

Potential of mean force (PMF)

We are going to determine the PMF between two argon atoms in an argon liquid. Argon is liquid at a temperature of 90 K.

PMF from an rdf

To do a 200 ps simulation of 200 argon atoms, go to the pmf directory and do:

```
grompp
```

```
mdrun
```

Make an rdf by doing:

```
g_rdf -s -bin 0.01
```

Have a look at the rdf, notice the oscillations

An rdf can be Boltzmann inverted to a PMF: $PMF(r) = -k_B T \log(g(r))$

In xmgrace go to Data → transformations → Evaluate expression

```
y = -0.75*ln(y+0.01)
```

where the 0.01 is to avoid the infinity of $\log(0)$

PMF from constraint simulations

For a PMF we need to determine the mean force at many different constraint distances. For that there is an extra moleculetype in `fe.top`, consisting of two argon atoms with a constraint between them.

Have a look at `fe.top`

You can see that the constraint distance for state A is 0 nm and for state B 1 nm. This means that the constraint distance will be equal to the coupling parameter λ .

We want to first create all input files and then send them into the grid to be computed. We must thus first make a simulation to create the suitable input coordinates for the different λ values. This is done using the “slow growth” method, which should not be used for calculating the actual free energy profile, because the fixed λ simulations at chosen λ values is a superior approach.

We will run a short simulation where λ is varied from the initial value to the final value with short steps and the system geometry is written down at suitable intervals to be used as the starting point at that λ value.

To do this have a look at `make_inp_gro.mdp` and observe the total `nsteps`, `nxtcout`, and `delta-lambda`. These values should result in 11 frames in the trajectory file.

```
Run grompp -c conf.gro -f make_inp_gro.mdp -p fe.top
```

and then

```
mdrun
```

You can have a look at the λ vs. simulation time with e.g. `xmgrace`

```
xmgrace dgdl.svg
```

The next step is to prepare the input files for the individual simulations at fixed λ values. First we convert the trajectory file into individual *.gro files with `trajconv` (you can have a look at the other uses of it by giving `trajconv -h`):

```
trajconv -f traj.xtc -sep -o init.gro
```

The actual input files for the simulations (*.tpr files) can be done using the script `make_inp_tprs.tcsh`. Have a look at the script. It includes also some documentation.

```
run ./make_inp_tprs.tcsh
```

The script creates a subdirectory for each λ value and a `topol.tpr` file in each of them. You can run these manually, sequentially, through a queuing system in a local machine, or distribute them to be executed on grid. To do the local sequential runs, you can have a look at another exercise at <https://extras.csc.fi/chem/courses/gmx2007/pmfexc.tar.gz>

To use the following scripts for Grid submission you need valid credentials to run the jobs in Nordugrid, see *e.g.* the Grid-tutorials on the workshop main page):

“log in” to the grid: `grid-proxy-init`

For more details on initialising Grid-resources, for MGrid see

http://www.csc.fi/english/research/Computing_services/

`grid_environments/mgrid/getting_started`

To submit the jobs use the following script:

```
DO_LAM_arc1_create_jobs_and_submit.csh
```

There is another script that monitors how many jobs are still in execution and how many are ready: `DO_LAM_arc2_monitor_resubmit_until_done.csh` and once all jobs are done, another script to retrieve the results from the Grid: `DO_LAM_arc3_fetch_results.csh`

After the results have been obtained and stored in a subfolder of the original job-folder. You can use the following script (`GET_MF`) to calculate the mean constraint force and an error estimate for each λ simulation and combine them into one file:

```
Run GET_MF by typing ./GET_MF 5 (start at 5 ps)
```

Have a look at the mean force by opening `mf.xvg` in `xmgrace`

The mean force can be integrated with the script `integrate_mf.xvg`

First do it without entropy correction by doing:

```
./integrate_mf.awk mf.xvg 0 > pmf0.xvg
```

Do it with entropy correction by using the correct temperature:

```
./integrate_mf.awk mf.xvg 90 > pmf.xvg
```

Have a look at both PMF's in `xmgrace` and notice the effect of the entropy correction. Compare the PMF from the constraint runs with the one obtained from the `rdf`.

If you are unhappy with some of the individual error bars in the $\frac{\partial H}{\partial \lambda}$ you can continue the simulation for that *lambda* point to improve accuracy.