figure 1 shows nine snapshots of the growth process, with a new snapshot selected after 12 particles have been added to the system. The larger particles are shown in gold and tend to evolve toward the outer shell. This tendency is increased if we slow the deposition rate, allowing more diffusive transitions between deposition events. These simulations represent a significant challenge for OLKMC. When the cluster is small, moves are highly concerted, with the transitions often resulting in all of the particles moving a significant distance as shown in Figure 2.

In Table 1, we record the efficiency $\mathcal{E}$ and the error $E$ of Algorithm 2, as defined in the previous section, for clusters in Figure 1, along with the scaled physical time, CPU time in hours, and the number of hops at which these clusters are obtained. We see that the efficiency is close to 1/2 for the larger clusters, meaning that, on average, we reject about every other candidate event. The error, as defined above, is around 4% for the larger clusters. As described earlier, this can be improved upon, but at the cost of increased rejection, by providing larger rate estimates.




(a) Minimum 1 (b) Saddle Point (c) Minimum 2

Figure 2: Example of a concerted move that includes the movement of all atoms except the constrained atom at the origin.

Table 1: The first two columns show the efficiency and error of Algorithm 2 for clusters in Figure 1, calculated via the rate estimate formula in Eq. (4) with $N_g = 200$ per atom. The last three columns are the scaled physical time, CPU time, and the number of hops at which these clusters were formed, respectively.

| Subfig. | Efficiency $\mathcal{E}$ | Error $E$ | Scaled Phys. time | CPU time (hrs) | No. of hops |
|---|---|---|---|---|---|
| (a) | 0.065 | 0.000 | 9.433 | 0.0985 | 139 |
| (b) | 0.25 | 0.007 | 23.680 | 2.436 | 872 |
| (c) | 0.32 | 0.000 | 42.796 | 13.812 | 2073 |
| (d) | 0.44 | 0.016 | 54.611 | 25.739 | 3081 |
| (e) | 0.44 | 0.011 | 71.050 | 63.756 | 4715 |
| (f) | 0.40 | 0.009 | 82.940 | 86.683 | 6153 |
| (g) | 0.45 | 0.028 | 94.000 | 142.453 | 7463 |
| (h) | 0.53 | 0.036 | 110.726 | 257.608 | 9796 |
| (i) | 0.54 | 0.041 | 114.679 | 309.312 | 10359 |

In Figure 3, we plot the number of events executed as a function of the scaled physical time for three realizations of the full OLKMC and the rejection-based algorithm as a means of demonstrating their near equivalence. Again, if one could achieve exhaustive saddle-point searches and strict bounds for the rate estimates, the two algorithms are stochastically equivalent. In Figure 4, we plot the CPU time as a function of the scaled physical time for this same set of realizations as a way of demonstrating the increased speed of the rejection scheme. For cluster sizes around fifty-five particles, the rejection algorithm is about ten times faster, and when they are around sixty-five particles, it is thirty times faster. This factor will continue to increase with larger cluster sizes, as the local search regions become a smaller fraction of the entire domain.

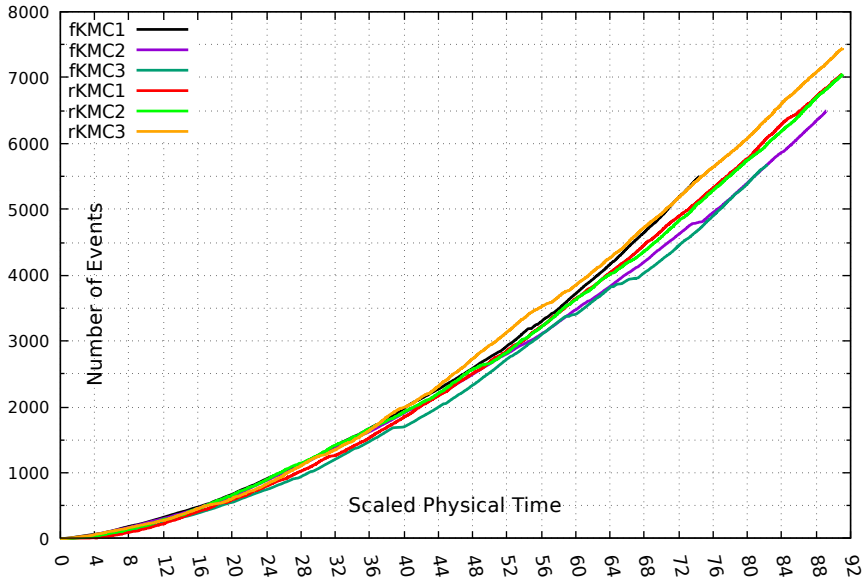



Figure 3: As a way of demonstrating the near equivalence of the two algorithms, this figure shows the number of events executed as a function of scaled physical time for three realizations of each algorithm. Different colors correspond to different seeds used for the random number generator.

