A guide to the aKMC script developed in Henkelman group

Lijun Xu and Graeme Henkelman

Department of Chemistry and Biochemistry, University of Texas at Austin, Austin, Texas 78712-0165

(Dated: January 11, 2010)

I. OBTAINING AKMC SCRIPTS

The akmc script is part of the vtstscripts package written by Henkelman group in the University of Texas at Austin, designed for using Vienna ab initio simulation package (VASP). The whole package is written in Perl language and is expected to be working on various platforms. In WINDOWS operation system, ActivePerl program is a possible choice to run the scripts. The whole package can be downloaded via anonymous CVS system (check the guide on http://theory.cm.utexas.edu/henkelman/code/). You HAVE TO download the whole vtstscripts package in order to run adaptive KMC simulation on VASP, simply because scripts are actually calling each other. By the way, your VASP must be compiled with VTST-TOOLS developed by Henkelman group. The standard VASP package you purchased doesn't work for this aKMC script. The webpage about AKMC is "http://theory.cm.utexas.edu/vtsttools/akmc/".

II. WHAT DOES AKMC SCRIPT DO?

Given a starting system configuration, the akmc script handles the whole adaptive KMC simulation using energy and forces calculated from VASP. It first minimizes the configuration it receives, then it will search for saddle points for the minimized configuration and quench each found saddle point into two minima directly connected with it. It can also calculates normal mode frequencies for the initial minimum and its saddles. After finishing saddle point searches, it will calculate the rate constants and prefactors with harmonic transition state theory and get a table of rates for each reaction mechanism. Using that table, it will do a KMC move to select one mechanism and update the system configuration to the end point connected with the minimum energy pathway (or mechanism). The whole process is repeated for the new system configuration. The whole operations are handled by just typing the command "akmc.pl".

III. PREREQUISITE READINGS

Graeme Henkelman and Hannes Jónsson, Long time scale kinetic Monte Carlo simulations without lattice approximation and predefined event table, J. Chem. Phys. 115, 9657-9666 (2001) [the original idea of akmc]

Lijun Xu and Graeme Henkelman, Adaptive kinetic Monte Carlo for first-principles accelerated dynamics, Journal of Chemical Physics 129, 114104 (2008) [details about algorithms used in this package]

IV. OPTIONAL READINGS

Graeme Henkelman and Hannes Jónsson, J. Chem. Phys. 111, 7010 (1999) [The dimer method]

Lijun Xu, Donghai Mei and Graeme Henkelman, Studying chemical reactions with DFT-based adaptive kinetic Monte Carlo simulations - methanol full decomposition pathways on Cu(100) surface, in preparation [An example of using AKMC to study chemical reactions]

V. EXAMPLES

After reading this manual, readers are strongly recommended to run the example given on our group webpage (http://theory.cm.utexas.edu/vtsttools/akmc/) to see how the scripts and simulations work out.

VI. GET STARTED

A. Files used by aKMC script

Create a home directory for aKMC calculations. Put the following files in the aKMC home (or designated) directory, set all the parameters and start the run by typing "akmc.pl"

CONFIG It has all the control parameters running aKMC simulation. Details will be given in the following section.

POSCAR This is the initial configuration given to the program. It will be updated with new structures as simulation goes. The file format must be consistent with VASP POSCAR requirements.

KPOINTS Control the k-points setup for VASP calcualtion.

POTCAR Pseudopoential for VASP calculations.

INCAR_sp It instructs how to run a saddle point search in VASP. It is in the same format as VASP INCAR files.

INCAR_min It instructs how to run a geometry minimization in VASP. It is in the same format as VASP INCAR files.

INCAR_dynmat It instructs how to calculate a dynamic matriix in VASP. It is in the same format as VASP INCAR files.

DISPLACECAR This is the VASP DISPLACECAR file for dynmaic matrix calculation, showing how much each degree of freedom is finitely displaced.

