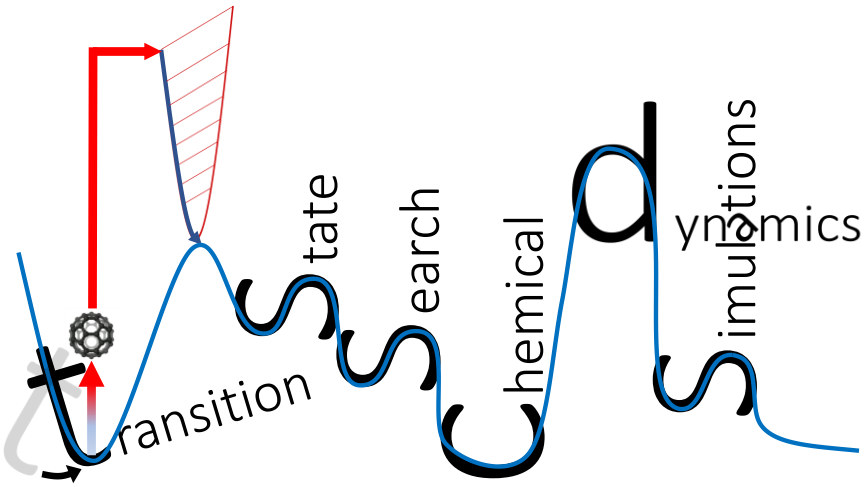


tsscds1.1

transition state search using chemical dynamics simulations

Main developer:

Emilio Martínez-Núñez
Departamento de Química Física, Facultade de Química
Avda. das Ciencas s/n
15782 Santiago de Compostela, SPAIN
emilio.nunez@usc.es

With contributions from:

George L. Barnes, Aurelio Rodríguez, Roberto Rodríguez-Fernández and Saulo A. Vázquez

Contents

1. Introduction

tsscds-1.1 computer program has been designed to discover reaction mechanisms and solve the kinetics in an automated fashion, using chemical dynamics simulations.

The basic idea behind this program is to obtain transition state (TS) guess structures from trajectory simulations performed at very high energies or temperatures. From the obtained structures, minima and product fragments can be optimized following the intrinsic reaction coordinate (IRC), and the reaction network can thus be constructed. Finally, the kinetics is solved using Kinetic Monte Carlo (KMC) method.

The program is interfaced with MOPAC2016 and GAUSSIAN09, but work is in progress to incorporate more electronic structure programs.

2. How to cite the program

Work based on this code should include the following references:

1) Martinez-Nunez, E. *Phys. Chem. Chem. Phys.* 2015, *17*, 14912–14921.
2) Martinez-Nunez, E. *J. Comput. Chem.* 2015, *36*, 222–234.
3) MOPAC2016, Version: 16.307, James J. P. Stewart, Stewart Computational Chemistry, web-site: HTTP://OpenMOPAC.net.

This tutorial is thought to guide you through the steps to obtain the reaction mechanisms and kinetics of formic acid (FA) decomposition.

3. Installation

Untar and unzip the file tsscds-SOURCE-1.1.tar.gz:

```
tar xvfz tsscds-SOURCE-1.1.tar.gz
```

Before installing tsscds, be aware that the program makes use of the following packages that need to be installed in your linux distribution:

environment modules
g09
parallel
python2 (with **numpy** and **scipy** libraries)
sqlite3
zenity (version 3)

Once "**environment modules**" is installed, you can add to your .bash_profile the following line to use the tsscds module:

```
module use path_to_program/modules
```

Where path_to_program is the path where you intend to install tsscds (like $HOME/tsscds-1.1).
You will also need **g09** to run the high-level calculations and it should be run as: g09<input>output.