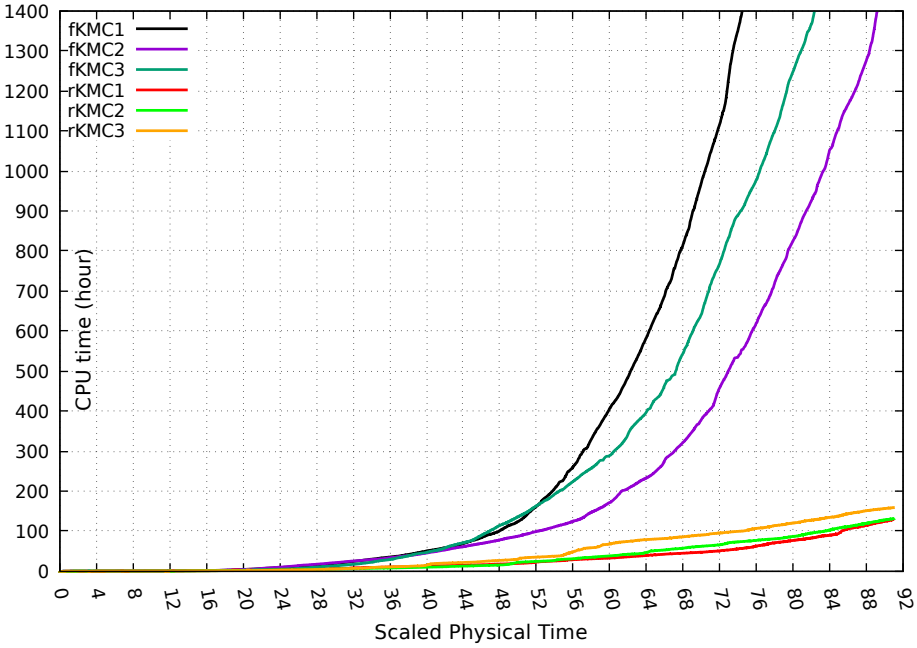



Figure 4: This graph shows the time needed by one CPU to perform both rejection and rejection-free (full) OLKMC schemes. The data plotted here is for the same realizations in Figure 3. When the system size is about fifty-five particles, the ratio of the black curve to the red and green curves is about ten and when it is about sixty-five particles, the ratio is about thirty.

# 5 Conclusion

In this paper we have demonstrated the viability and potential for using a rejection scheme to accelerate OLKMC. The main idea is to partition the set of rates that must be found in a way that allows for a local search procedure. If this can be achieved, one can expect an $O(N_p)$ boost in performance. In this paper, the partitioning is accomplished by identifying moves with the atom that moves the furthest in the transition, and rate estimates rely on a notion of bond-counting similar to what one finds in the lattice-based KMC literature. There remain many unexplored variations on both of these approaches. For example, one could partition moves based on energy changes rather than distance moved, and rate estimates could incorporate prior information similar to the way saddle-point reconvergence/recycling has been used in other OLKMC work,[2,5,28] especially when the size of the system is large.

# Acknowledgements

The authors acknowledge support from NSF grant number 1613729. We also thank Dmitri Schevarchov for many helpful discussions regarding this work.

# References

(1) Kelly, A.; Knowles, K. M. *Crystallography and Crystal Defects: Second Edition*; Wiley: Chichester, West Sussex, UK, 2012; Chapters 8, 9 & 10, pp 241-333.

(2) Henkelman, G.; Jónsson, H. Long Time Scale Kinetic Monte Carlo Simulations without Lattice Approximation and Predefined Event Table. *J. Chem. Phys.* **2001**, *115* (21), 9657-9666.

(3) Kara, A.; Trushin, O.; Yildirim, H.; Rahman, T. S. Off-Lattice Self-Learning Kinetic

Monte Carlo: Application to 2D Cluster Diffusion on the fcc(111) Surface. *J. Phys. Condens. Matter* **2009**, *21* (8), 84213.

(4) Konwar, D.; Bhute, V. J.; Chatterjee, A. An Off-Lattice, Self-Learning Kinetic Monte Carlo Method Using Local Environments. *J. Chem. Phys.* **2011**, *135* (17), 174103.

(5) El-Mellouhi, F.; Mousseau, N.; Lewis, L. J. Kinetic Activation-Relaxation Technique: An off-Lattice Self-Learning Kinetic Monte Carlo Algorithm. *Phys. Rev. B - Condens. Matter Mater. Phys.* **2008**, *78* (15), 153202.

(6) Lam, C.-H.; Lee, C.-K.; Sander, L. Competing Roughening Mechanisms in Strained Heteroepitaxy: A Fast Kinetic Monte Carlo Study. *Phys. Rev. Lett.* **2002**, *89* (21), 216102.