DISPLACECAR_sp This file is in the same format as DISPLACECAR, but each value in the file is the Gaussian width with which a random displacement is to be created for that degree of freedom. It controls how the random saddle point searches are prepared.

akmc_submit.pl This is your own script that submits a job from the directory the job is supposed to run in and prints a job id as the only screen output. It has to be placed in the same folder as akmc.pl and other VTSTSCRIPTS. An example of this scripts is provided on the webpage.

akmc_check.pl This is your own script that takes a job id and checks the status. The only screen output should be "running", or, "stopped", or "queue". It has to be placed in the same folder as akmc.pl and other VTSTSCRIPTS. An example of this scripts is provided on the webpage.

akmc_kill.pl This is your own script that kills a job. It has to be placed in the same folder as akmc.pl and other VTSTSCRIPTS. An example of this scripts is provided on the webpage.

akmc_sp.sub This is the submission script used by akmc_submit.pl for submitting saddle point searching jobs to the queue system.

akmc_min.sub This is the submission script used by akmc_submit.pl for minimization jobs to the queue system.

akmc_dynmat.sub This is the submission script used by akmc_submit.pl for dynamic matrix calculation jobs to the queue system.

B. Files and folders created by aKMC script

After running aKMC, there will be new files and folders to be created automatically. In the following list, you will see them in the order of being created by the script.

jobs.dat This file contains all the job information: work dirctory, job id, job type, energy, force, curavture.

min This folder is created to run the very beginning minimization. The minimized structure will be used to start the search of saddle points. A minimized structure represents a state of the system.

akmc.dat This file keeps the information of each aKMC step: showing the configuration of the step, barrier, accumulated time, etc. It is also a key file for the script to figure out the stop point from the previous running of "akmc.pl".

st0001 After the initial minimization, this state folder is created to contain all the saddle point searching processes for the first state. There are subfolders and files inside this directory. We take the first one st0001 as the example to show the tree structure of the state folders.

st0001/POSCAR, etc. This is the POSCAR copied from the AKMC home directory as the configuration of the state "st0001". Other types of files in home directory such as POTCAR, INCAR_min,INCAR_sp, DISPLACECAR_sp and KPOINTS will also be copied into this state folder.

st0001/neighborlist.dat This neighbor list file, if created, is for random displacement. Details are in the reference.

st0001/st.dat This file is the key file for a state: it records how many saddle point searches are running in the sr0001, and the evaluation of each saddle searching process: good, bad or something else. Together with jobs.dat, it does the main control of the aKMC job running. Any manual handle should target towards "jobs.dat" and "st.dat".

st0001/pr0001 This folder is the first saddle point search for state "st0001". It has subfolders and files.

st0001/pr0002 This folder is the second saddle point search for state "st0001". It has subfolders and files.

st0001/pr0001/POSCAR POSCAR for a saddle point searching job in VASP. It is created by randomly displacing the state POSCAR.

st0001/pr0001/INCAR INCAR for a saddle point seaching job in VASP. Note: all three INCAR_* files are copied inside folder st0001 and pr0001. INCAR_sp will be copied into INCAR automatically when the job is being submitted. akmc_sp.sub will be copied into akmc.sub which is the standard submission file in the akmc_submit.pl script.

st0001/pr0001/POTCAR Same POTCAR file as in the home directory.

st0001/pr0001/KPOINTS Same KPOINTS file as in the home directory.

st0001/pr0001/inter A unfinished but stopped job are cleaned up into this inter, it will be renamed to inter1, inter2, as more come in, but this "inter" is the most recent one. It is same as what vfin.pl does in our normal VASP job cleanup.

st0001/pr0001/final A converged (saddle searching) job is cleaned up into this folder. It is same as what vfin.pl does in our normal VASP job cleanup. DO NOT rename this folder. Both "inter" and "final" are required standard folder names in this package.

st0001/pr0001/POSCAR_sp It is the saddle point configuration file. The CONTCAR (or CENTCAR) of a converged job will be copied into this file. DO NOT rename this file.

st0001/pr0001/dynmat If dynamic matrix calculation is required, this folder will be created to do the dynmat calculation. POSCAR_sp will be copied into dynmat/POSCAR, INCAR_dynmat wll be copied into dynmat/INCAR, etc.

st0001/pr0001/dynmat/final A converged dynmat job is cleaned up into this folder. It is same as what vfin.pl does in our normal VASP job cleanup.