Once, the above packages are installed, go to tsscds-SOURCE-1.1 folder and type:

```
./configure
```

This will install tsscds in $HOME/tsscds-1.1 by default. If you want to install it in a different directory, type:

```
./configure --prefix=path_to_program
```

Complete the installation:

```
make
make install
```

4. How to start using the program

To start using any of the scripts described below, you have to load tsscds/1.1 module:

```
module load tsscds/1.1
```

If you use the program in FinisTerrae infrastructure (CESGA, Spain), type the following instead:

```
module load tsscds/1.1ft2
```

5. Finding reaction mechanisms and solving the kinetics

All electronic structure calculations described in this section employ semi-empirical Hamiltonians and use MOPAC. To guide you in this tutorial you can try the formic acid (FA) test case that comes with the distribution. Make a working directory and copy FA.dat and FA.xyz files from path_to_program/examples into your working directory. All the scripts described below (except select.sh) should be run from your working directory.

a) Description of the input files

To find reaction mechanisms and solve the kinetics you just need two input files:

i) **molecule.xyz**. Cartesian coordinates of the system, where molecule is the name of our system (like FA.xyz).

ii) **molecule.dat.** This contains all parameters of the calculation, where molecule is the name of the system (like FA.dat).

The file has different sections explained in the following.

General section. Here the user provides details of the system and electronic structure calculations**.**

```
--General section--
 molecule FA
 HighLevel b3lyp/6-31G(d,p)
 charge 0
 mult 1
```

Description of the keywords

molecule**:** is the name of the system (FA in the example). A file **FA.xyz** must exist in the folder.

LowLevel: is any of the semiempirical methods implemented in MOPAC2016. You can use a combination of MOPAC keywords. For instance, PM7 singlet excited state calculations can be invoked with:

```
LowLevel pm7 singlet cis c.i.=6 root=2 meci
```

If the keyword is not present (like in the example above) PM7 is the default.

Highlevel: indicates the high level of theory employed. To correct the energy using a higher level, the following could be employed:

```
HighLevel ccsd(t)/6-311+G(2d,2p)//b3lyp/6-31G(d,p)
```

Supported methods are HF, MP2 and DFT for geometry optimizations and HF, MP2, DFT and CCSD(T) for single point energy calculations.

charge: is the charge of the system.

mult: is the multiplicity of the system.

CDS (Chemical Dynamics Simulations) section. Here the user provides details of the accelerated dynamics simulations.

```
--CDS section--
sampling microcanonical
ntraj 1
```

Description of the keywords

sampling: indicates the method employed to select the initial conditions: microcanonical, canonical, association and external**.**

Microcanonical and canonical samplings provide initial conditions for the dynamics using energies and temperatures, respectively. See "advanced users" for more details.

Common keywords of the above two samplings:

seed: can be employed to run a test trajectory. This is the seed for the random number generator. If you plan to run more than one trajectory do not use this keyword, and every trajectory will have a different random number seed.

ntraj: is the number of trajectories.

Association. This sampling is explained in section f.

External. With this option you can couple your dynamics results, obtained with any other code, with our BBFS algorithm. You just need to copy the dynamics xyz files (with a time step of 1 fs) in folder coordir. The xyz files have to be named molecule_dynX.xyz, where X goes from 1 to ntraj, and molecule is the name of your system. Example for 10 external trajectories.

```
--CDS section--
sampling external
ntraj 10
```

CAVEAT: Finally, if you combine external trajectories with those computed internally using MOPAC, you need to employ the same labelling in both cases.

BBFS (Bond Breaking/Formation Search) section. BBFS algorithm selects TS guess structures monitoring changes in the adjacency matrix,[1] but the user can impose some restrictions based on the energy and imaginary frequency of the structures found.

```
--BBFS section--
emaxts  200
emints -100
freqmin 200
```

Description of the keywords

emaxts: is the maximum energy (in kcal/mol) of the TSs that will be selected.

emints: is the maximum energy (in kcal/mol) of the TSs that will be selected.

