Supplementary Material for "Accelerating rare events using Temperature Programmed Molecular Dynamics: A review"

Saurabh Shivpuje [a], Methary Jaipal [b] and Abhijit Chatterjee [b]*

[a] Department of Chemistry, Indian Institute of Technology Bombay and

[b] Department of Chemical Engineering, Indian Institute of Technology Bombay,

*Email: abhijit@che.iitb.ac.in

1. Ag trimer system

We consider an example of silver (Ag) trimer system. In this system, three adsorbed Ag atoms present in trimer configuration are placed on the surface of silver slab. The initial configuration of Ag trimer is shown in Figure 1(a), which we denote as state index 0. Transitions from state 0 sought using TPMD. Newly discovered final states are denoted by positive integer numbers 1 to n, n being the number of kinetic pathways discovered using TPMD. Panels (b) and (c) in Figure 1 show the states most typically visited from state 0.

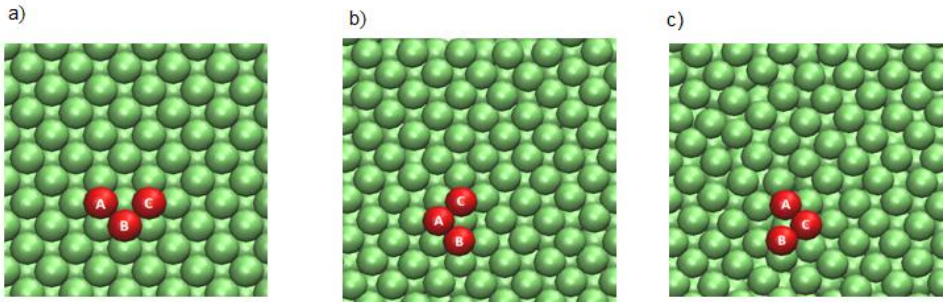

Figure 1. Ag trimer atoms shown in red color, and atoms in the Ag slab are shown in green color: (a) Initial state 0, (b) state 1, (c) state 2.

One thousand independent TPMD calculations were performed from state 0 by randomizing the initial velocities. The temperature is increased from 300 K to 900 K with $\Delta T = 100$ K and each stage is of duration $\tau = 30\ ps$. The temperature program is given by the solid line in Figure 2(a), whereas symbols denote temperature versus time for three different MD trajectories.

The MD trajectories collected were processed. After detecting the transition the final state index and the first passage time (FPT) are stored in the file `statestat.txt`. A histogram for the first passage time is shown in Figure 2(b).

