

Database of atomistic reaction mechanisms with application to kinetic Monte Carlo

Rye Terrell, Matthew Welborn, Samuel T. Chill, and Graeme Henkelman^{a)}

Department of Chemistry and Biochemistry and the Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas 78712-0165, USA

(Received 16 May 2012; accepted 8 June 2012; published online 3 July 2012)

Kinetic Monte Carlo is a method used to model the state-to-state kinetics of atomic systems when all reaction mechanisms and rates are known *a priori*. Adaptive versions of this algorithm use saddle searches from each visited state so that unexpected and complex reaction mechanisms can also be included. Here, we describe how calculated reaction mechanisms can be stored concisely in a kinetic database and subsequently reused to reduce the computational cost of such simulations. As all accessible reaction mechanisms available in a system are contained in the database, the cost of the adaptive algorithm is reduced towards that of standard kinetic Monte Carlo. © 2012 American Institute of Physics. [<http://dx.doi.org/10.1063/1.4730746>]

I. INTRODUCTION

The interesting kinetics of many chemical and material systems are governed by rare events. Chemical reactions and diffusion in solids, for example, typically occur on time scales of milliseconds or longer. Standard molecular dynamics algorithms are limited by the femtosecond time scale of atomic vibrations and are not suitable for directly modeling rare events.

A. Kinetic Monte Carlo

Fortunately, in many rare event systems, there is a natural separation of time scales between fast vibrations within stable states and slow kinetics between states. If the states can be characterized, as well as the transition times between them, the kinetic Monte Carlo (KMC) algorithm is suitable to model the state-to-state kinetics of the system.^{1,2} In the case of Markovian dynamics, the transition times between two adjacent states are characterized by a rate constant. For each state visited along a KMC trajectory, an exit process i is chosen stochastically with a probability, $\Gamma_i/\Gamma_{\text{tot}}$, proportional to its rate, Γ_i , where $\Gamma_{\text{tot}} = \sum \Gamma_j$ is the escape rate to any product state. The simulation time is incremented after each transition by $\Delta t = -\ln(\mu)/\Gamma_{\text{tot}}$, where μ is a random number distributed uniformly on (0, 1].

In order to calculate a KMC trajectory, the mechanism and rate of every processes that might occur during the course of the simulation must be known *a priori*. For this reason, the applicability of KMC is limited to the simplest of chemical and material systems.

B. Adaptive kinetic Monte Carlo

The adaptive kinetic Monte Carlo (AKMC) algorithm relaxes the requirement to know all reactive events *a priori* by determining the exit processes and rate for each state that is

visited during the the simulation.³ Using a min-mode following algorithm, such as the dimer method,⁴ saddle points are found along minimum energy pathways exiting the current state. The rate for leaving the current state can then be calculated with harmonic transition state theory.^{5,6} Similar approaches have been used in the hybrid eigenvector-following method⁷ to evaluate rates in reaction networks⁸ and, more recently, in the kinetic activation-relaxation technique.⁹ The AKMC method has also been referred to as on-the-fly and off-lattice KMC.¹⁰

For an accurate AKMC simulation, one needs to be confident that all energetically relevant processes leading out of each state visited are included in the rate table for that state. A statistical confidence parameter has been defined,

$$C = 1 - 1/(\alpha N_r), \quad (1)$$

which is a function of the number of sequential searches N_r that find previously discovered processes that are also energetically relevant. The constant α is the relative probability of finding the least likely process as compared to the most likely.¹¹ The criterion for a process to be energetically relevant is for it to have a barrier within a specified energy window of the lowest barrier process. Only these processes are relevant because the chance of selecting a process with a barrier of 30 kT higher than the lowest barrier process is negligible (ca. e^{-30}). To have confidence in an AKMC simulation, saddle searches are performed in each new state visited until C reaches a specified value, at which point a KMC step is taken to the next state.

The advantage of AKMC over KMC is that reaction mechanisms can be revealed during the simulation. The disadvantage is that AKMC requires more computational resources to search for processes in each new state visited. The computational cost of the saddle searches is particularly evident when energies and forces are evaluated with an electronic structure method, such as density functional theory (DFT). The number and cost of these saddle searches can be significantly reduced—particularly in the case of large

^{a)}henkelman@mail.utexas.edu.

systems—using the method of saddle point recycling.¹¹ Since the method of saddle point recycling is related to that of the kinetic database, it is briefly reviewed here.