The above energies are relative to the optimized structure provided in FA.xyz. In the figure above ***emints*** is set to **-100** kcal/mol because the initial structure might not be the global minimum.

freqmin: is the minimum imaginary frequency (in absolute value and cm^{-1}) of the TSs that will be selected.

Screening of the structures section. To screen the structures for possible redundancies and/or fragmentations, some features of each structure are calculated: energy, SPRINT coordinates,[2] degrees of each vertex and eigenvalues of Laplacian matrix.[3] Comparing these values for two structures, the mean absolute percentage error (MAPE) and the biggest absolute percentage error (BAPE) can be obtained. While **avgerr** and **bigerr** set maximum values for MAPE and BAPE, respectively, **thdiss** refers to the eigenvalues of the Laplacian (EL). The number of zero ELs provides the number of fragments in the system.

```
--Screening of the structures section--
avgerr 0.008
bigerr 2.5
thdiss 0.1
```

Description of the keywords

avgerr: is the maximum value of the MAPE for two structures to be considered equal.

biggerr: is the maximum value of the BAPE for two structures to be considered equal.

thdiss: is the threshold for a EL to be 0. In the above figure, if a EL<0.1, then EL=0.

Kinetics section. This part is employed to provide details of the kinetics calculations at the experimental conditions.

```
--Kinetics section--
Rate  microcanonical
EKMC  150
```

Description of the keywords

Rate: can either be ***canonical*** or ***microcanonical***, which means that the rate constants will be calculated according to Transition State Theory (TST) or Rice-Ramsperger-Kassel-Marcus (RRKM) theory, respectively.
EKMC**:** is the energy (in kcal/mol) of the experiment you want to simulate.
TKMC**:** is the temperature (in K) of the experiment you want to simulate.

b) Running the accelerated dynamics

Canonical and microcanonical sampling methods provide initial coordinates and momenta to run accelerated dynamics simulations. Select the number of trajectories with ***ntraj*** and remember to avoid ***seed*** keyword if the number of trajectories is greater than 1. In you want to run 10 trajectories your CDS section should look like:

```
--CDS section--
sampling microcanonical
ntraj 10
```

And the dynamics can be run using:

```
tsscds.sh FA.dat >tsscds.log &
```

You may want to have a look at the ouput file tsscds.log, which gives you information of the calculations. Besides, a working directory called tsdirLL_FA is created with some temporary information that might be checked throughout the calculations. At any time, you can check the transition states that have been found using:

```
tssll_view.sh
```

The output will be something like this:

```
  ts #  MOPAC file name  w_imag    Energy     w1     w2     w3     w4 traj #   Folder
  ----  ---------------  ------    ------   ----   ----   ----   ---- ------   ------
     1           ts1_FA 1587.3i    -35.71  204.3  438.3  461.3  726.8      1       FA
     2           ts2_FA 2009.6i    -17.61  327.2  472.7  522.7 1078.6      2       FA
     3           ts3_FA 2930.8i    -20.17  450.6  586.9  908.6  997.2      7       FA
```

where the first column is the label of each TS, the second is the MOPAC output file name, the third is the imaginary frequency (in cm^{-1}), the fourth one is the energy in kcal/mol, the next four numbers are the four lowest vibrational frequencies (in cm^{-1}). Finally, the last two columns are the trajectory number and the name of the folder where the accelerated dynamics were run.

CAVEAT: since the dynamics employ random number seeds, the results shown in the above figure may differ from those obtained in your computer, and the next few explanations refer to the tslist file shown above.