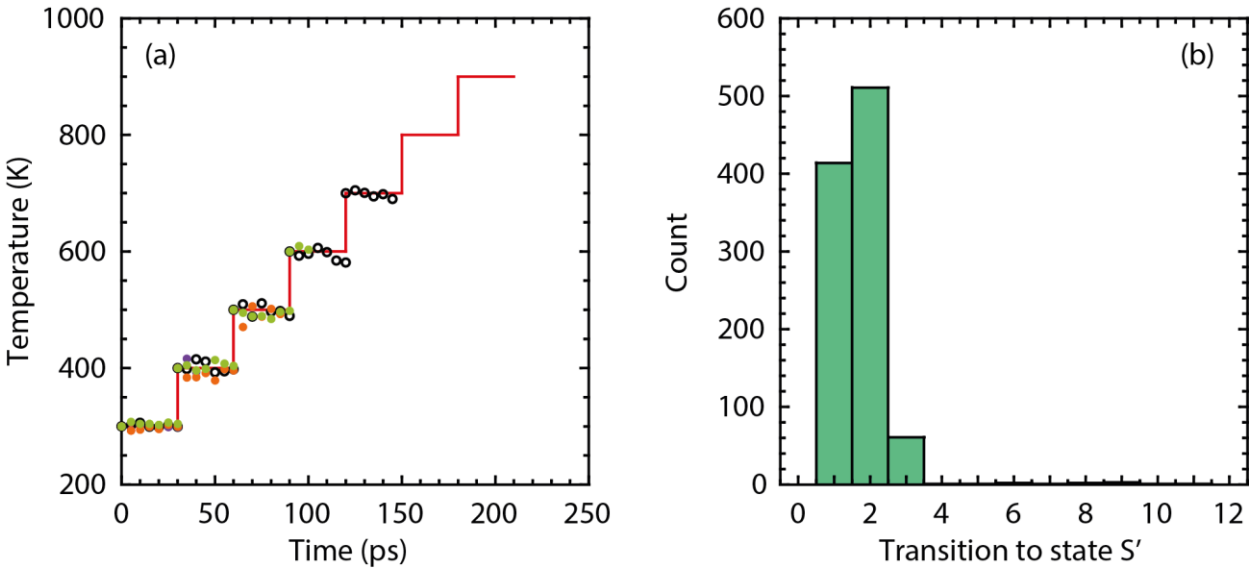

Figure 2. (a) Set-point temperature for different stages (in lines) alongside example of temperatures measured with 3 different MD simulations, (b) Count of transitions in each temperature stage.

Next, statestat.txt is analyzed using codes discussed in Section 2.3. This analysis provides estimates of the rate constants at each temperature stage and also calculates the pre-exponential factor and energy barrier.

2. Procedure

There are three steps involved in deriving the results for any general system. Here each step is performed using different program.

Requirements:

1) LAMMPS : an open source molecular dynamics program
2) Bash shell: a command language interpreter present in almost all UNIX based OS
3) Fortran 90: programming language; gfortran compiler

2.1 TPMD using LAMMPS

First step of this process requires LAMMPS package. LAMMPS is open source molecular dynamics software [1]. The detailed explanation about its installation and usage is given on its official website and in its instruction manual.

The (zip) directory TPMD contains three files which are required as an input by LAMMPS:

Filename	Purpose
in.agtrimer	input file for LAMMPS executable

ag.txt	Simulation box details, like dimensions, atom indexes and their coordinate
Ag_u3.eam	EAM potential parameter file [2]
Note: LAMMPS executable file (lmp_mpi or lmp_serial, based on user machine) One should create this file by referring to LAMMPS manual. NOTE: For this particular example, one must make sure to include MANYBODY package while creating executable file	

Here we explain the syntax of LAMMPS input file which implements the TPMD algorithm (although without performing on-the-fly detection of transitions). Transitions are separately detected via post-processing. The input file is as below (File: in.agtrimer, Directory: TPMD):

```
1  #############INITIALIZATION#####################
2  dimension              3
3  units                  metal
4  atom_style             atomic
5  boundary               p p f
6  variable v equal 47845  #random_number
7
8  #################ATOM DEFINITION#################
9
10 read_data ag.txt
11 group mobile id 1:787
12
13 ########################POTENTIALS#######################
14
15 pair_style eam
16 pair_coeff * * Ag_u3.eam
17
18 ########################TPMD#######################
19
20 dump          1 all custom 3000 nvt.lammpstrj id type x y z vx vy vz
21 dump_modify 1 sort id
22
23 timestep       0.001
24
25
26 velocity all create 300 $v
27 velocity all scale 300
28 fix 1 mobile nvt temp 300 300 0.1
29 run 30000
30 unfix 1
31
32 ###Implementation of the temperature program###
33 velocity all create 400 $v
34 velocity all scale 400
```

```
fix 2 mobile nvt temp 400 400 0.1
run 30000
unfix 2

velocity all create 500 $v
velocity all scale 500
fix 3 mobile nvt temp 500 500 0.1
run 30000
unfix 3

velocity all create 600 $v
velocity all scale 600
fix 4 mobile nvt temp 600 600 0.1
run 30000
unfix 4

velocity all create 700 $v
velocity all scale 700
fix 5 mobile nvt temp 700 700 0.1
run 30000
unfix 5

velocity all create 800 $v
velocity all scale 800
fix 6 mobile nvt temp 800 800 0.1
run 30000
unfix 6

velocity all create 900 $v
velocity all scale 900
fix 7 mobile nvt temp 900 900 0.1
run 30000
```

The MD calculation is performed till the end of the 900 K stage. Later while detecting transitions, we shall only note the first transition from state 0. The low energy barriers involved in the transition to states 1 and 2 imply that it is quite likely that the transitions will be observed in the MD calculations.

In the above script, a line beginning with '#' is considered as comment and will not be executed in LAMMPS. The script contains four different sections to simplify this explanation.