C. Saddle point recycling

Suppose that from a reactant state R , the set of saddles and their corresponding products, S_i and P_i , have been determined. Saddles are usually found by making local random displacements from R and converging to nearby saddle points with a min-mode following algorithm. Assuming that a specified confidence in the rate table is reached, one of these processes, j , is selected by the KMC algorithm and the system is moved to the corresponding product state, P_j . It is now possible to check if any of the processes that were available to the system in the original state R are still available to the system in the new state $R' = P_j$. Suggestions of new saddle point geometries can be made by applying the vector between R and each connecting saddle P_i to the current state,

$$S'_i = R' + (S_i - R). \quad (2)$$

Another approach, used in Ref. 11, is to set the position of any atom in S'_i to the saddle geometry S_i if it moved significantly between R and R' and in the position of R' otherwise. In either scheme, the min-mode following algorithm is used to re-converge the recycled saddle point approximations, S'_i . Processes which are unaffected by motions of atoms in the selected process j will converge rapidly in the new state, and usually cost far fewer force evaluations than a typical random search. When a conflict occurs, a recycled saddle geometry, S'_i , may not converge, and the process is discarded. Recycling can make a qualitative difference for the cost of AKMC simulations. In order to reach confidence in state R' , only saddle searches with initial displacements in the neighborhood of atoms that moved significantly from state R to R' are required. If this region is local, the overall computational cost—measured in terms of the number of force evaluations—does not increase with the total system size.¹¹

Recycling is a fast, simple method that works well in many situations, but it has two limitations. First, recycling cannot make reliable suggestions for the region around the atoms that moved significantly from state R to state R' . Second, processes are only recycled from the previous state of the simulation; the method cannot take advantage of process information acquired at any earlier state. An illustration of these limitations can be seen in the example of a heptamer island diffusing on a (111) surface (see Fig. 1). The interaction potential is taken to be of the Morse double exponential form, using parameters to match bulk Pt.¹² At room temperature, the heptamer diffuses by sliding as a unit in one of three directions from FCC to HCP hollow sites. After the slide occurs, it is energetically unfavorable to slide again in the same direction because the atoms in the island would move from (favorable) hollow sites to (unfavorable) top sites. The recycling algorithm then fails to predict saddles for two reasons: all processes available in the previous state contain the same set of moving atoms as the selected process, and cannot occur in the current state because of the symmetry of the substrate.

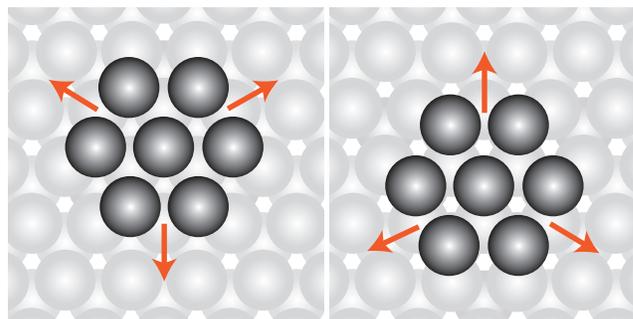


FIG. 1. A heptamer island slides on a (111) surface. At room temperature, the island slides as a unit, but not in the same direction consecutively.

D. Kinetic database

The kinetic database (KDB) is developed as a generalization of the process recycling method that does not suffer from the two problems mentioned. In our implementation, processes are added to the KDB using a minimal representation that includes only moving atoms and their immediate environment. The KDB can be queried with a new configuration to provide suggestions of available saddle point geometries. These suggestions are used to speed up AKMC simulations by reducing the number of random searches needed to reach confidence. Further, if a KDB is sufficiently populated and trusted, it can be used to provide processes and rates for a KMC simulation with no need to perform random searches. In this way, when the kinetic events of the system are known, we can reduce the computational cost of AKMC towards that of KMC. Finally, the KDB is very general so that it can be used to suggest reaction mechanisms from any interesting configuration under investigation, by hand, or as part of an algorithm for sampling or modeling the kinetics of atomic motion.

E. Previous work

The KDB is not the first method to store configurational and kinetic information for use in a KMC simulation; there have been a number of such efforts along these lines. One approach, called self-learning KMC (SL-KMC),¹³ uses a two-dimensional occupancy matrix to define environments in the neighborhood of diffusing atoms on a crystal surface. When an atom is found to be in a known environment, the possible diffusion events can then be looked up in a stored table. This method can be valuable for short-ranged interactions between species on a two-dimensional lattice, but as the interaction length increases (even to second neighbors), the number of possible environments makes the approach intractable. A recent extension to three dimensions has a similar scaling with interaction range.¹⁴

Another interesting idea, put forward by El-Mellouhi *et al.*,⁹ is to store the local environments around active atoms in terms of their bonding topology. An algorithm called NAUTY (Ref. 15) is used to generate a hash code of the bonding graph, allowing for a very efficient comparison of different topologies by simple string matching. It is the binary definition of a bond, based upon the distance between neighbors, which enables this concise description of the

environment around an atom. The network picture of bonding is appropriate for a crystalline solid, such as Si, but it has limitations when there is a continuous distribution of bond lengths between atoms, such as in disordered materials. Then, many topologies are required to accurately describe the range of events available to the system. Furthermore, if a sufficient number of neighbors are included in the bonding topology to accurately define the environment of an atom, the number of topologies continues to grow in the course of a simulation, and there is a diminishing chance of revisiting topologies.¹⁶