MOPAC files are stored also in tsidrLL_FA Try, for instance:

```
molden tsdirLL_FA/ts1_FA.out
```

You can also watch the animation of the trajectory, which is stored in coordir folder inside FA:

```
molden coordir/FA_dyn1.xyz
```

c) Running the dynamics in parallel

If you have access to several processors and want to run the dynamics in parallel you can use the script tsscds_parallel.sh. For instance, to submit 50 trajectories split in 5 different tasks (10 trajectories each) you should use:

```
tsscds_parallel.sh FA.dat 5
```

This will create temporary directories batch1, batch2, batch3, batch4 and batch5 that will be automatically removed later on (when the minima are obtained). The TSs found in each individual task will be copied in a common working directory (tsdirLL_FA in our test case), and, as indicated above, using tsll_view.sh script you can monitor the process. If you have slurm workload manager you can submit the jobs to slurm using:

```
sbatch [options] tsscds_parallel.sh FA.dat 5
```

which employs the following default values when no options are specified:
#SBATCH --time=01:00:00

```
#SBATCH -n 40
#SBATCH --output=tsscds_parallel-%j.log
#SBATCH -p shared
#SBATCH --qos=shared
```

Those values can be changed when you submit the job with *options*.

CAVEAT: the use slurm workload manager might be faster but you have to wait until all tasks are completed before going on.

d) Screening the structures, IRC, minima, reaction network, and kinetics

1) Since irc.sh script includes an initial screening of the TS structures, you can check the screening this way:

```
irc.sh screening
```

The above sentence will do the screening and stop. The process involves the use of tools from Spectral Graph Theory using the three threshold values indicated above: ***avgerr, bigerr*** and ***thdiss***. The redundant and fragmented structures are printed on screen as well as in the file **screening.log**. When you start obtaining these structures, please check them and change the above parameters if needed.
Should you change some of the above parameters (***avgerr, bigerr, thdiss***) you need to redo the screening with the new parameters:

```
redo_screening.sh
```

You can repeat the above process until you are happy with the "screening". Now, you can proceed with the irc calculations:

2) Obtaining the IRCs:

```
(sbatch [options]) irc.sh
```

3) Optimizing the minima:

```
(sbatch [options]) min.sh
```

4) Creating the reaction network:

```
rxn_network.sh
```

Once you have created the reaction network, you can grow your TS list by running more trajectories (with tsscds_parallel.sh), which will start the dynamics from the newly generated minima. Convergence in the total number of TSs can be obtained doing:

```
track_view.sh
```

When you are happy with the obtained TSs or you achieved convergence, you can proceed.

5) Solving the kinetics using KMC with the parameters given in the kinetics section:

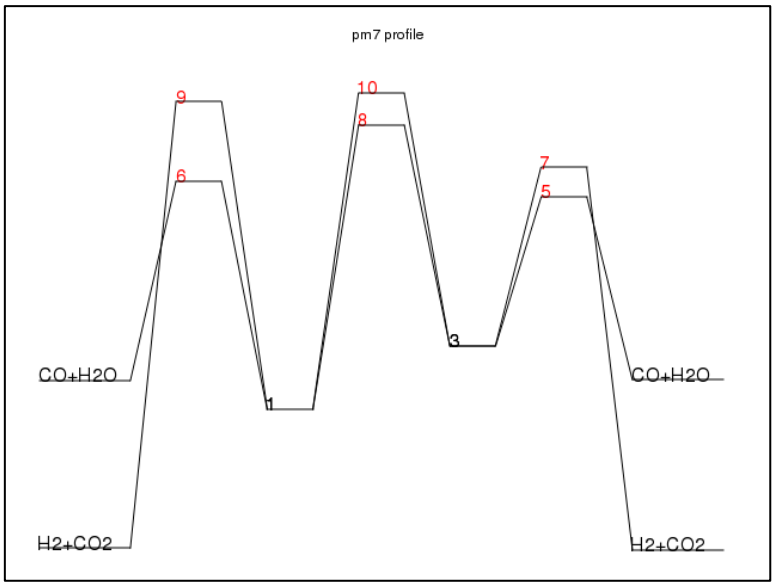
```
kmc.sh
```

6) Gathering all relevant information in folder FINAL_LL_FA:

```
final.sh
```

This folder contains several files:

Energy_profile.gnu: is an energy diagram with the **relevant paths** in gnuplot format. If you change the value of ImpPaths in the kinetics section(see "Advanced users"), you will incorporate/remove some pathways.