st0001/pr0001/mins This folder contains jobs of quenching the converged saddle into two adjacent minima.

st0001/pr0001/mins/min1 This folder contains the minimization of one image along the MEP. A normal VASP minimization job.

st0001/pr0001/mins/min2 This folder contains the minimization of the other image along the MEP. A normal VASP minimization job.

st0001/pr0001/mins/min1/final A folder for converged minimization job. Same as before.

st0001/pr0001/mins/min2/final A folder for converged minimization job. Same as before.

st0001/jobs.dat Once st0001 is done, the jobs.dat in the aKMC home directory will be copied into the state folder st0001.

st0002 If st0001 is finished, KMC will make a move and update the system into a new state. Then st0002 will be created to start the saddle point searches for the new state. AaKMC continues like this until it reaches the maximum number of steps indicated in CONFIG file.

StEnergyFile This is to store the energy of each state the AKMC simulation has visited. It is for the sake of avoiding reading the st.dat when the code goes back to check if the new state has appeared somewhere before. Any state in this file must be a finished state.

VII.   FOUR KEY FILES: CONFIG, ST.DAT, JOBS.DAT AND AKMC.DAT

Those four files determine the main stream of the aKMC script.

A.   CONFIG

Here is an sample "CONFIG" file.

—————————————————-
MaxJobs = 12
RandSeed = 5
NumSearches = 10
AkmcSteps = 100000
SimR = 0.1
NumKT = 20
Temperature = 300.0
Ediffmax = 0.05
Rdiffmax = 0.04
SearchesAlgo = dimer
BarrierMAX = 10.0
JobFile = jobs.dat
StFile = st.dat
AkmcFile = akmc.dat
DynMatFile = freq.dat
Prefactor = 1.0e12
RateTableFile = RateTableFile.dat
OldRunDirFile = OldRunDirFile.dat
StEnergyFile = StEnergyFile.dat
GrabQuenched = 1
Population = 0.2
PrRecycleAll = 1
PrRecycle = 1
PrRecycleShift = 1
SurfaceRecShift = 1
ConvergenceAlgo = 1
equivalency = 1
DisplaceAlgo = 1
DisplaceRange = 3
NN_rcut = 2.6
MaxCoordNum = 8
UseKDB = 0 ———————————————————-

File "config" is in the aKMC home directory and contains the following flags.

MaxJobs The maximum number of jobs can be submitted through aKMC. This number must be smaller than the maximum allowed by the queue system. For example, if the queue allows you to have at most 8 jobs running and waiting in the queue and you use three jobs to do things other than AKMC, you will have to set MaxJobs to 8-3=5. You have to tell the scripts how many jobs are allowed. There is no way for the script to figure out the number by itself.

NumSearches The number of searches for convergency standard. If there are no new good saddle points found during that many straight successful searches, the searching will stop and an event table will be compiled for KMC. Default: 10. Details are in the reference.

AkmcSteps The total number of KMC steps. Use "1" for mechanism study. Default: 1

SimR The distance (in Å) between two images centered around the saddle point along the minimum energy path. Those two images are expected to roll into two end points of the MEP upon minimization. Default: 0.1

Temperature The temperature in kelvin. Default: 77

NumKT The energy gap above the lowest saddle point. Saddle points above the highest point are considered unimportant to the final rate table. The unit is kT and T is for temperature. Normally, this number is at least 10, which means 10 times thermal energy at the temperature. Default: 20

EdiffMax The maximum energy difference (in eV) between two configurations, below which geometry difference will be calculated. Default: 0.05

RdiffMax The maximum difference (in Å) between any two corresponding atoms from two configurations, below which two geometries will be considered as identical. In aKMC, two POSACR files are frequently compared to determine if they are for the same geometry. Default: 0.05

SearchAlgo The saddle point searching mechanisms. Right now, it can only be "dimer". "Lanzcos" may be added to the script in the future. Default: "dimer".

BarrierMax The maximum energy allowed before killing a search for the searching point energy being too high. Default: 10

JobFile The filename for jobs.dat in the aKMC home directory. Default:" jobs.dat".