Recently, another approach called local environment KMC (LE-KMC) was reported.¹⁷ In LE-KMC, local environments are stored as a set of positions between a central atom and its neighbors, up to a radial cutoff. As in SL-KMC, an environment is matched to a query configuration when all neighbors are present at the specified relative position. Off-lattice atoms are allowed in LE-KMC by using a histogram of distances, instead of only those which correspond to fixed lattice sites. The limitations of the two methods, however, are similar. In both cases, a large environment region, which provides the most accurate rates, also leads to an exponentially large number of possible environments. LE-KMC was shown to be effective for modeling diffusion on a crystalline surface, but it remains to be seen if it can work in amorphous materials, or even when a significant number of defects are present.

The KDB presented here has many similarities to existing methods. The goals are the same: to store kinetic events calculated previously so they can be used in current simulations. The method details, however, are different both in philosophy and implementation. As described next, our strategy is to store a minimal representation of a kinetic event so that the number of events in the database remains tractable. Allowing for the rotation and translation of configurations in our database when matching a query configuration further minimizes duplicate entries. Our aim is not to quantify an environment well-enough for the database to provide a reliable energy barrier or rate. Rather, the KDB provides only estimates of reactant, saddle, and product geometries, which are consistent with a query configuration. A subsequent optimization step is necessary to verify that the process is valid and to determine the activation barrier. So, while the KDB can provide reaction mechanisms for a KMC simulation, it does not replace the calculation of the rate table. Finally, it is designed to be quite general so that it can be used for the determination of reaction mechanisms outside of KMC simulations.

II. METHOD

A. Populating the kinetic database

When a new reaction mechanism is found, such as in the course of an AKMC simulation, the process is described by the atomic configurations of the reactant, saddle, and product states. The subset of atoms in these configurations which are local to the process are used to populate the KDB. Atoms which move significantly during the process, as well as their neighbors, are considered local to the process. More specifically, atom i is determined to have moved significantly if

$$\max(|r_i - s_i|, |s_i - p_i|, |r_i - p_i|) \geq d_d, \quad (3)$$

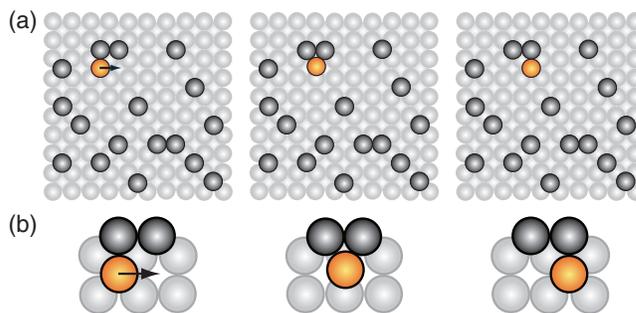


FIG. 2. (a) A crystal surface (light grey) with adatoms (dark grey) in which one adatom (orange) undergoes a hopping process and is identified as a moving atom. This adatom and its neighbors are determined to be local to the process. (b) The coordinates of the local atoms are extracted and stored in the database as reactant, saddle, and product configurations.

where r_i , s_i , and p_i are the positions of atom i in the reactant R , saddle S , and product P , configurations, respectively, and d_d is a cutoff distance. In test cases of adatom diffusion, 0.7 \AA was found to be an effective value. Atom j is determined to be a neighbor of atom i if

$$\min(r_{ij}, s_{ij}, p_{ij}) < d_n, \quad (4)$$

where $r_{ij} = |r_i - r_j|$ and the neighbor bond length $d_n = \sigma(c_i + c_j)$ is taken to be the sum of the covalent radii c_i and c_j for the two atoms multiplied by a scaling factor σ .¹⁸ Our value of $\sigma = 1.2$ allows for a neighbor to be 20% beyond the covalent bond length.

The atoms which are local to the process are inserted into the database in the reactant, saddle, and product states. An example of this extraction is illustrated in Fig. 2(a). The hopping adatom is determined to have moved significantly by Eq. (3), and its neighboring atoms are determined by Eq. (4). The minimal representation of the process, which is inserted into the database, is shown in Fig. 2(b).

B. Querying the kinetic database

The KDB is typically queried with a minimum energy configuration (query configuration) for which saddle point suggestions are desired. In broad strokes, the query algorithm searches the query configuration for any local configuration of atoms that match the reactant or product configuration of any process in the KDB. Once a list of candidate processes has been determined, the saddle points in the KDB are mapped to the query configuration to generate saddle point suggestions.