(7) Lung, M. T.; Lam, C. H.; Sander, L. M. Island, Pit, and Groove Formation in Strained Heteroepitaxy. *Phys. Rev. Lett.* **2005**, *95* (8), 86102.

(8) Russo, G.; Smereka, P. Computation of Strained Epitaxial Growth in Three Dimensions by Kinetic Monte Carlo. *J. Comput. Phys.* **2006**, *214* (2), 809-828.

(9) Russo, G.; Smereka, P. A Multigrid-Fourier Method for the Computation of Elastic Fields with Application to Heteroepitaxy. *Multiscale Model.* **2006**, *5* (1), 130-148.

(10) Schulze, T. P.; Smereka, P. An Energy Localization Principle and Its Application to Fast Kinetic Monte Carlo Simulation of Heteroepitaxial Growth. *J. Mech. Phys. Solids* **2009**, *57* (3), 521-538.

(11) Schulze, T. P.; Smereka, P. Simulation of Three-Dimensional Strained Heteroepitaxial Growth Using Kinetic Monte Carlo. *Commun. Comput. Phys.* **2011**, *10* (5), 1089-1112.

(12) Schulze, T. P.; Smereka, P. Kinetic Monte Carlo Simulation of Heteroepitaxial Growth: Wetting Layers, Quantum Dots, Capping, and Nanorings. *Phys. Rev. B* **2012**, *86* (23), 235313.

(13) Boateng, H. A.; Schulze, T. P.; Smereka, P. Approximating Off-Lattice Kinetic Monte Carlo. *Multiscale Model. Simul.* **2014**, *12* (1), 181-199.

(14) Eyring, H. The Activated Complex in Chemical Reactions. *J. Chem. Phys.* **1935**, *3*, 107-114.

(15) Truhlar, D. G.; Garrett, B. C.; Klippenstein, S. J. Current Status of Transition-State Theory. *J. Phys. Chem.* **1996**, *100* (31), 12771-12800.

(16) Vineyard, G. H. Frequency Factors and Isotope Effects in Solid State Rate Processes. *J. Phys. Chem. Solids* **1957**, *3* (1-2), 121-127.

(17) Voter, A. F.; Doll, J. D. Dynamical Corrections to Transition State Theory for Multistate Systems: Surface Self-Diffusion in the Rare-Event Regime. *J. Chem. Phys.* **1985**, *82* (1), 80-92.

(18) Hänggi, P.; Talkner, P.; Borkovec, M. Reaction-Rate Theory: Fifty Years after Kramers. *Rev. Mod. Phys.* **1990**, *62* (2), 251-341.

(19) Henkelman, G.; Jónsson, H. A Dimer Method for Finding Saddle Points on High Dimensional Potential Surfaces Using Only First Derivatives. *J. Chem. Phys.* **1999**, *111* (15), 7010-7022.

(20) Olsen, R. A.; Kroes, G. J.; Henkelman, G.; Arnaldsson, A.; Jónsson, H. Comparison of Methods for Finding Saddle Points without Knowledge of the Final States. *J. Chem. Phys.* **2004**, *121* (20), 9776-9792.

(21) Heyden, A.; Bell, A. T.; Keil, F. J. Efficient Methods for Finding Transition States in Chemical Reactions: Comparison of Improved Dimer Method and Partitioned Rational Function Optimization Method. *J. Chem. Phys.* **2005**, *123* (22), 224101.

(22) Kästner, J.; Sherwood, P. Superlinearly Converging Dimer Method for Transition State Search. *J. Chem. Phys.* **2008**, *128* (1), 14106.

(23) Mitchell, I.; Irle, S.; Page, A. J. A Global Reaction Route Mapping-Based Kinetic Monte Carlo Algorithm. *J. Chem. Phys.* **2016**, *145* (2), 24105.

(24) Miron, R. A.; Fichthorn, K. A. Multiple-Time Scale Accelerated Molecular Dynamics: Addressing the Small-Barrier Problem. *Phys. Rev. Lett.* **2004**, *93* (12), 128301.

(25) Lorentz, H. A. Ueber Die Anwendung Des Satzes Vom Virial in Der Kinetischen Theorie Der Gase. *Ann. Phys.* **1881**, *248* (1), 127-136.

(26) Berthelot, D. Sur le Mélange des Gaz. *C. R. Hebd. Séances Acad. Sci.* **1898**, *126*, 1703-1855.

(27) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Oxford University Press: New York, USA, 1987; Chapter 1, pp 21.

(28) Xu, L.; Henkelman, G. Adaptive Kinetic Monte Carlo for First-Principles Accelerated Dynamics. *J. Chem. Phys.* **2008**, *129* (11), 114104.

# Graphical TOC Entry