

MDANSE 2018 SCHOOL TUTORIAL
Analysis of MD trajectories using MDANSE
(Miguel A. Gonzalez, September 2018)

1. Introduction

In this tutorial, we will see how to use MDANSE to convert and analyse classical MD trajectories obtained with DL_POLY, LAMMPS and Gromacs.

2. Tools used

MDANSE (v. 1.2.0)¹: Package for the analysis of MD simulations.

A few scripts in python to perform simple analysis and plots.

LAMP² to fit simulated $I(Q,t)$ and $S(Q,\omega)$.

3. MDANSE

MDANSE is a python application maintained and developed at ILL by Eric Pellegrini and Remi Perenon and designed for computing properties that can be directly compared with neutron scattering experiments, such as the coherent and incoherent intermediate scattering functions and their Fourier transforms, the elastic incoherent structure factor, the static coherent structure factor or the radial distribution function. It can also compute quantities such as the mean-square displacement, the velocity autocorrelation function as well as its Fourier Transform (the so-called vibrational density of states) enlarging the scope of the program to a broader range of physicochemical properties.

¹ Freely available from www.mdanse.org

² Freely available at www.ill.eu/users/support-labs-infrastructure/software-scientific-tools/lamp/download-links/

4. Case 1: Argon near the triple point

The file `md_files.tar.gz` contains the results of an NVE simulation done with DL_POLY for liquid argon near the triple point.

Simulation details:

864 atoms

Cubic box, $L = 3.431415 \text{ nm}$: $\rho = 1.4185 \text{ g/cm}^3$

$T \approx 86 \text{ K}$

Reduced values: $T^* \approx 0.73$, $\rho^* = 0.8442$

Simulation time = 500 ps

25001 frames saved every $\Delta t = 0.02 \text{ ps}$

Tasks:

1. Extract the simulation files: `> tar xvf md_files.tar.gz`
2. Examine the input files (CONFIG, FIELD, CONTROL)
3. Examine the file OUTPUT. Check average temperature, pressure, energy and fluctuations.
4. Use DL_POLY GUI: `> dl-gui (Analysis / Statistics)` to check the evolution of the thermodynamic quantities during the simulation. You can also try some of the analysis available there.
5. Launch MDANSE and convert the trajectory (File / Trajectory converters / DL-POLY). You can do everything from the GUI, but a good practice consists in saving each analysis before running it. This will save a python script that can be used both as a reference for the parameters used in the analysis and to redo the analysis without launching the GUI.
6. Check the types of analysis available and compute the main quantities:
 - Radial distribution function
 - Static structure factor
 - Mean square displacement
 - Velocity autocorrelation function
 - Density of states
 - Incoherent scattering function: $I_{\text{self}}(Q,t)$ and $S_{\text{inc}}(Q,\omega)$
 - Coherent scattering functions: $I(Q,t)$ and $S_{\text{coh}}(Q,\omega)$
 - Current correlation functions (transverse and longitudinal)
7. Compare these results to MD and experimental results published in some classical papers (given in file `ref_dat.tar.gz`).
8. Check the consistency of the values of the self-diffusion coefficient obtained from the mean square displacement, the velocity autocorrelation function, the self-intermediate scattering function and the incoherent dynamical structure factor.

5. Case 2: Rigid water (TIP4P/2005 model)

The file md_files.tar.gz contains the results of an NVT simulation done with LAMMPS for liquid water using the rigid TIP4P/2005 model.

Simulation details:

360 water molecules

Cubic box, $L = 2.2124$ nm: $\rho = 0.9945$ g/cm³

$T \approx 298$ K

Simulation time = 100 ps

1001 frames saved every $\Delta t = 0.1$ ps

Tasks:

1. Extract the simulation files: `> tar xvf md_files.tar.gz`
2. Examine the input files (waterTIP4P2005_360mol.in and .data)
3. Examine the output log: waterTIP4P2005_360mol.log. You can use the script lammps_rms.py to get the average value of the thermodynamic properties (T, P, V, energy components) and their fluctuations.
4. Launch MDANSE and convert the trajectory (File / Trajectory converters / LAMMPS). **Your first try will fail!** The reason for this is that MDANSE uses the masses given in the data file in order to identify the atoms that compose the system. If they are not the same within a given tolerance as the masses contained in its database, then MDANSE will not be able to assign some atoms and the conversion will fail. You can play with the mass tolerance (the default tolerance is 10^{-5}) to avoid this, but if the tolerance is increased too much, then there will be several possible atoms or isotopes that are compatible with the mass given in LAMMPS file, making the conversion to fail again. The safest way to avoid this issue is to check which are the masses used by MDANSE and then edit the LAMMPS data file and ensure that the masses written there are the same. In the conversion interface, you also need to provide the exact time step used in the integration of the equations of motion during the MD run (declared with the keyword timestep) and how many frames you want to convert.