Specifically, the query algorithm first selects those KDB entries that contain at least as many atoms, by type, as the query configuration. For each candidate KDB entry, a set of mappings are generated between atoms from the KDB configuration to atoms in the query configuration. Each mapping is a transformation of the KDB configuration to a location in the query configuration where the local geometry closely resembles that of the KDB configuration.

The first step of the mapping procedure identifies atoms in the query configuration with the same number and types of neighbors as the mobile atoms in the KDB process. This is illustrated in Fig. 3, where the query configuration is a

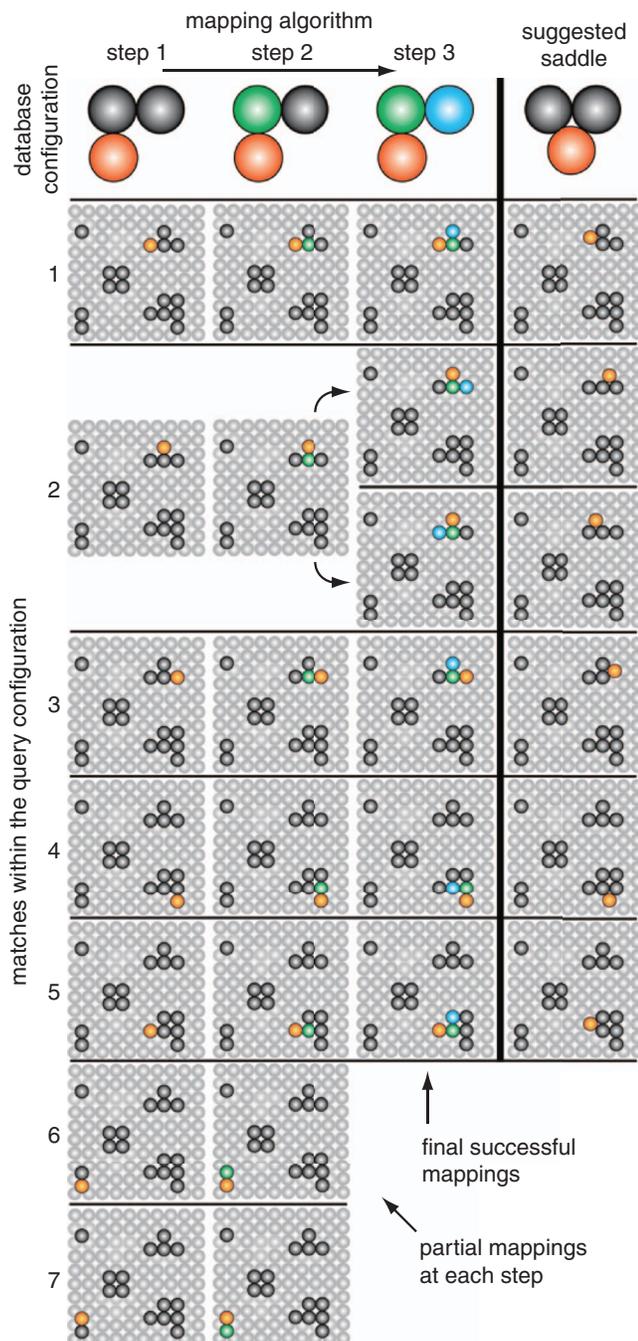


FIG. 3. The mapping algorithm identifies locations within the query configuration that match the geometry of a minimum configuration in the database. The algorithm generates a list of mappings from the database configuration atoms to their counterparts in the query configuration. The resulting mappings are used to suggest saddle points for the query configuration (final column).

crystal substrate with supported adatom clusters. For simplicity, only the adatoms are considered in the matching algorithm; an extension to include the substrate atoms does not change the nature of the algorithm. The top row in Fig. 3 shows the reactant configuration of the KDB entry under consideration (the database configuration). The entire KDB entry is shown in Fig. 2. In each step of the query algorithm, a different atom from the KDB configuration is matched to an atom in the query configuration. In Fig. 3, the atoms are

matched in steps 1–3, in the sequence red, green, and then blue.

In the first step, the (red) mobile atom is selected from the KDB configuration and is mapped to atoms in the query configuration that share the same number and type of neighbors (one adatom neighbor, in this case). All seven matching atoms are colored red in the seven rows of matching configurations.