StFile The file name for st.dat in the state folder (such as st0001). Default: "st.dat"

AkmcFile The file name for akmc.dat. Default: "akmc.dat"

DynMatFile The file containing normal mode frequencies. Keep the default name unless you modify the dymmatrix.pl script. Default: "freq.dat".

Prefactor The dynamic matrix calculation will be disabled if this flag is turned on. The positive value will be used for every process in the coming calculations. You have to keep this flag consistent throughout your whole aKMC calculations. Default: 0 (disabled)

RateTableFile The file for storing the rate table. Default: "RateTableFile.dat"

OldRunDirFile The file listing all the database folders. Default: "OldRunDirFile"

StEnergyFile The energy of each state's configuration is listed in here. Default: "StEnergyFile".

Population An obsolete convergence standard. It will be only used for ConvergenceAlgo=2 which we don't recommend. Default: 0

GrabQuenched If it is set to be 1, the code will check if the newly converged saddle has been quenched before (not limited to this state, including all the previous states and database). Default: 0

PrRecycleAll If it is set to be 1, the code will check if the new state is one of the end points in those mechanisms have been found in previous states and database. If yes, the code copies those processes and recover those saddles for the new state. Default: 1.

PrRecycle If it is set to be 1, any good saddle from the previous (or equivalent) state will be recycled and reconverged for the new state. Default: 0.

PrRecycleShift If it is set to be 1, a vector-shifting recycle will be used. Default: 0.

SurfaceRecShift If it is set to be 1, the system will be a surface science problem: adsorbates on substrate. In POSCAR, adsorbates will be listed as the last group. Any good saddle from the previous state will be reconverged by adjusting the adsorbate orientation. This is not a mature technique. We do not recommend it unless you are familiar with the code itself. Default: 0.

ConvergenceAlgo The key flag controlling the ending of the saddle point searches: "0" for a fixed number of saddle seacrhes; "1" for a dynamic standard (coupled with "NumKT" flag)

Equivalency If it is set to be 1, the system will test if the new sate configuration is actually equivalent to some previous states by checking the the one-to-one matching between atoms in two POSCAR files (or "the radial distribution function" of adsorbates if "SurfaceRecShift" is set to 1.). Saddle points from any equivalent state will be reshuffled based on the matching and used directly. Default: 0.

Screenoutput the file for keeping the screen output from running "akmc.pl". Default: "out.dat".

DisplaceAlgo The displacing algorithm used in creating random dimer configurations: "0" for selecting one of the most under-coordinated atoms as the displacement center; "1" for one of the choosing the less (¡= MaxCoordNum) coordinated atoms as the displacement center; "2" for dividing the surface into islands and doing the "0" option in islands; "3" for dividing the surface atoms into islands and doing the "1" option in islands; any other values will lead to a random atom to be selected as the displacement center. Note: only those atoms (or degrees of freedom) marked as non-zero in the DISPLACECAR_sp can be the candidate displacement center in any of those algorithms. Default: 1. Details are in the reference. It is important to be familiar with these algorithms.

DisplaceRange The radius around the displacement center within which all of those atoms will be displaced by a random vector. Default: 3.

NN_rcut The maximum pair distance for two nearest neighbors. Default: 2.6

MaxCoordNum The maximum coordination number used in DisplaceAlgo=1. Default: 8.

BoltzmanEqu The threshold for two states being in Boltzman equilibrium: if one MEP shared by two states are dominating at least that amount of percentage in each rate table. Default: 0.999999.