MINinfo: contains information of the minima:

```
MIN #    DE(kcal/mol)
    1       -8.340
    2        0.000
    3        5.283
    4        6.710
    5       15.338
Conformational isomers are listed in the same line:
1 2
3 4 5
```

TSinfo: contains information of the TSs:

```
TS  #    DE(kcal/mol)
    1        1.873
    2        9.625
    3       25.137
    4       32.852
    5       37.596
    6       40.962
    7       43.960
    8       53.165
    9       58.155
   10       60.011
   11       90.312
Conformational isomers are listed in the same line:
8 10
```

***table*.db**: with *table* being *min*, *prod* or *ts*. These are sqlite3 tables containing the geometries, energies and frequencies of minima, products and TSs, respectively. The different properties can be obtained using the following, which should be run from the FINAL_LL_FA folder:

```
select.sh property table label
```

where *property* can be: *natom*, *name*, *energy*, *zpe*, *g*, *geom*, *freq*, *formula* (only for prod) or *all*, and label is one of the numbers shown in RXNet (see below) employed to label each structure. For instance, to obtain the geometry of the first transition state you should use:

```
select.sh geom ts 1
```

RXNet: contains information of the reaction network.

```
TS #    DE(kcal/mol)    -------Path info--------
    1        1.873        MIN    1 <-->  MIN    2
    2        9.625        MIN    3 <-->  MIN    4
    3       25.137        MIN    1 <-->  MIN    1
    4       32.852       PROD    1 <--> PROD    2
    5       37.596        MIN    4 <--> PROD    2
    6       40.962        MIN    1 <--> PROD    2
    7       43.960        MIN    3 <--> PROD    1
    8       53.165        MIN    1 <-->  MIN    4
    9       58.155        MIN    2 <--> PROD    1
   10       60.011        MIN    2 <-->  MIN    5
   11       90.312       PROD    2 <--> PROD    7
PROD 1 H2 + CO2
PROD 2 CO + H2O
PROD 7 H2 + CO2
```

RXNet.cg: cg stands for “coase-grained”. By default (see below) the KMC calculations are “coarse-grained”, i.e., conformational isomers form a single state (corresponding to the lowest energy isomer). Such reaction network, which also remove bimolecular channels is shown in the figure below.

```
TS #    DE(kcal/mol)    -------Path info--------
    5       37.596        MIN    3 <--> PROD    2
    6       40.962        MIN    1 <--> PROD    2
    7       43.960        MIN    3 <--> PROD    1
    8       53.165        MIN    1 <-->  MIN    3
    9       58.155        MIN    1 <--> PROD    1
   10       60.011        MIN    1 <-->  MIN    3
PROD 1 H2 + CO2
PROD 2 CO + H2O
PROD 7 H2 + CO2
```

kineticsFvalue: This file contains the kinetics results, namely, the final branching ratios and the population of every species (with non-zero population) as a function of time. In the name of the file, F is either “T” or “E” for temperature or energy, and “value” is the corresponding value. For instance, the kinetics results for a canonical calculation at 298 K would be printed in a file called kineticsT298. A gnuplot file called **populationFvalue.gnu** is also available. It is a plot with the population of each species as a function of time.

Finally, the normal modes of TSs and minima are collected in normal_modes folder in molden format.