After the initial set of mappings have been found in step 1, the algorithm selects a second atom from the unmapped set of atoms in the KDB configuration (step 2, first row, green atom) and calculates the distance from that atom to all mapped atoms in the KDB configuration (at this step, only the red atom). The algorithm iterates over partial mappings found in step 1 and finds atoms in the query configuration where the distance from that atom and the already-mapped atoms for that mapping match the distances determined for the KDB configuration. The distances are considered to match when

$$|r_{ij}^{(k)} - r_{ij}^{(q)}| < d_c, \quad (5)$$

where r_{ij} is the distance between the selected (unmapped) atom i and the mapped atoms j , in the KDB (k) and query (q) configurations. The cutoff distance d_c is taken to be 0.3 \AA , which we have found to be a robust value for a variety of solid state systems.

Matching atoms are used to generate a new set of possible mappings from the previous step. This process is illustrated in the second column of Fig. 3 (step 2), where the green atom of the KDB configuration has been mapped to the green atoms of the query configuration. In this case, all mappings at step 1 result in a single mapping at step 2, but in general, there can be more or fewer mappings in each subsequent step.

The matching procedure is repeated until all atoms of the KDB configuration have been mapped to a corresponding atom within the query configuration. In Fig. 3, the procedure is complete in step 3 of the algorithm, where the third and final atom of the KDB configuration (blue) has been mapped to atoms (blue) in the query configuration. Note that in step 2, a single partial mapping has branched into two in the final step, and that the mappings in rows 6 and 7 have been rejected because no atom was found whose distances to the mapped atoms matched the distances found for the third (blue) KDB configuration atom.

After a set of mappings from the atoms in the KDB minimum energy configuration to atoms in the query configuration are found, an affine transformation $\Phi(R) = R'$ is used so that the transformed position of atom i is

$$r'_i = \Phi_{\text{rot}} r_i + \Phi_{\text{trans}}, \quad (6)$$

where Φ_{rot} is a 3×3 rotation matrix and Φ_{trans} is a translation vector, chosen to minimize

$$\sum_i^{N^{(k)}} |r'_i - r_i^{(q)}|^2. \quad (7)$$

Here, r'_i and $r_i^{(q)}$ are the positions of atom i in the transformed KDB and query configurations, and $N^{(k)}$ is the number of atoms in the KDB configuration. There are a number of algorithms to determine the optimal rotation to minimize the root-mean-square deviation (RMSD) between the

mapped atoms. While the Kabsch algorithm¹⁹ is a common choice, we used a quaternion-based approach that avoids rotoinversions.²⁰

We apply the transformation to the KDB configuration and calculate a score for the match,

$$\max |r_i^{(k)} - r_i^{(q)}|, \quad (8)$$

which is the greatest distance between any two mapped atoms in the KDB and query configurations. If this distance is less than a desired tolerance, the saddle configuration of the KDB entry S is used to make a saddle point suggestion by applying Φ to S . The saddle suggestions for the query configuration in Fig. 3 are shown in the final column.

III. RESULTS

A. Performance of the KDB

The accuracy and performance of the KDB was compared to the recycling algorithm using the model system of Pt heptamer island formation and diffusion on a Pt(111) surface.¹² The AKMC algorithm without recycling or the KDB was applied to the starting configuration A in Fig. 4(a). For this simulation, we used a confidence parameter of 0.99, which corresponds to a confidence requirement of 100 random saddle searches without finding a new saddle (assuming unity α in Eq. (1)). In the initial configuration (A), adatoms are scattered randomly on the surface. Over the course of the simulation the atoms form clusters (B), then a compact island (D), and finally the stable heptamer (E) after $0.5 \mu\text{s}$.

A comparison of saddle point recycling and the KDB algorithm with standard AKMC are shown in Fig. 4(b). The number of saddle searches required to reach confidence in each state with the standard AKMC algorithm is given by the green line.

The simulation was then run again with the recycling algorithm. The same confidence parameter and state-to-state trajectory was imposed upon the simulation to ensure a fair comparison to standard AKMC. The red lines show that the recycling algorithm successfully predicts saddles and reduces the average number of saddle searches required to reach confidence. Note that as the island forms, the number of relevant processes that can occur outside of the region of the most recent process tends to decrease and the fraction of saddles found with the recycling algorithm along with it.

The performance of the KDB was measured in the same way (blue lines). The KDB predicts, on average, a greater percentage of saddles than the recycling algorithm. The advantage of the KDB becomes obvious as the simulation progresses, in spite of the increasing proximity of processes. After the heptamer island forms, it diffuses via sliding along the surface. After this process is stored in the KDB, it performs perfectly in the diffusive part of the trajectory, where the recycling algorithm performs poorly.

B. Bridging the gap to KMC

If the KDB is sufficiently populated, it is possible to run an AKMC simulation using only the suggestions from the

database, supplemented by no additional saddle searches. In this simulation mode, for each new state reached, the KDB is queried for suggestions, those suggestions are then refined and, if valid, are used to populate the rate table for that state.