In the first section named 'Initialization' we have defined the primary requirements for our job.

In the second section 'Atom Definition' we defined the system configuration. The coordinates of each atom in the system is given in file named 'ag.txt', which is referred using *read_data* command in LAMMPS (line 10). The lower atomic layers of Ag slab are frozen, while other are

allowed to move. To achieve this, the moving atoms are grouped together using *group* command (line 11).

In the third section we provide the potential parameters that describe all the interactions in the system. The EAM potential is contained in the potential file named 'Ag_u3.eam (line 16). All these potential parameters are provided according to the syntax of LAMMPS mentioned in latest version.

Note that the thermalization step has not been included. In the fourth section of above script, TPMD, the temperature program is written. On line 23 we set the timestep to be 1 femtosecond. On lines 20 and 21 we define the format of trajectory we want to get as output. From line 26 onwards, the temperature program is set up. Each block consisting five consecutive lines are to execute one temperature stage. Every time we increase the temperature by rescaling the velocities of atom; for example on line 26 and 27 we set the temperature for first stage which is 300K. On the next line we define the thermostat, in this example it is Nose-Hoover as described in LAMMPS. Then we set the duration of each temperature stage using 'run' command in terms of number of timestep; in this example it is 30000 timestep, each timestep is of 1 fs as defined earlier; which means each is stage of duration 30 picosecond. While going from one temperature stage to next, we delete the earlier thermostat fixes so that velocities can be rescaled to new temperature. This is done using 'unfix' command. Similar to this block, the next temperature stages are defined.

The output files generated after above process are:

Filename	Purpose
log.lammps	Default log file that is generated in LAMMPS. It mainly contains the thermodynamic details of process. One can control the format and content of this file using various commands as given in LAMMPS manual.
nvt.lammpstrj	The trajectory file that we have asked to generate. This is the input file for our next step.

2.2 Detecting transition and generating FPT data

This step requires the Bash shell software, which is present in mostly all UNIX based OS. It is assumed that each of the n independent trajectory file (nvt.lammpstrj) are placed in separate directories with names: $1, 2, 3, \ldots, n$. Next, copy the seven files from (zip) directory named FPT to the parent directory containing directories numbered 1 to n. The user inputs are provided in 'parameters.txt', which the user is free to modify.

An example of parameters.txt is given below. Do not to make any changes in the text highlighted in bold font below:

#Number of significant digits involved in calculation is depended on input we provide here. It is recommended to use distance parameter with atleast 4 significant digit after decimal point

Total number of atoms in the system=1179
Number of trajectories=1000
Number of consequtive frames for transition validation=5
Tolerance(distance unit)=0.500
Distance limit from initial position to cosider as transition(distance unit)=2.05
Size of box x-direction(distance unit)=28.6230
Size of box y-direction(distance unit)=28.6230
Size of box z-direction(distance unit)=35.0000
Number of frames in each trajectory (excluding initial configuration frame)=70
Time duration between two subsequent frames (time unit)=3

After editing parameters.txt, execute ‘./run.sh’. Before running this command make sure each ‘.sh’ extension file has access permission as ‘executable’. It can be done using command “chmod +x filename”. If you do not see any error on terminal during its full run, then it means the program is executed successfully.
Six other files in FPT directory have the following purpose:

Filename	Purpose
run.sh	The main script that generates the list of FPTs
transition.sh	Detects the transition in the MD trajectory
state.sh	Analyzes the transition and assigns the state index for the configuration seen after the transition
scounter.txt	Keeps count of number of states during analysis **Note: Before starting make sure file contains only the text ‘1’ in it**
unpbc.sh	Takes care of periodic boundary conditions while detecting transitions
sign.awk	Supporting file for unpbc.sh

The output files generated after above analysis are:

Filename	Purpose
log.txt	Log file that gives information of total number of states observed and the details of transition in order: trajectory number, transition atom number, FPT, state number
statestat.txt	The output file which contains the data in two columns in the form of: State index followed by FPT. This is the input file that we will be using in the next and final step of process.

The transition detection scheme provided in the bash script file state.sh has limited functionality. In state.sh it is assumed that only one atom is involved in the move from state 0. Typical output contained in statestat.txt for the Ag trimer system is given below:

1 102
2 93
1 27
1 72
2 45
1 96
1 90
2 102
1 42
1 87
2 171
1 147
2 78
3 75

2.3 Analysis of the first passage times

Following codes have been provided to analyze the first passage times. All codes should be present in a single directory.

