FHI-aims cheat sheet including a micro-tutorial: Transition State Searching

Yingyu Yao and Luca M. Ghiringhelli

The chain of states methods (via Nudged Elastic Band, NEB [1] or String Method [2]) are powerful and established transition state searching tools when the initial and final states are known. Here we provide a fully loaded and easily adjustable package to run the "chain-of-states" methods with FHI-aims.

These methods work by evolving a chain of interconnected nodes, or images, on the potential energy surface (PES), and the result is a set of geometries describing a path such that the gradient of the PES is always parallel to the path – better known as the minimum energy path (MEP). Once the MEP is found, we can locate the transition state by finding the point of maximum energy along the path.

How to begin?

To run a MEP calculation, the following files are required.

- Two geometry files, one for the initial state and one for the final state. The atoms listed in the two files must be matched line by line, so that the nth atom in the initial geometry corresponds to the nth atom in the final geometry. When there are several different possible combinations, the user should use his chemical intuition to select and try the most probable configurations.
- An FHI-aims *control.in* configuration file is needed.
- A *chain.in* configuration file for the package. You will find plenty of examples shortly.

We have provided some test cases so you can familiarize yourself with the varies functionalities of the package. In order to quickly demonstrate the features of the pakage, Tutorials 1–4 adopt analytic potential energy surfaces (PESs), for which the MEP is exaclty known. This is done by substituting the call to FHI-aims (line beginning with `run_aims` in *chain.in*) with a call to a python script, in these cases *getf.py* that defines the PES and simulates the output of FHI-aims. In these cases, *control.in* and *geometry.in* are mere signposts.

For tutorials 1–4, please do the following.

1. Goto *tutorialx/run* and execute *job.sh*
2. Return to *tutorialx*
3. Execute *gnuplot plot.gnu* to view the result.

[1] K. W. Jacobsen H. Jonsson, and G. Mills. *Nudged Elastic Band Method for Finding Minimum Energy Paths of Transitions.* World Scientific, 1998.
G. Henkelman and H. Jonsson. *Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points.* J. Chem. Phys., **113**, 9978 (2000)

[2] W. E, W. Ren, and E. Vanden-Eijnden. *Simplified and improved string method for computing the minimum energy paths in barrier-crossing events.* J. Chem. Phys. **126**, 164103 (2007)

These tutorials will help you in establishing a visual understanding of the chain of states methods, and will help you in understanding the effects of various keywords. Feel free to adjust&test with your own settings.
In detail: Tutorial 1 is the most basic MEP searching configuration, with no additional functionalities.
Tutorial 2&3 demonstrates the use of climbing image(CI) to locate the saddle point(transition state) This method will move the image with highest energy toward a higher energy geometry along the MEP, thereby converging to the saddle point. Note that it is not recommended to reduce the number of images for CI, which is done here to make the process more visually apparent.
Tutorial 4 shows how to accelerate the calculation using previous knowledge. This could be a guess for the transition states, or a few image outlining the potential MEP. Try commenting out *external_geometry* in *chain.in* and compare the result.
Tutorials 5–7 show how the package work for realistic (elementary) molecular activated processes. Tutorial 5 for the inversion of ammonia molecule, and tutorial 6 for the exchange of a methyl group between two chlorine atoms.

For tutorial 5&6 please do the following.

1. Goto *tutorialx* and submit *job.sge*
2. Wait for several minutes. You can check the evolution of the MEP search by printing *forces.log*, where, at the end of each iteration, the current value of the residual forces of the system is appended. Note: The convergence criterion for the forces (in eV/Å) is specified by the keyword `force_thres` in *chain.in*.
3. Look at the result in the *optimized* directory. By printing the lines containing "Energy" in *optimized/path.xyz*, you easiy find the estimated maximum along the MEP.

Tutorial 7 is crucial for periodic systems. If you are only working with clusters you can safely ignore it.

1. Goto *tutorial7/sample7.1 & 7.2* and execute *interpchain.py*
2. Look at the *.xyz* file generated in the *interpolation* directory

The *interpchain.py* will generate the initial interpolation the same way as an actual calculation.

The difference in the two cases is that in the former sample the initial path is generated directly, while the latter is generated while considering periodic boundary condition. This can often be necessary when working with periodic system, and it is recommended that *interpchain.py* is used to check the initial interpolation prior to running.

What do we get?

The calculation will generate a few files in the working directory, and here are the more important ones.

`forces.log` and/or `climbing_forces.log`: They record the residual forces in the system for each iterations. This is the most straightforward file to check when inspecting a running calculation.
`./optimized/` directory: This is where the optimized path is put when the calculation is over. It will include all the images and their aims output, as well as a *.xyz* animation.
`./paths/` directory: This is where you can find information about specific iterations. If you find something strange about a particular iteration in the force log, you can find all the corresponding data here.

Attention!

These cheat sheets are not intended to substitute for reading the manuals of the programs involved and the original literature cited therein! Have fun!