To test the efficacy of this technique, we created 128 AKMC simulation trajectories of Pt island ripening, as described in the previous section. Each simulation was started with isolated adatoms and run until they coalesced into the compact Pt heptamer island. The KDB was populated using the information from two of these trajectories. Then, 128 *new* AKMC trajectories were simulated using only the saddle suggestions from the KDB. The center panel of Fig. 5 is a plot of the elapsed time as a function of the AKMC step for each of the standard AKMC simulations (red) and the KDB-only simulations (blue). Most trajectories follow a path of fast motion of surface atoms (e.g., bottom left panel) coalescing into a asymmetrical heptamer island and then undergoing some slow processes (e.g., middle left panel) rearranging into the symmetrical heptamer island. The primary line in the figure corresponds to the fast diffusion processes, while the vertical lines are the slow island rearrangement processes. The odds of reaching the symmetric island through the diffusion of single atoms and smaller islands is low as compared to forming an asymmetric island that must rearrange to form the lowest energy compact structure, which can be seen by the small fraction of trajectories that end on the primary line.

The AKMC simulations found the compact island on time scales between 10^{-11} to 10^{-4} s; the KDB-only approach found a similar distribution. There was, however, one

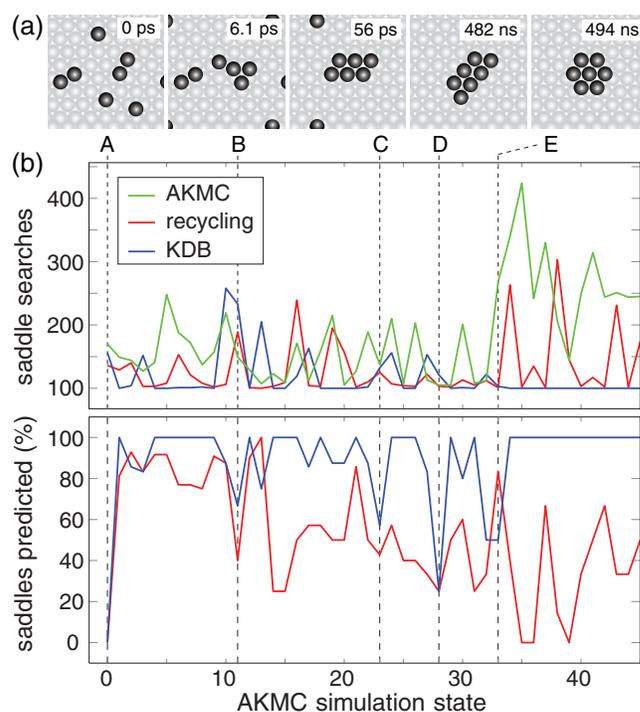


FIG. 4. (a) AKMC trajectory of a heptamer island formation process on a crystal surface. (b) At least 100 saddle searches are required (N_r in Eq. (1)) to reach the convergence criterion $C = 0.99$; more are required if new saddles are discovered. The method of saddle point recycling reduces this number, and the KDB is even more effective, particularly after a compact island forms (state E) and slides diffusively by a known mechanism.