As indicated above, by default, in the KMC simulations, the different conformational isomers form a single state, which speeds up the calculations. If you prefer that each conformational isomer will be a single state in the KMC calculations, you should run the RXN again, and solve the kinetics with the new conditions:

```
rxn_network.sh allstates
kmc.sh
final.sh
```

If you are solving the kinetics for a given temperature (rate canonical and TKMC specified in the input file), and you want to obtain the kinetics results for a different temperature, you can use kinetics.sh script with the following arguments:

```
kinetics.sh temp calc (allstates)
```

Where temp is the new temperature of the system (in K), and calc is either ll (low-level) or hl (high-level). At this point, you should employ ll, but hl is available when you complete the hl calculations (vide infra). Finally, with no other options, (as above) the conformational isomers will form a single state, and using allstates as the last argument, will take every conformational isomer as a different state.

e) Summary of the steps needed to find reaction mechanisms and solve the kinetics

1. Load tsscds module:

```
module load tsscds/1.1
```

2. Run the accelerated dynamics in parallel:

```
(sbatch [options]) tsscds_parallel.sh molecule.dat ntasks
```

3. Run the irc calculations:

```
(sbatch [options]) irc.sh (screening)
```

4. Optimize the minima:

```
(sbatch [options]) min.sh
```

5. Create a reaction network:

```
rxn_network.sh
```

6. Here, you might want to iteratively run tsscds going back to step 2.

7. Solve the kinetics:

```
kmc.sh
```

8. Create FINAL_LL_molecule folder:

```
final.sh
```

9. If you are simulating a thermal process, you can solve the kinetics at a different temperature using:

```
kinetics.sh temp calc (allstates)
```

All the above steps can be done using a single script:

```
llcalcs.sh molecule.dat ntasks niter
```

Where niter is the number of iterations. If zenity is available in your linux distribution, the script can be run without arguments, and a GUI will help you enter the arguments.

CAVEAT: The use of llcalcs.sh is only recommended once you verified that the screening process works fine for your system.

f) Finding intermolecular complexes between two molecules

The input file for this type of calculation is slightly different. An example of such input file can be found in path_to_program/examples/assoc.dat (see figure below). Two more additional files are also needed for this example cat.xyz and co.xyz, which are also available in the same folder.

```
--General section--
charge 0
mult 1

--CDS section--
sampling association
A= cat
B= co
rotate 2 com 2.0 1.0

--Screening of the structures section--
avgerr 0.0001
bigerr 5
thdiss 0.1
```

This type of sampling only needs three sections: General, CDS and Screening.
This sampling is employed to optimize complexes between two fragments A and B. It just needs three keywords in the CDS section.

Description of the keywords

***A*=**: is the name of fragment A. A file with the Cartesian coordinates fragA.xyz must also be present.
***B*=**: is the name of fragment B. A file with the Cartesian coordinates fragB.xyz must also be present.
rotate: refers to the method employed to optimize the complexes. The calculation is carried out by taking 100 relative structures of both fragments, obtained via random rotations. Thus, the next two fields **2 com** indicate the pivot positions of the rotations: atom 2 of fragment A and the center of mass (com) of fragment B. The last two numbers **2.0** and **1.0** are values in Å that indicate the distance between both pivots and the minimum intermolecular distance between any two atoms of both fragments, respectively.
With this sampling, you do not need BBFS and kinetics sections. However, you still need to provide the parameters for the screening (*vide supra*).
To run the calculations:

```
tscds.sh assoc.dat
```

This job will submit 100 jobs to find the structures. After the jobs finished, the script will automatically remove duplicates and select the best association "complex".
You can check the optimized structures in folder assoc_cat_co. The program will also select the "best" structure according to the minimum number of structural changes between the complex and the individual fragments and its energy. The structure selected will be called cat_co.xyz. For fragments containing metals (like in this example), the selection is also based on the valence of the metal center.
The file assoc_cat_co/assoclist_sorted collects a summary of the structures and their energies, and the MOPAC2016 output files of each of them are called assocN.out, where N is a number from 1 to 100.

6. Advanced users

The following are keywords that experienced users can employ:

General section

iop: can be employed to specify an iop in gaussian. Example:

```
iop iop(X/Y=Z)
```

where X, Y and Z denote the overlay, option and value, respectively.

CDS section

etraj: can be employed along with microcanonical sampling and is the energy (in kcal/mol) of the accelerated dynamics simulations. This can be a single value (200) or a range, in which case the energy is randomly selected in the given energy range (200-300 for instance).