Another usual problem with LAMMPS trajectories is that they are not ordered by “atom number”, and the order in which the atoms are written may even change from one frame to another. You can use the “dump_modify group sort id” in the input file to LAMMPS (give a look to the file waterTIP4P2005_360mol.in) to ensure that the initial atom order is preserved.

5. Try to compute the following properties:

Radial distribution function:

MDANSE will calculate the partial radial distribution functions (here OO, OH, and HH)

$$g_{\alpha\beta}(r) = \frac{\rho_{\alpha\beta}(r)}{\langle \rho_{\alpha\beta} \rangle}$$

and the total $g(r)$:
$$g(r) = \frac{\sum_{\alpha\beta}^n c_{\alpha} c_{\beta} w_{\alpha} w_{\beta} g_{\alpha\beta}(r)}{\sum_{\alpha\beta}^n c_{\alpha} c_{\beta} |w_{\alpha}| |w_{\beta}|}$$

where c_{α} and c_{β} are the relative concentrations of species α and β , and w_{α} and w_{β} their respective weights selected from the available list of atomic properties (typically `b_coherent` or `proton` for this type of analysis in order to compare to neutron or x-ray experimental data).

The partial r.d.f. can be compared with the experimental ones given by A. K. Soper (table 5 in *ISRN Physical Chemistry*, 279463 (2013), which are provided in file `soper13_rdf.dat`.

You can use the Export option in the graphics window of MDANSE to export the 3 computed partials as ASCII files and then use your preferred plotting program to compare simulation and calculation. Or otherwise use directly the python scripts provided: `compare_pdf_oo.py`, `compare_pdf_oh.py`, and `compare_pdf_hh.py`.

They read directly the netCDF output generated by MDANSE, but to do this they require the python-netcdf4 module. So first install it as:

```
> sudo dpkg -I python-netcdf4_1.2.2-2_amd64.deb
```

```
> sudo apt-get install -f
```

And then you can run: `> python compare_pdf_oo.py`

Static structure factor:

MDANSE computes the partial $S(Q)$'s as the Fourier transform of the partial $g(r)$, corresponding to the Faber-Ziman definition:

$$S_{\alpha\beta}(Q) = 1 + \frac{4\pi\rho_0}{Q} \int_0^{\infty} r [g_{\alpha\beta}(r) - 1] \sin(Qr) dr$$

The total $S(Q)$ is computed as a weighted sum similar to the one used for the total $g(r)$. In the case of the analysis 'X-ray Static structure factor', the Q-dependence of the atomic form factors is taken into account in this weighted sum.

Again Soper has provided experimental data (table 4 in *ISRN Physical Chemistry*, 279463 (2013), given in file `soper13_fx.dat`). Here a source of confusion is that the data can be normalized in different ways (see Soper's paper). Using the normalization II in that reference we have that:

$$\begin{aligned} D_x(Q) &= \frac{\sum_{\alpha\beta\geq\alpha} (2 - \delta_{\alpha\beta}) \times c_{\alpha} c_{\beta} f_{\alpha}(Q) f_{\beta}(Q) [S_{\alpha\beta}(Q) - 1]}{\sum_{\alpha} c_{\alpha} f_{\alpha}^2(Q)} \\ &= [S(Q) - 1] \times \frac{\sum_{\alpha\beta} c_{\alpha} c_{\beta} f_{\alpha}(Q) f_{\beta}(Q)}{\sum_{\alpha} c_{\alpha} f_{\alpha}^2(Q)} \end{aligned}$$

Where $S(Q)$ would be the static structure factor (going to 1 at large Q) computed by MDANSE. Therefore even after using MDANSE we should recalculate the x-ray observable using the atomic factors. Here to make a quick comparison just try:

```
> python compare_sq_xr.py ,
```

But this script neglects the Q-dependence and approximate the second term by a constant factor of 0.5, so remember that it is not completely correct!

To compute the typical $S(Q)$ measured with neutrons using D_2O , you need to select the H atoms, transmute them to D, and then run the static structure factor analysis.

Now the experimental data provided by Soper (table 3 in *ISRN Physical Chemistry*, 279463 (2013), given in file `soper13_Dn.dat`) correspond to the quantity:

$$D_n(Q) = I_n(Q) - \sum_{\alpha} c_{\alpha} \langle b_{\alpha}^2 \rangle = \sum_{\alpha\beta} c_{\alpha} c_{\beta} b_{\alpha} b_{\beta} \times [S(Q) - 1]$$

For D_2O , the sum $\sum_{\alpha\beta} c_{\alpha} c_{\beta} b_{\alpha} b_{\beta} = 0.4072 \text{ barn/sr/atom}$, so we need to use this scaling factor to compare the output with the experimental values. The script `compare_sq_d2o.py` does this.