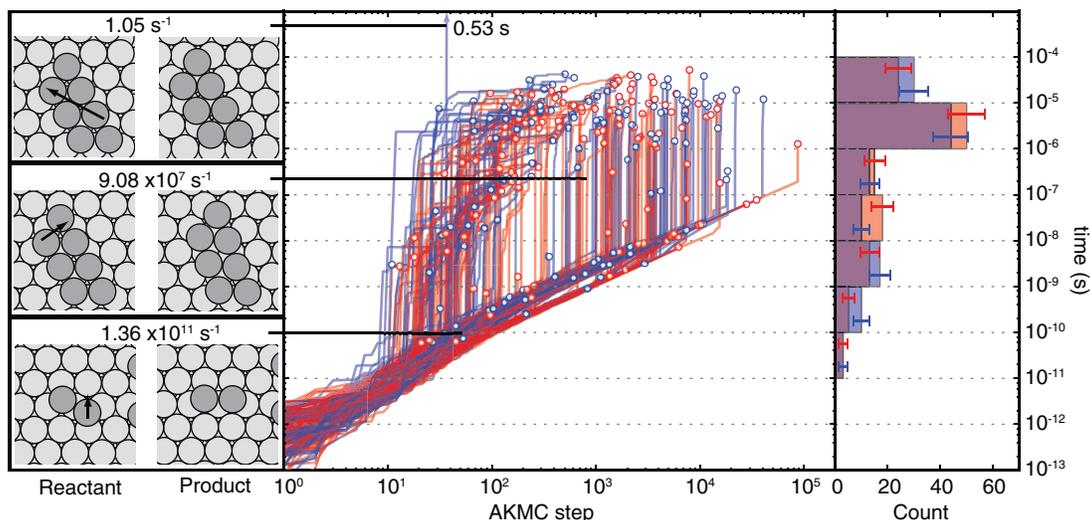


FIG. 5. In the center panel, the simulation time is shown over the course of 128 AKMC (red) and AKMC+KDB (blue) simulations. Each simulation is run until the lowest energy compact heptamer island, shown in Fig. 1, is reached. The KDB was populated with all processes discovered by only two of the AKMC simulations. The AKMC+KDB simulations were then run using only saddle points suggested from the KDB, with no additional saddle searches. The comparable distribution of the island formation times is shown in the right panel. On the left, three events are shown. The lower panel shows a typical single-atom fast diffusion process, which forms the primary initial path of trajectories. The middle is an example of a slower concerted mechanism involving a pivot of the cluster between hcp and fcc hollow sites. On top is a slow concerted sliding mechanism, which was followed in an AKMC+KDB; an unlikely event caused by the incomplete population of the KDB. The middle mechanism, which is much faster, can occur from the same state. Had that trajectory also been used to populate the KDB, the slow mechanism would almost certainly have been avoided.

outlier in the KDB-only distribution with an island formation time of 0.53 s. This outlier is due to the simulation trajectory reaching a local configuration for which there was incomplete information in the KDB, and subsequently selecting the low rate process shown in the top left panel of Fig. 5. The KDB-only trajectory failed (in terms of accuracy) in this case because neither of the trajectories that were used to populate the KDB passed through a state that had a process like the one shown in the middle left panel, with a much faster rate. The accuracy of simulations that use the KDB-only approach is dependent on how complete the KDB is with respect to the trajectory of the simulation. In the limit of a complete database, the KDB distribution of island formation times is indistinguishable from that of the AKMC simulations.

The computational cost for the 128 KDB-only simulations with two seed trajectories was 2.1% that of the original 128 simulations, measured in terms of number of force evaluations including the force evaluations from the two AKMC simulations used to populate the KDB. The cost of each KDB-only trajectory was, on average, 1.4% the cost of a standard AKMC trajectory.

C. Application to an off-lattice DFT system

To demonstrate the applicability of the KDB algorithm on a somewhat more complex system, we modeled the kinetics of a bulk Si lattice at 500 K with two B atoms initially sharing one interstitial site, a so-called B₂I cluster. Boron is commonly used as a dopant for p-type Si. Due to the scaling relations of semiconductor devices, smaller devices require a higher concentration of dopants. At very high con-

centrations, the dopant atoms can coalesce into inactive clusters. The mechanism of B-cluster break-up and diffusion is important for understanding the fabrication of nanoscale Si devices.²¹

In our simulation of B₂I, the energy of the system and the forces on the atoms were calculated with DFT using the Vienna *ab initio* simulation package.²² For our calculations, we used the PW-91 exchange-correlation functional,²³ a plane wave cutoff of 200 eV, and a $2 \times 2 \times 2$ Monkhorst-Pack k-point mesh.²⁴ States at the Fermi level were smeared by a Gaussian width of 0.05 eV. A Si supercell, initially containing 64 Si atoms, was relaxed in a cubic box of side length 10.914 Å. One Si atom was removed and replaced with a pair of B atoms; the geometry was relaxed to create the initial state. Geometries were considered converged when the force on all atoms dropped below 0.01 eV/Å.

For each simulation state, a confidence parameter of 0.95 was used. First, KDB-suggested saddles were converged, followed by the required number of random searches to reach confidence. The number of successful KDB predictions as a fraction of total saddles found in each state is shown in Fig. 6. In the initial state, the KDB has not yet learned anything about the system, so it predicts no saddles. State 2 is identical to the initial state by a rotation, and the KDB predicts all relevant saddles. State 10 has a local configuration that is unique among those seen so far, resulting in the identification of only the reverse process being identified by the KDB. In state 20, the B dimer is separated by a Si atom. This state also contains local configurations unique among those visited, so the KDB is unable to predict most of the available saddles. As the simulation progresses, however, the fraction of saddles predicted by the KDB is very high. Overall, the KDB dramatically improves computational efficiency, allowing for a 32-state

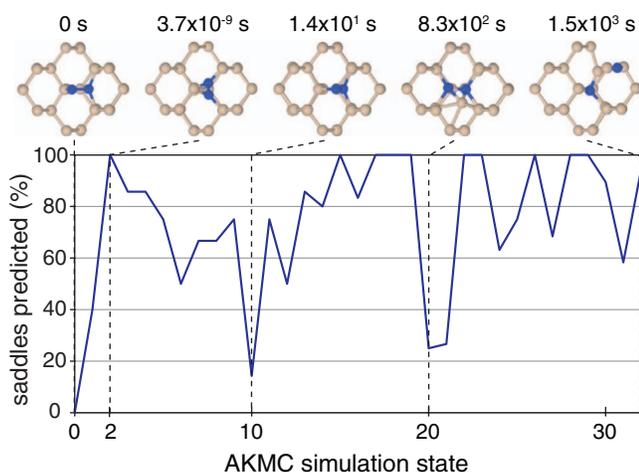


FIG. 6. The fraction of saddles successfully predicted with the kinetic database used during an AKMC simulation of bulk Si with two B atoms sharing a Si site, demonstrating the KDB's ability to deal with a complex, off-lattice system with energies and forces calculated with DFT.

simulation showing the breakup of the B_2I cluster on a time scale of an hour at 500 K.