File	Purpose
fpt.input	Provide user inputs for analyzing the first passage times
run.sh	Can help user create the file fpt.input, and thereafter analyze the first passage times.
*.f90 files	Fortran files for analyzing the first passage times. Files included as main.f90, tpmdmodule1.f90, tpmdmodule2.f90, tpmdmodule3.f90, tpmdvariables.f90

Make sure the file run.sh is executable. It can be made executable using command "chmod +x run.sh". There are four steps in run.sh. The first three are used to create fpt.input.

To execute type './run.sh <filename> <skipstep>'. Here filename contains the first passage times (in our example statestat.txt) and skipstep refers to the number of steps to be skipped in run.sh.

The command `./run.sh statestat.txt 3`, results in first three steps to be skipped. Note that skipstep is an optional input; run.sh provides an interface to create the fpt.input file. The fpt.input file contains the following information

```
"statestat.txt"  input file containing TPMD FPT data
11 number of states found
998 number of first passage times
30 Stage duration tau (in ps)
300 Temperature in stage 0 (initial stage, in K)
100 Temperature step (in K)
900 Maximum temperature (in K)
0 Index for initial state (typically 0/1)
2 number of states to be included/excluded
1  final state index (1)
2  final state index (2)
```

The analysis is performed for a group of states. Rate constants are calculated based on the transitions to these states. For our trimer example, states 1 and 2 are symmetrically equivalent. It is useful to group these states. Therefore we mention in the fpt.input file:

```
2 number of states to be included/excluded
1  final state index (1)
2  final state index (2)
```

When number of states is positive (here +2), it means that these many states are to be included in the analysis. Next we mention that states 1 and 2 should be included.

Output obtained from the code after analyzing the first passage times:

```
$ ./run.sh statestat.txt 3
 Analyzing rates for the paths ...
 >>>    0->   1 >>>    0->   2  from file: statestat.txt
 File has been read ..
 >>> T0(K):   300.000
 >>> deltaT(K):   100.000
 >>> Tmax(K):   900.000
 >>> tau(ps):    30.000
 >>> #datapoints:   998
 >>> MLE for Arrhenius parameters


 ################## STEP 1. Calculate Arrhenius parameters #################
 Obtain MLE rate (Arrhenius) ...
Temperature stages: 0 - 6
Stage  0 .. temperature    300.0 K
Stage  1 .. temperature    400.0 K
Stage  2 .. temperature    500.0 K
Stage  3 .. temperature    600.0 K
Stage  4 .. temperature    700.0 K
Stage  5 .. temperature    800.0 K
Stage  6 .. temperature    900.0 K


 Breakup of theta for first 3 transitions...
   1:    30.000    30.000    30.000    12.000     0.000     0.000     0.000 FinalTemperature:
600.00000000000000      Stage:           3
   2:    30.000    30.000    30.000     3.000     0.000     0.000     0.000 FinalTemperature:
600.00000000000000      Stage:           3
   3:    27.000     0.000     0.000     0.000     0.000     0.000     0.000 FinalTemperature:
300.00000000000000      Stage:           0


 Sum of inverse T_finalstage:   1.9782196482196470E-003
 Sum of ln T_finalstage:  0.54410495748378174
 npts:         925
theta:    2.94840E+04    2.67720E+04    1.85280E+04    8.11800E+03    1.93800E+03    4.02000E+02
4.20000E+01  SUM:   85284.000000000000
count:    35   171   352   259    88    15     5  SUM:         925
   Iteration   Effective Ea(eV)    Residual    DeltaResidual
...  1         2.100E-01        -1.068E-04        -1.919E-03
...  2         1.610E-01         8.416E-08        -2.479E-03
...  3         1.611E-01         8.203E-12        -2.478E-03
 Effective nu & barrier:  0.71822845382957323       0.16106052867238790
 ################## STEP 2. Calculate uncertainty in Arrhenius parameters #################
 >>> Effective prefactor:    7.18228E-01  +/-  1.11E-01
 >>> Effective barrier:    1.61061E-01  +/-  6.27E-03
 >>>
 >>>
```

```
################## STEP 3. MLE for rates @ constant temperature #################
Validating Arrhenius approximation by comparing MLE rate to Arrhenius rate from TPMD data ...

_____________________________________________
>>>Stage    0 MLERate    1.197E-03 +/-    2.04E-04
>>>           ArrRate    1.414E-03 +/-    4.06E-04
>>> Temperature     300.000 Ratio:    8.462E-01

_____________________________________________
>>>Stage    1 MLERate    6.423E-03 +/-    5.02E-04
>>>           ArrRate    6.714E-03 +/-    1.60E-03
>>> Temperature     400.000 Ratio:    9.567E-01

_____________________________________________
>>>Stage    2 MLERate    1.903E-02 +/-    1.05E-03
>>>           ArrRate    1.709E-02 +/-    3.62E-03
>>> Temperature     500.000 Ratio:    1.113E+00

_____________________________________________
>>>Stage    3 MLERate    3.214E-02 +/-    1.99E-03
>>>           ArrRate    3.187E-02 +/-    6.25E-03
>>> Temperature     600.000 Ratio:    1.008E+00

_____________________________________________
>>>Stage    4 MLERate    4.541E-02 +/-    4.48E-03
>>>           ArrRate    4.974E-02 +/-    9.24E-03
>>> Temperature     700.000 Ratio:    9.130E-01

_____________________________________________
>>>Stage    5 MLERate    3.731E-02 +/-    1.45E-02
>>>           ArrRate    6.944E-02 +/-    1.24E-02
>>> Temperature     800.000 Ratio:    5.373E-01

_____________________________________________
>>>Stage    6 MLERate    0.000E+00 +/-    2.33E-02
>>>           ArrRate    9.003E-02 +/-    1.57E-02
>>> Temperature     900.000 Ratio:    0.000E+00
```