The $S(Q)$ computed for H_2O does not tend to 1, but to 0.016. This is due to the use of the modulus in the denominator of the weighting expression, while the negative scattering lengths for H are used in the numerator³: $S(Q) = \frac{\sum_{\alpha\beta}^n c_{\alpha} c_{\beta} w_{\alpha} w_{\beta} S_{\alpha\beta}(Q)}{\sum_{\alpha\beta}^n c_{\alpha} c_{\beta} |w_{\alpha}| |w_{\beta}|}$.

The sum in the denominator is 0.1960 barn/sr/atom, so now in order to compare to experiment we can do: $D_n(Q) = 0.1960 \text{ barn/sr/atom} \times \{S(Q) - 0.016\}$

This is done in the script `compare_sq_h2o.py`.

Mean square displacement

Then determine the self-diffusion coefficient from the m.s.d. (`msd_analysis.py`)

Velocity autocorrelation function

Density of states

Incoherent scattering functions: $I_{\text{self}}(Q, t)$ and $S_{\text{inc}}(Q, \omega)$

Compare your results to the neutron and MD data published by Qvist (*J. Chem. Phys.* **134**, 144508 (2011)). If you have time you could try to fit your simulated curves with the same model employed by Qvist et al. and check if you reproduce their results.

Coherent scattering functions: $I(Q, t)$ and $S_{\text{coh}}(Q, \omega)$

6. Case 3: Flexible water (TIP4P/2005f)

The file `md_files.tar.gz` contains the results of an NPT simulation done with Gromacs for liquid water. It employs the flexible version of the TIP4P/2005 model, using the input files available in www.sklogwiki.org/SklogWiki/index.php/GROMACS_files_for_the_TIP4P/2005f_model.

Simulation details:

500 water molecules

Cubic box, L (at start) = 2.46394 nm: ρ (at start) = 0.9999 g/cm³

³ This was done to avoid the large oscillations that appear when terms in the sum in the denominator cancel to a large extent giving a very small number. However this is confusing and it will probably change in the future.

$T \approx 298 \text{ K}$, $P \approx 1 \text{ bar}$

Simulation time = 20 ps

1001 frames saved every $\Delta t = 0.02 \text{ ps}$

Tasks:

1. Extract the simulation files: `> tar xvf md_files.tar.gz`
2. Examine the input files (TIP4P2005f-N500.mdp, top and .gro)
3. Examine the output log: TIP4P2005f-N500.log. Check average temperature, pressure, energies and box size.
4. Gromacs provides a large number of exploration and analysis tools. For example:
`> gmx traj -f TIP4P2005f-N500.xtc -s TIP4P2005f-N500.tpr -ob box.svg` will print the box size to the file box.svg. This is a Grace file that can be directly plotted using: `> xmgrace box.svg`
Or the command `> gmx msd -f TIP4P2005f-N500.xtc -s TIP4P2005f-N500.tpr -o msd.svg` will compute the mean square displacement.
And there are many more analysis available⁴.

5. Launch MDANSE and convert the trajectory (File / Trajectory converters / Gromacs)⁵. You need a PDB file to convert a Gromacs trajectory. As there is not a PDB among the files provided, you will need to generate one. You could use Avogadro for this, but you can also use one of the Gromacs utilities:

```
> gmx editconf -f TIP4P2005f-N500.gro -o TIP4P2005f-N500.pdb
```

However the generated pdb does not match exactly the required format, so **MDANSE will fail with an error “Atoms/m not found” or something similar**. This can be easily solved by editing the pdb file and doing the following replacements (respect the spaces in order to keep the remaining columns unchanged):

```
'OW1' --> 'O '  
'HW2' --> 'H1 '  
'HW3' --> 'H2 '  
'MW4' --> 'MW '
```

6. You can run the most common analysis and compare to the results obtained before in order to see if there is any appreciable difference between the rigid and flexible models.

The directory Case3b contains a longer trajectory of 100 ps, with a frame saved every 0.1 ps in order to be able to compare similar trajectories for both cases.

⁴ Check <http://manual.gromacs.org/programs/bytopic.html>.

⁵ Caution: In MDANSE 1.2.0 the times are incorrectly taken in fs, instead of ps! This has been corrected in MDANSE 1.2.1.

7. Case 4: Rigid water (TIP4P/2005)

The file md_files.tar.gz contains the results of an NVT simulation of TIP4P/2005 water. This is very similar to the system studied in Case2, but now we employed DL_POLY to do the simulation, the system is slightly larger and a longer simulation was run (1 ns).