UseKDB This is about using database to prepare saddle point searches for the new state. If it is on, random searches will be done after using the initial guesses from databases. Default: 0

KDBcutoff This is a standard used in UseKDB to determine how similar two systems should be in order to exchange saddle point information with each other. Default: 0.1

B. ST.DAT

Here is an sample 'st.dat' file.
————————————————-

st0001 -376.931759
status running
dynmat done
NumSearchesLeft 10
numprst 12
process status quality barrier final/(min1) final/(min2)
pr0001 done good 1.997 mins/min1 -375.549
pr0002 done good 0.595 mins/min1 -376.521
pr0003 done good 2.463 mins/min1 -375.566
pr0004 done bad 1.809 -375.988 -375.339
pr0005 killed saddleenergytoohigh_search na na na
pr0006 done bad 3.126 -375.409 -376.264
pr0007 killed positive_curvature_search na na na
pr0008 done bad 0.795 -376.433 -376.931
pr0009 done repeat*pr0008 0.795 na na
pr0010 done repeat*pr0002 0.595 na na
pr0011 quench promising 2.531 na na
pr0012 search pending na na na

————————————————-

File "st.dat" is in the state folder (such as st0001) and contains the following information (Let's take st0001/st.dat as the example).

First line "st0001 -376.931759" for the relative address and energy of the state.

Second line "status running" for indicating the state is not done yet. "status done" is for a done state.

Third line "dynmat running" for indicating the dynamic matrix calculation for the state configuration has not finished yet. "dynmat done" means it is either finished or tuned off by setting the Prefactor flag in CONFIG file.

Fourth line "NumSearchesLeft 10" shows how many more good saddle searches without seeing a new saddle are left. The state will be taken as "done" if it is set to zero.

Fifth line "numprst 12" shows that there are 12 process folders for 12 saddle point searching jobs.

Sixth line It is an optional description line showing the meaning of each column in the following parts of the file: "process status quality barrier final/(min1(ev)) final/min2(ev)"

Next line: column one It is the folder name for the process: e.g., pr0001, pr0002. etc.

Next line: column two It is the status for the process: "done" means the searching (and dynmat if needed) and quenching are all finished. Otherwise, the status would be "search", or "quench", or "dynmat" for each type of operations. If the search is killed for some reason, "killed" will appear in this column and a reason follows it.

Next line: column three It is the quality of the saddle point: "good" means one and only one of the end points of the minimum energy path is "same" as the configuration of the state (i.e., same as the POSCAR in folder st000x); "bad" means neither of the end points is same as the state configuration; "promising" means the barrier of the saddle (relative to the initial state) is not larger than "BarrierMax" in CONFIG file and the saddle is being quenched to determine if it is good or not; "bad" can also mean other situations including negative narrier, barrier being too high, etc; "repeat*pr0003" means this saddle is same as the saddle point in pr0003 and pr0003 has been finished (or at least the quenching jobs have been finished); "pending" means the saddle is not converged yet.

Next line: column four It is the barrier of the saddle point in eV.

Next line: column five If the saddle is "good", this column will record the address of the final state of the minimum energy path (the initial state will be same as the state configuration), e.g., "mins/min1" means that the final state is in "st0001/pr0001/mins/min1/POSCAR" and implies that the initial state is "st0001/pr0001/mins/min2/POSCAR" which is same as "st0001/POSCAR". If the saddle is "bad", this column will be simply the energy of the state in "st0001/mins/min1/" in eV. Otherwise, an "na" will be put in there for "not available".

Next line: column six If the saddle is "good", this column will record energy of the state in column 5. If the saddle is "bad", this column will be simply the energy of the state in "st0001/mins/min2/" in eV. Otherwise, an "na" will be put in there for "not available".

C. JOBS.DAT

Here is an sample "jobs.dat" file.

————————————————-

location Status JobId JobType Energy Force Curv.
st0008/pr0001 completed 1 dimer -374.9 0.01 -1.34
st0008/pr0001/mins/min1 queue 2 minimization - - -
st0008/pr0001/mins/min1 queue 3 minimization - - -

————————————————-

File "jobs.dat" is in the aKMC home directory and contains the following information.

First line It is a mandatory description line showing the meaning of each column in the following parts of the file. "joblocation JobStatus JobId JobType Energy(eV) Force(eV/A) Curvature" are quite self-explanatory.

column one This is the relative address of the job location.

column two This is the job status: "submitted", "queue", "running", "stopped", "completed" or "2bsubmitted". Among them, "completed" means the job has really converged based on the INCAR settings. "2bsubmitted" means somehow the job was not successfully submitted and the script will try next time; the reasons include the number of jobs in the queue is exceeding the MaxJobs in CONFIG or the queue system limit, or just bad files.

column three This is the job type which can be "dimer", "minimization", or "dynmat".

column four to six Those are for energy, force and curvature, respectively. No values will be marked "na".

