3rd Tutorial on MD - Introduction to LAMMPS and Diffusion of Polymers

We aim to get started with the MD simulation software LAMMPS (short for Large-scale Atomic/Molecular Massively Parallel Simulator), which is free and open-source. It contains a huge range of functionality for simulations (potentials, thermostats, barostats, output formats, data analysis tools, etc.) and can perform MD simulations much faster than our self-written code. You also receive a couple of example scripts, which we will use today, with the exercise materials (zip-file).

Installation of LAMMPS

At first, go to *http://lammps.sandia.gov* and download the latest stable version as a "tarball" - this might take a few minutes. Next, we need to compile LAMMPS. For this, move the file `lammps-stable.tar.gz` into a folder of your preference and unpack it. In the console, type `tar -xvf lammps-stable.tar.gz` and then go into the folder `src/`, which contains the Makefile. Enter `make` to see the possible options for compiling and enter `make package` to see the available packages (bundles of functions and extensions) of LAMMPS. To simulate polymer chains we need to install the package `molecule`, so for this enter `make yes-molecule` to make this setting. Then, finally enter `make serial` to compile the program. This may take a while, again.

After compiling has finished, you will find a file named `lmp_serial` in the same folder `src/`. This is the executable program, which can run only serial LAMMPS jobs, i.e. only jobs using a single CPU (no multithreading). We will next consider some examples to start learning how to use the program.

Mean squared displacement and diffusion coefficient

To understand and investigate diffusion processes on a molecular level, it is useful to consider the mean squared displacement (MSD), an observable defined as

$$\langle \Delta^2 r(t) \rangle = \frac{1}{N} \sum_{i=1}^{N} \langle (\vec{r}_i(t) - \vec{r}_i(0))^2 \rangle,$$

where $\vec{r}_i(t)$ is the position of a molecule (or monomer) i at the time t in the fluid with fixed N and V. The angular brackets denote an ensemble average, i.e. over several different starting conditions. Furthermore, the MSD of a fluid typically behaves like a power law of the time, i.e.

$$\langle \Delta^2 r(t) \rangle \sim t^{\alpha},$$

while the exponent α depends various conditions. In an n-dimensional space the MSD of is related to the instantaneous (or time-dependent) diffusion constant by

$$D(t) = \frac{1}{2n} \frac{\partial \langle \Delta^2 r(t) \rangle}{\partial t}.$$

MD Simulations with LAMMPS

To get started with using LAMMPS, we want to consider some example scripts to measure the MSD and diffusion constant of an (example) ideal gas, a Lennard-Jones gas and a polymer in solution.

Ideal gas

You receive a commented LAMMPS script `in.2d_ideal` (in the folder `2d_ideal`) to simulate a 2-dimensional ideal gas. An equilibrated starting configuration is stored in `2d_ideal.data`. Get an overview over what the script contains and does. You can run the script by entering in the console `*filepath*/lmp_serial < in.2d_ideal,` where `*filepath*` is the path `to the src`-folder of your LAMMPS installation.[1]

1.) Consider first what the exponent α of the time-dependence of the MSD will be.

2.) Next, run the simulation. An output file `msd.txt` will be filled with a measurement of the MSD. Write a Python-Script - or in any language you prefer - to plot the resulting MSD $\langle \Delta^2 r(t) \rangle$ as a function of time on double-logarithmic axes and to extract the diffusion constant $D(t)$.

Optional: Plot the velocity histogram and compare it to the Maxwell-Boltzmann distribution in two dimensions.

Lennard-Jones gas

3.) What time-dependence (exponent α) do you expect to find for a LJ gas and why?

4.) Next, copy the script `in.2d_ideal` and `2d_ideal.data` into the folder `2d_lj`. Uncomment the suitable parts of the script to make LAMMPS run a simulation of a Lennard Jones gas.

5.) Run the script to perform the simulation. Again, plot the $\langle \Delta^2 r(t) \rangle$ and $D(t)$ and compare with your prediction.

Optional: Plot again the velocity histogram and compare it to the Maxwell-Boltzmann distribution in two dimensions.

Polymer chains in solution

6.) Go into the directory `2d_chain` which contains a script `in.2d_chain` and starting configuration `2d_chain.data` for a chain with $N = 40$ monomers in an implicit solvent. Inspect the script and the starting configuration and identify the changes. What information does LAMMPS need to simulate polymer chains?

7.) What time-dependency of the MSD to you expect now?

8.) Run the simulation and plot $\langle \Delta^2 r(t) \rangle$ and $D(t)$. What do you find?

[1] Alternatively, if you copy `lmp_serial` into your folder, you can execute the simulation by `./lmp_serial < in.2d_ideal`