Simulation details:

500 water molecules

Cubic box, $L = 2.46642 \text{ nm}$; $\rho = 0.9969 \text{ g/cm}^3$

$T \approx 298 \text{ K}$

Simulation time = 1000 ps

1001 frames saved every $\Delta t = 1 \text{ ps}$

Tasks:

You can use the trajectory files to repeat the previous analysis. **You will encounter a new problem when trying to convert the trajectory in MDANSE, as here the pseudo-site of the TIP4P2005 model has been called Q.** However there is not a Q atom in the database, so you will need to use the atom aliases box to tell MDANSE how to interpret unknown atom keys. Thus, in this case, you would give {'Q': 'Du'} to tell MDANSE that this is a dummy atom, which exists in the database.

As the atom keys used in DL_POLY can be ambiguous, it is convenient to check always that the elements have been correctly identified. For example, in many cases, the names 'HO' and 'CA' are used to name a hydrogen in an OH group and an aromatic carbon. However, they will be interpreted by MDANSE as holmium and calcium, respectively. In this case, using the atom aliases box to give {'HO': 'H', 'CA': 'C'} would solve the issue.

8. Case 5: Flexible benzene (OPLS2005)

The file mdnpt_files.tar.gz contains the results of an NPT simulation of phase I of crystalline benzene. The simulation box is orthorhombic and contains $3 \times 3 \times 3$ unit cells. The purpose of this first simulation is to equilibrate the system and check that the crystal is stable and the lattice parameters given by the potential used in the simulation are reasonably close to the experimental ones ($a = 7.243 \text{ \AA}$, $b = 9.31 \text{ \AA}$, and $c = 7.656 \text{ \AA}$ at 296 K, 0.3 GPa⁶).

Simulation details:

108 water molecules

Orthorhombic box: $L_x = 2.1729 \text{ nm}$, $L_y = 2.7930 \text{ nm}$, $L_z = 2.0268 \text{ nm}$.

$T \approx 296 \text{ K}$, $P \approx 3 \text{ kbar}$

Simulation time = 100 ps

Tasks:

⁶ A. Budzianowski and A. Katrusiak, *Acta Cryst. B* **62**, 94-101 (2006)

Control the evolution of T, P and V and compare the lattice constants by the end of the run with the experimental values.

The file `mdnvt_files.tar.gz` contains the results of an NVT simulation started with the last configuration of the NPT run. The system was equilibrated in the NVT ensemble for 50 ps and then a simulation run of 1000 ps was done. During the production run, 1001 frames were saved every 1 ps.

Tasks:

Verify that temperature and pressure have remained close to the desired values (296 K, 3 kbar).

Convert the trajectory. **You will have to use the alias option to make MDANSE understand that CA means a carbon!**

Check the structure using the pair distribution function analysis. Do you see some evidence of long-range order?

Compute also the static structure factor, $S(Q)$. It shows clearly more features than the $S(Q)$ of a liquid (for example, you can compare it with the $S(Q)$ of water, or even better, make a simulation for liquid benzene and compare both). However, there are no clear Bragg peaks, as we could expect. Why?

Compute the mean square displacement. How does $u^2(t)$ evolve with time? There is a plateau appearing, but only at large times of the order of a few hundred ps and showing quite a large value, about 12 \AA^2 . Can you imagine why?

To check that we still have really a crystal and that no diffusion (even if small) is taking place in our simulation, re-compute the mean square displacement, but using the Group coordinates option and selecting by molecule. This will calculate the m.s.d. of the centre of mass of each individual molecule, as well as the average of all of them. As you can see, there are no signs of molecular diffusion and the amplitude of motion of the centres of mass is very small. Therefore we can conclude that we have a stable crystal.

Compute also the coherent and incoherent $S(Q, \omega)$, as well as the elastic incoherent structure factor (EISF). You can try to fit the calculated EISF to the available QENS theoretical models for rotation. Which one would you expect to work better?

As a longer exercise, you could try to extract the EISF from the computed $S_{\text{inc}}(Q, \omega)$, following the same procedure that you would use to analyze real experimental data. Note that there can be some subtleties when comparing simulated and measured EISFs obtained through different routes⁷.

The directory Case5b contains a shorter trajectory of 40 ps, with 2001 frames saved every 0.02 ps and writing both positions and velocities in the HISTORY file. Use this trajectory to compute the vibrational density of states.

⁷ See N. D. Morelon et al., *J. Chem. Phys.* **109**, 2883 (1998)

9. Case 6: Rigid benzene (OPLS2005)

This is entirely analogous to the previous case, but using a rigid body to represent the benzene molecule. You can repeat the same analysis and determine the influence of including or not benzene's intramolecular degrees of freedom in our simulation.

As before, the directory Case6b contains a shorter trajectory of 40 ps, with 2001 frames saved every 0.02 ps and writing both positions and velocities in the HISTORY file. As now we are using a rigid model, only the low-frequency intermolecular modes should be visible.