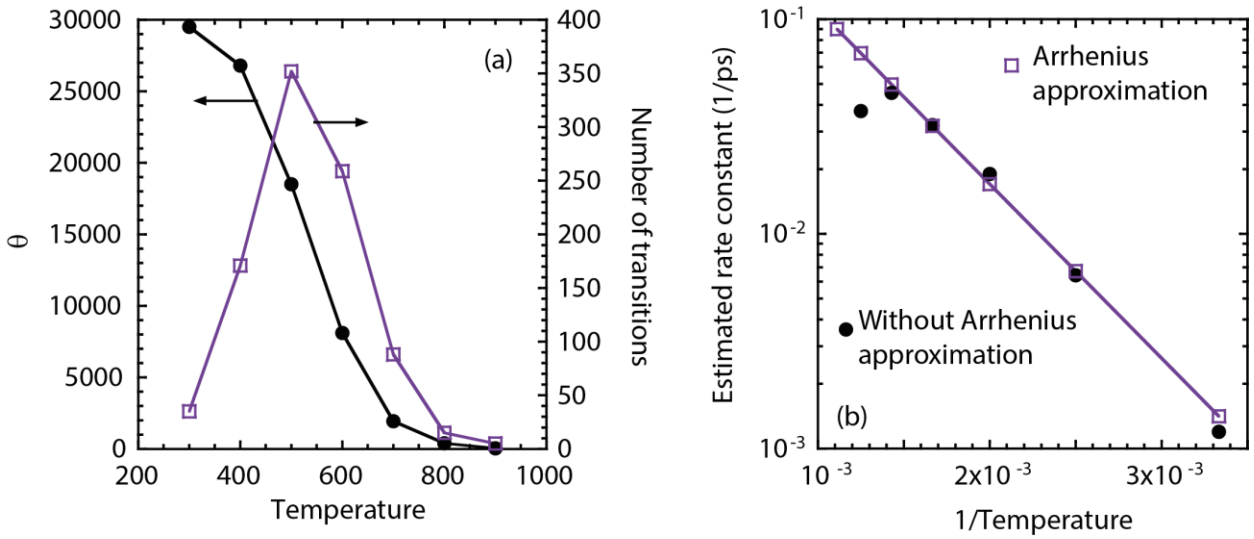

Figure 3. (a) Time accrued (in *ps*) at each temperature state (θ) using TPMD simulations and number of transitions for the trimer example. (b) Comparison of the rate constants estimated with and without the Arrhenius approximation.