If etraj is not specified, the program automatically employs the following range of energies: $[16.25\times(s-1)-46.25\times(s-1)]$ kcal/mol, where s is the number of vibrational degrees of freedom of the system. Those values have been determined from the formic acid results and making use of RRK theory. Those are the initial values of etraj, but the program automatically adjusts the range to obtain at least 60% reactivity at the boundaries.

temp: can be employed with canonical sampling and is the temperature of the accelerated dynamics in K. As above, it can be a single number or a range.

In the absence of this keyword, the program automatically defines the following range of temperatures: $[5452.04\times(s-1)/natom-15517.34\times(s-1)/natom]$ K, which has been optimized for formic acid. However, as above, the boundaries are adjusted "on the fly" to obtain a minimum reactivity of 60%.

modes: can be employed together with microcanonical sampling. The default is "all", which means that all modes are excited. If the user wants to excite just the three lowest normal modes something like this should be employed (note that the labels of the normal modes must be comma separated):

```
modes 3 1,2,3
```

atoms: is analogous to ***modes*** explained above, but for a canonical ensemble. It indicates the atoms that are excited when a canonical ensemble is employed (the default is "all").

thmass: can be employed together with canonical, to specify the required minimum mass (in a.u.) of an atom to be initially excited. An example of an input file for a canonical example can be found in this distribution (tsscds_canonical.dat).

fs: is the simulation time (in fs). The default is 500.

BBFS section

Fastmode: using this keyword in a single reaction path, only one point is picked and the Hessian update in the EF optimization is the default in MOPAC. This is the default.

Slowmode: using this keyword up to 3 points are picked in a single reaction path and the Hessian is updated every 10 steps. Obviously this option is slower than fastmode, but it might be tested when the BBFS algorithm is not able to find transition states with efficacy.

Kinetics section

***imin*:** is the starting minimum in the kinetics simulations. The default is the starting reference structure.

nmol: is the number of molecules in the KMC simulations. The default is 1000.

Stepsize: indicates that the population of the species in a KMC run is printed every "Stepsize" reactions. The default is 10.

MaxEn: is the maximum allowed energy of the TSs for inclusion in the reaction network. The default is 40 kcal/mol when rate is canonical and it equals EKMC when rate is microcanonical.

PathInfo: can be used to run the KMC simulations only for the relevant paths. In that case, you have to run final.sh script and then

`PathInfo Relevant`

can be specified in the input file before running again kmc.sh.

***ImpPaths*:** is the minimum percentage of processes occurring through a particular pathway that has to be achieved in order to be considered relevant and finally plotted in an energy diagram (vide infra). If you want to plot them all use 0.

On the other hand, there are a number of available methods to bias the dynamics towards specific reaction pathways. So far, these are the available options:

1) In this algorithm, [4] selected bond lengths are not allowed to stretch more than 30% with respect to their initial values. This can be useful to prevent the breakage of certain bonds. This option can be used adding the following keyword in the CDS section:

```
nbondsfrozen 2
1 13
2 8
```

That would "freeze" two bond distances connecting atoms 1-13 and 2-8, respectively. The labels of the atoms must follow the line starting with ***nbondsfrozen***.

2) The second algorithm bias the dynamics towards a particular reaction mechanism. Suppose you want to get the H_2 elimination transition states from formic acid. There is an example of this option in path_to_program/examples/FA_biasH2.dat. You would also need FA.xyz file.

In this case, the reaction coordinate is composed of bond distances: 3-5, 1-4, and 4-5. While the first two bonds have to break, the last one has to form in this process. The following keywords should be added to the CDS section:

```
nbondsbreak 2
3 5
1 4
nbondsform 1
4 5
Kapparep  100
Kappa     100
rmin      0.5
iexp        1
irange     10
```

Keywords ***nbondsbreak*** and ***nbondsform*** follow the same syntax as ***nbondsfrozen*** above. The other keywords are explained below.