It is mandatory that each process in the state must have an entry in the jobs file. If the users want to add some processes in the state folder (e.g., st0001), they also need to create entries in jobs.dat for those added processes.If a trivial jobid is needed so that the script will take the new processes as "finished saddle point searches", we recommend using "jobid_marker".

D. AKMC.DAT

Here is an sample 'akmc.dat" file.

————————————————-

Step stnumber property address stenergy(eV) ftenergy(eV) ChosenPr steptime(s) totaltime(s)
1 st0001 new st0001 -374.2 -374.8 pr0002 3.7758e-07 3.7758e-07
2 st0002 new st0002 -374.8 -375.4 pr0008 1.6978e-07 5.4736e-07
3 st0003 new st0003 -375.4 -375.5 pr0012 1.0949e-07 6.5685e-07
4 st0004 new st0004 -375.5 -376.0 pr0020 1.1486e-05 1.214285e-05
5 st0005 new st0005 -376.0 -375.7 pr0032 2.4170e-05 3.631285e-05
6 st0003 repeat st0003 -375.7 -376.0 pr0011 2.4171e-05 6.048385e-05
7 st0006 new st0006 -375.7 -376.0 pr0018 2.4171e-05 8.465485e-05
8 st0007 repeat /somewhere/st0002 -375.7 -376.0 pr0038 2.4171e-05 1.0882585e-04
9 st0008 new st0008 -375.7 -376.0 pr0029 1.6978e-07 1.0899563e-04

————————————————-

File "akmc.dat" is in the aKMC home directory and contains the following information.

First line It is an mandatory description line showing the meaning of each column in the following parts of the file: "akmcStep stnumber property target stenergy(eV) ftenergy(eV) totaltime(s)".

column one This tells the simulation is at which KMC step.

column two This is the name of the state.

column third This tells if the state at that time was new or not. "new" means that state had never appeared before (including in the possible database) while "repeat" means the opposite.

column four This is the address of the state. It looks trivial at most time, but if the state is repeated in some database, this address will be the key to get access to the information of the repeated state (see st0007 in the sample file).

column five The chosen process from that state.

column six The energy of the state, in eV.

column seven The energy of the chosen state, in eV.

column eight The time elapse from this state, in eV

column nine The total simulation time, in seconds.

VIII. WHAT HAPPENS WHEN YOU TYPE "AKMC.PL"

First of all, you should only run akmc.pl in the aKMC home directory, i.e., where the "config" file is. After you type the command of "akmc.pl", the script first reads "config" file. Then it checks MaxJobs set by the "config" file. It has to be a non-negative number. After that, the script decides if the dynamic matrix calculation is necessary. The next step is to figure out the status of this aKMC run. It reads the "akmc.dat" file to pick up its memory of the previous step and current step and figure out if the current is a new state, or if it has been started, etc. If the script finds the current step (state) is still being calculated, it will tell the code ("akmc_check.pl)

to check jobs.dat file and update the job status information. Then it will switch to the main loop by reading information from "st.dat" file in the current state folder and handles each process based on its status and the updated job information. The screen output will tell you the progress of the script running.

IX. INTRODUCTION TO ADAPTIVE KINETIC MONTE CARLO SIMULATION

The original idea about adaptive kinetic Monte Carlo (aKMC) simulation was presented in the article by Graeme Henkelman and Hannes Jónsson [*G. Henkelman and H. Jónsson, Long time scale kinetic Monte Carlo simulations without lattice approximation and predefined event table, J. Chem. Phys. 115, 9657-9666 (2001)*]. A new article on the improvement of aKMC is published by Lijun Xu and Graeme Henkelman [*Lijun Xu and Graeme Henkelman, Adaptive kinetic Monte Carlo for first-principles accelerated dynamics, Journal of Chemical Physics 129, 114104 (2008)*]. The following paragraphs have also been posted on the web page of Henkelman group in the University of Texas at Austin (http://theory.cm.utexas.edu/henkelman/research/ltd/).