Figure 3a shows that transitions were observed in almost all stages. Figure 3b shows that the rate constants with and without Arrhenius approximation are in good agreement. From these results we conclude that the Arrhenius approximation is valid.

```
################## STEP 2. Calculate uncertainty in Arrhenius parameters #################
 >>> Effective prefactor:    7.18228E-01  +/-  1.11E-01
 >>> Effective barrier:    1.61061E-01  +/-  6.27E-03
```

In comparison, energy barrier calculated with climbing-image NEB using 12 images and spring constant of 0.1 eV/Å^2 is 0.21 eV.

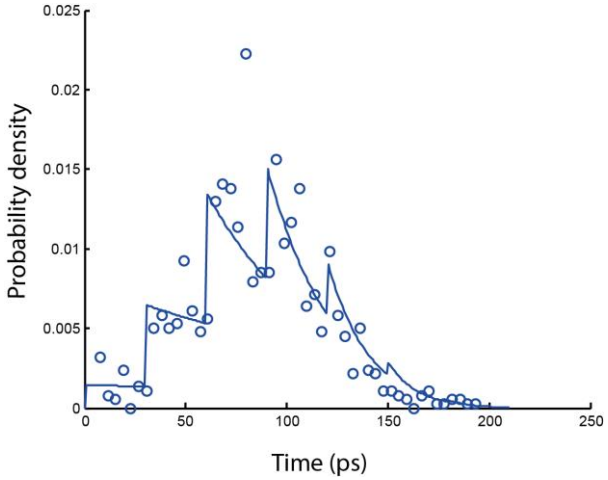

Figure 4. Measured probability density (circles, in ps^{-1}) and TPMD distribution (lines, in ps^{-1}) using the Arrhenius approximation.

A comparison of the measure probability density (circles) and TPMD distribution (lines) using the Arrhenius approximation is shown in Figure 4. The good agreement confirms that the TPMD assumptions are valid.

Since transitions to states 1 and 2 were included, the pre-exponential factor for the pathways $0 \rightarrow 1$ or $0 \rightarrow 2$ will be $0.359\ ps^{-1}$.

From the above analysis the rate constant from state 0 at 300 K is $1.197 \times 10^{-3}\ ps^{-1}$. Thus, the average time for a transition at 300 K is $835\ ps$. The average time for a transition in TPMD is $92\ ps$. This is obtained using the total time ($85284\ ps$) accrued divided by number of transitions (925). See

```
theta:    2.94840E+04    2.67720E+04    1.85280E+04    8.11800E+03    1.93800E+03    4.02000E+02
4.20000E+01  SUM:   85284.000000000000
count:    35   171   352   259    88    15     5  SUM:         925
```

The acceleration depends on the kinetic pathway. Although the activation barrier is small ($0.16\ eV$) the acceleration obtained with TPMD is 9.07 times (calculated as 835/92). See Ref. [3]

for discussion on high barrier events involving the trimer example which happen at millisecond timescales at 300 K. The acceleration is a million times in this case.

Notes

We thank the unknown reviewers for making the suggestion to incorporate an example with TPMD, which led to the creation of these demonstration codes with LAMMPS. Note that the LAMMPS and transition detection scripts do not employ all aspects described in the main text in its entirety, and have been presented for the purpose of illustrating some features of our approach. On the other hand, the fortran codes for analysis of first passage times are fairly advanced and can be employed with a wide-range of materials applications. The codes, which will be updated on a regular basis in future, can be downloaded from website of Prof. Chatterjee's research group. More information can obtained by sending an email to Prof. Chatterjee (abhijit@che.iitb.ac.in).

References

[1] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, J. Comput. Phys. 117 (1995) 1–19.

[2] S.M. Foiles, M.I. Baskes, M.S. Daw, Embedded-atom-method functions for the fcc metals Cu, Ag, Au, Ni, Pd, Pt and their alloys, Phys. Rev. B. 33 (1986) 7983–7991.

[3] S. Divi, A. Chatterjee, Accelerating rare events while overcoming the low-barrier problem using a temperature program., J. Chem. Phys. 140 (2014) 184115. doi:10.1063/1.4875476.