A similar test can be performed on the same molecule to get the H_2O elimination TS, whose input file is also available in path_to_program/examples/FA_biasH2O.dat.

Additionally, a Diels-Alder reaction has also been tested (ethylene+1,3-butadiene→cyclohexene) and. the input files can be copied from path_to_program/examples/diels_bias.dat and path_to_program/examples/diels.xyz.

In practice, a bias potential is added to the potential energy obtained using the semi-empirical Hamiltonian. The bias potential has the following simple form:

$$V = \sum_{i=1}^{nbondsbreak} V_{rep}(r_i) + \sum_{j=1}^{nbondsform} V_{attrac}(r_j)$$

$$V_{rep}(r_i) = \frac{\kappa_{rep}}{r_i}$$

$$V_{attrac}(r_j) = \frac{r_{min}^{iexp}}{2} \times \frac{\kappa}{r_j^{2\times iexp}} - \frac{\kappa}{r_j^{iexp}}$$

Where κ_{rep}, κ , r_{min} and $iexp$ are parameters (corresponding to the keywords ***Kapparep***, ***Kappa***, ***rmin*** and ***iexp***), and their units are such that the potential energy is in kcal/mol and the distances in Å. Additionally, the keyword (parameter) ***irange*** corresponds to the time window (in fs) employed by BBFS.[1] This parameter takes a default value of 20 fs when nonbiased simulations are performed.

The following table shows the parameters employed for the above examples:

Diels-Alder rxn		H_2 elimination from FA	H_2O elimination from FA
κ_{rep}	200	100	100
κ	200	100	100
r_{min}	0.5	0.5	0.5
$iexp$	1	1	1
irange	10	10	10

The units are such that the potential energy is in kcal/mol and the distances in Å.

The above examples can be tested using tsscds.sh script:

```
tsscds.sh inputfile
```

Should you try this second option, you have to optimize the parameters for your own system, perhaps starting from those collected in the above table.

7. Reaction mechanisms and kinetics using high-level calculations

The High-Level (HL) calculations are carried out by g09.

1. From your working directory (FA in the example), run:

```
(sbatch [options]) TS.sh molecule.dat
```

The default values in this case for a job submitted to slurm are:

#default sbatch FT2 resources
#SBATCH --time=04:00:00
#SBATCH -n 8
#SBATCH --output=tsscds_parallel-%j.log
#SBATCH -p shared
#SBATCH --qos=shared
#SBATCH --ntasks-per-node=2
#SBATCH -c 10

2. The following scripts needed to build the reaction network and solve the kinetics are the same as those described above for the LL calculations. Namely:

```
(sbatch [options]) IRC.sh
(sbatch [options]) MIN.sh
RXN_NETWORK.sh (allstates)
KMC.sh
```

Remember that the use of slurm involves checking that every script has finished before proceeding with the next one.

3. The product fragments are optimized using

```
(sbatch [options]) PRODs.sh
```

CAVEAT: Step 3 is mandatory before proceeding to step 4. Run step 3 only when you are sure the first 2 steps have been successfully completed and you do not need to add more transition states.

4. Finally, to make a summary of the calculations in folder FINAL_HL_FA:

```
FINAL.sh
```

References

1. Martínez-Núñez, E., An automated method to find transition states using chemical dynamics simulations. *J. Comput. Chem.* **2015,** *36*, 222-234.
2. Pietrucci, F.; Andreoni, W., *Phys. Rev. Lett.* **2011,** *107*, 085504.
3. Martínez-Núñez, E., An automated transition state search using classical trajectories initialized at multiple minima. *Phys. Chem. Chem. Phys.* **2015,** *17*, 14912-14921.
4. Martinez-Nunez, E.; Shalashilin, D. V., Acceleration of classical mechanics by phase space constraints. *J. Chem. Theor. Comput.* **2006,** *2* (4), 912-919.