A. Limits of Molecular Dynamics

Long time dynamics (LTD) refers to dynamic simulation of physical and chemical processes over the time scale much longer than traditional molecular dynamics does. Molecular dynamics simulates dynamic processes by solving Newtonian equation of motion via finite difference approximation. The time step is typically at the femtosecond level. Therefore, molecular dynamics can only simulate events efficiently up to pico- or nano-seconds. However, a lot of interesting events, such as diffusions and chemical reactions, can only happen at the time scale of milli-seconds, seconds or even hours or days. Those events are usually called "infrequent", "rare", or "slow" events. Obviously, it would take an enormous number of steps to simulate rare events with molecular dynamics.

B. Transition State Theory

The potential energy surface (PES) is a multi-dimensional surface on which each energy point corresponds to a possible configuration of the system. There are regions where the system spends most of its time. Those regions are usually local minima or low energy areas corresponding to stable system configurations (e.g., a stable particle, a favorable adsorbate conformation etc). Meanwhile, there are also regions the system rarely visits. From the view of dynamics on the PES, system always vibrates around those local minima (or basin areas), while it occasionally hops from one local minimum to another. That hopping could contribute most to the LTD process.

Transition state theory (TST) assumes that system has to cross over some transition state (usually a narrow bottle-neck type of high energy region on the PES) which the system rarely visits, to move from one local minimum to another. TST uses a statistical estimation of how fast the transition is in terms of a rate constant at a certain temperature. For a lot of solid systems, a harmonic approximation (hTST) can be assumed for both minima and transition states. Then, TST can give a most probable transition pathway (minimum energy path or mechanism) which connects two local minima via a saddle point. The rate of the transition can be calculated from the energies and normal modes of both local minimum and saddle point. Therefore, TST provides a solution to simulate the most important part of LTD - rare events. Combined with molecular dynamics simulation of frequent evens, a multiple time scale of LTD can be achieved with good efficiency.

C. Kinetic Monte Carlo

Combined with hTST, kinetic Monte Carlo (KMC) simulation is very useful in long time scale dynamics. Given those rate constants calculated from transition state theory, KMC can build an event table with all the transition mechanisms (for rare events). An event can be selected from Boltzman distribution by generating a random number; fast events have larger rate constants and therefore, are more probable to be chosen. The system configuration will be updated the end point of the chosen minimum energy pathway. The whole process is repeated for the new state. By this means, the evolution of the system on the PES can be efficiently simulated over a long time scale.

D. Adaptive kinetic Monte Carlo

The key step in KMC combined with hTST is to build a reliable event table for each state the system visits. It is rare that those events in the table can be known beforehand (except for some simple systems). In most cases, the event table has to be built on the fly (or adaptively) for each specific state. We call this type of long time scale simulation "adaptive kinetic Monte Carlo" (aKMC). It is going to be our main LTD tool.

E. Saddle point searching methods

By now, the whole LTD problem has been reduced to finding those saddle points for a given configuration. There are quite a few methods on searching for saddle points (see review). Our choice is the Dimer Method developed by Graeme Henkelman and Hannes Jnsson.

It's a minimum mode following method using only the first order derivative of the potential.

F.   Empirical potential and DFT

Adaptive kinetic Monte Carlo simulation based on dimer method requires an accurate evaluation of the system energy and forces on atoms. Based on how energy and forces are calculated, aKMC can be carried out through either empirical potentials or first principle methods such as density functional theory. Our group has developed a distributed computing system (EON) for using empirical potentials in the aKMC simulation. It takes advantage of idle computing resources on the internet. The server sends out dimer searching jobs to client computers, collects the results when the jobs are finished and does the KMC steps. If there are no reliable empirical potentials available for the system, some quantum calculations have to be used to get energy and forces for the system. We have also developed a scripting program (AKMC) to do the aKMC simulation through the VASP density functional calculations. The AKMC scripts handles every detail of the adaptive kinetic Monte Carlo simulation and frees people from tedious things such as submitting and checking jobs. A single step of aKMC simulation can be a powerful tool for mechanism studies in Surface Science and Catalysis.