Exercise_3 Molecular Dynamics Simulation

Andrea Scaioli

May 2025

1 Introduction

In this exercise we deal with water models and temperature couple algorithms. A water model allows us to take into account the effect of the solvent to the system in MD simulations in order to perform better analysis, while the temperature couple algorithms play the role of thermostats, maintaining the temperature constant during the simulation. These algorithms simulate the effect of heath bath and are essential for ensembles like NVT.

The most famous temperature couple algorithms are Berendsen and Nose-Hoover. The Berendsen thermostat rescales velocities to a factorλ in order to maintain the temperature constant $\lambda = \sqrt{1 + \frac{\Delta T}{\tau_T}\left(\frac{T_0}{T} - 1\right)}$. The more the system deviates from the target temperature, the more the velocities are adjusted. The temperature evolves according to $\frac{dT}{dt} = \frac{T_0 - T}{\tau_T}$ where τ_T is the coupling constant.This algorithm is very quick and is efficient at quickly relax the system to the desired temperature, but does note generate the correct canonical ensemble, the system lacks of natural temperature fluctuations, the kinetic energy is too narrow and is not suitable for computing ensemble averages.

The Nose-Hoover thersmostat is based on an exetend hamiltonian that includes a fictitious degree of freedom, a thermostat variable that acts as a bath. The system exchange energy with this thermostat variable acting as a bath. In this way the system change energy with the heat bath, leading to proper canonical ensemble. Mathematically the Newton's equations become: $a = \frac{F}{m} - \xi v$ and $\frac{d\xi}{dt} = \frac{1}{Q}\left(\frac{2K}{N_f K_B T} - 1\right)$ where ξ is a friction coefficient, Q is the thermostat mass parameter, K is the kinetic energy and N_f is the number of degrees of freedom. On the one hand, his thermostat has the advantage to generate the correct canonical ensemble. On the other hand, it is slower to equilibrate and can exhibit oscillations if not well tuned because is more sensitive to parameters choices.

2 Methods

The time series of temperature, kinetic energy, potential energy, total energy and heat capacity has been computed in all of the six simulation performed in this exercise. Each simulation differes from the other for the thermostat or for the water model used.

In every simulation 523 water molecules are simulated using SPC or TIP3P water model coupled with an NVE, Berendsen or Nose-Hoover thermostat. SPC and TIP3P models are three site rigid water models that descrybe water with three atoms, one oxygen and two hydrogens. The bonds and the angles of the molecules are fixed and point cherges and LJ interactions are present to mimic real's water behaviour.

THey differ in the values of the parameters aformetioned, TIP3P reproduces dielectric constant better while SPC gives slightly better density.both overestimates diffusion.

To perform the simulations we ran the following commands:

- `grompp` and `mdrun` to minimize water boxes (to perform to each water model)
- `grompp` and `mdrun` to minimize the thermostats (to perform for each thermostat and each water model)
- `grompp` and `mdrun` to execute the simulation (to perform for each thermostat and each water model)
- `energy` to gain energies temperatures and heat capacity for each simulation

3 Results

In Figure 1 are shown the time series for potential, kinetic and total energy and temperature obtained for each simulation ran with different thermostat and water models.
From the plot is evident that Berendsen thermostat with SPC and TIP3P water models perform similar results as Nose Hoover thermostat with SPC water model.
The Nose Hoover with the TIP3P water models produces slighty higher energy values, while le NVE ensemble is not suitable for such simulations because energy cannot fluctuate.
Berendsen simulations are similare because velocities are rescaled in order to maintain the temperature constant. Differently, The Nose Hoover thermostat exchanges energy with a fictitious variable so it depends more on the water model adopted. The SPC's water molecules are slightly less polarized compared with the Nose Hoover ones and this should be the reason for the energy variation between the two Nose Hoover simulations.