IV. CONCLUSION

A database of kinetic events has been introduced and used to accelerate the AKMC algorithm by reducing the number of saddle searches needed to reach confidence in the rate table from most states. We have shown how a well-populated KDB can be used as the basis of a KMC simulation without the need for any additional saddle point searches, and therefore requiring only a fraction of the computational effort. The KDB is a very general tool for predicting available saddles to a query configuration; it can be used for off-lattice simulations and in combination with DFT. Software for the KDB is available at <http://theory.cm.utexas.edu/kdb>.

ACKNOWLEDGMENTS

The work was supported by the National Science Foundation (Award No. CHE-1152342). The authors are grateful for allocations of computing resources at the Texas Advanced Computing Center, and for donations of computer time from participants of the EON distributed computing project.

- ¹A. B. Bortz, M. H. Kalos, and J. L. Lebowitz, *J. Comput. Phys.* **17**, 10 (1975).
- ²D. T. Gillespie, *J. Comp. Phys.* **22**, 403 (1976).
- ³G. Henkelman and H. Jónsson, *J. Chem. Phys.* **115**, 9657 (2001).
- ⁴G. Henkelman and H. Jónsson, *J. Chem. Phys.* **111**, 7010 (1999).
- ⁵C. Wert and C. Zener, *Phys. Rev.* **76**, 1169 (1949).
- ⁶G. H. Vineyard, *J. Phys. Chem. Solids* **3**, 121 (1957).
- ⁷L. J. Munro and D. J. Wales, *Phys. Rev. B* **59**, 3969 (1999).
- ⁸D. J. Wales, *Mol. Phys.* **100**, 3285 (2002).
- ⁹F. El-Mellouhi, N. Mousseau, and L. J. Lewis, *Phys. Rev. B* **78**, 153202 (2008).
- ¹⁰A. F. Voter, F. Montalenti, and T. C. Germann, *Annu. Rev. Mater. Res.* **32**, 321 (2002).
- ¹¹L. Xu and G. Henkelman, *J. Chem. Phys.* **129**, 114104 (2008).
- ¹²G. Henkelman, G. Jóhannesson, and H. Jónsson, in *Progress on Theoretical Chemistry and Physics*, edited by S. Schwartz (Kluwer Academic, New York, 2000), pp. 269–299.
- ¹³O. S. Trushin, A. Karim, A. Kara, and T. S. Rahman, *Phys. Rev. B* **72**, 115401 (2005).
- ¹⁴G. Nandipati, A. Kara, S. I. Shah, and T. S. Rahman, *J. Comput. Phys.* **231**, 3548 (2012).
- ¹⁵B. D. McKay, *Congr. Numer.* **30**, 45–87 (1981).
- ¹⁶L. K. Béland, P. Brommer, F. El-Mellouhi, J.-F. Joly, and N. Mousseau, *Phys. Rev. E* **84**, 046704 (2011).
- ¹⁷D. Konwar, V. J. Bhute, and A. Chatterjee, *J. Chem. Phys.* **135**, 17103 (2011).
- ¹⁸B. Cordero, V. Gomez, A. E. Platero-Prats, M. Reves, J. Echeverria, E. Cremades, F. Barragan, and S. Alvarez, *Dalton Trans.* 2832 (2008).
- ¹⁹W. Kabsch, *Acta Crystallogr., Sect. A: Cryst. Phys., Diffr., Theor. Gen. Crystallogr.* **32**, 922 (1976).
- ²⁰B. K. P. Horn, *J. Opt. Soc. Am. A* **4**, 629 (1987).
- ²¹X.-Y. Liu and W. Windl, *J. Comput. Electron.* **32**, 203 (2005).
- ²²G. Kresse and J. Hafner, *Phys. Rev. B* **47**, R558 (1993).
- ²³J. P. Perdew, in *Electronic Structure of Solids*, edited by P. Ziesche, and H. Eschrig (Akademie Verlag, Berlin, 1991), pp. 11–20.
- ²⁴H. J. Monkhorst and J. D. Pack, *Phys. Rev. B* **13**, 5188 (1976).