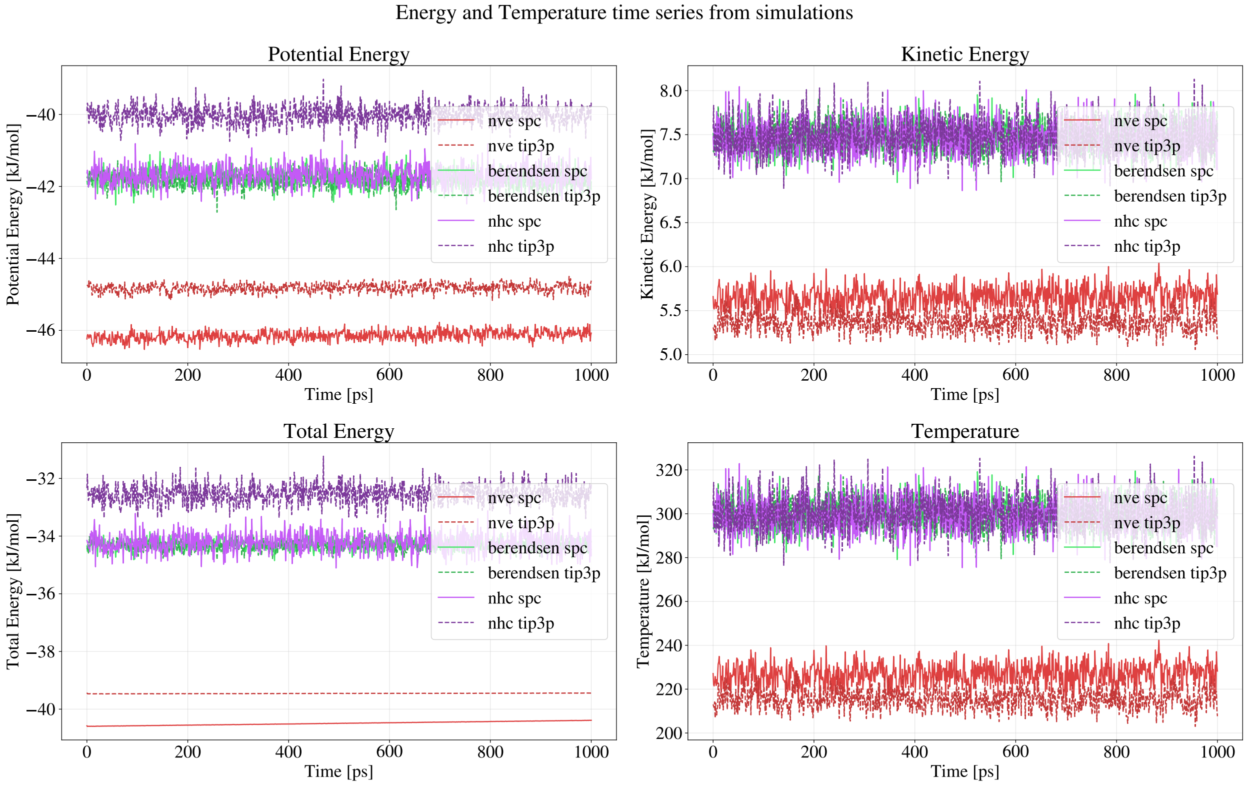

Figure 1: Potential, kinetic, total energy and temperature values for the 6 different simulations. Is Evident that Nose Hoover spc, Berendsen spc and Berendsen tip3p are always similar.
The Nose Hoover tip3p has slightly higer values for the energies while for temperatures is similare to the other three combination.
The NVE simulation are the worst ones, there are no energy fluctuations and the temperature is very underestimated.

To understand better the thermostats is useful to compare the heat capacities computed from each simulations. The heat capacity can be used as an indicator of the quality of the different simulations because measure the quality of the canonical ensemble generated depending on the energy's fluctuations. As aforementioned the only thermostat that generates a good canonical ensemble is the Nose Hoover.

Heat Capacities [J/(mol K)]

	NVE	Berendsen	Nosé–Hoover
SPC	4.41	18.35	75.46
TIP3P	0.081	17.95	71.69

Table 1: Heat capacities' values for the six different simulations. Is evident that the only values correct are with the ones computed with Nose Hoover thermostat that are closer to the literature value $C_v = 74.4$.

4 The Flying-Icecube-Effect with a Polymer Model

The Flying Ice Cube Effect refers to the unphysical accumulation of kinetic energy in low-frequency modes (such as translation and rotation of the entire system), while high-frequency modes (like bond vibrations) are increasingly dampened over time. This effect violates the equipartition theorem, which states that each degree of freedom should have the same average energy in thermal equilibrium.

The two differences between the `polymer.mdp` used in this exercise and the file used in the last session are the integrator (md vs sd) and the temperature coupling (berendsen vs none). The Flying Ice Cube Effect is caused by the velocity rescaling in the berendsen thermostat. Some frames of this effect are shown in Figure 2.

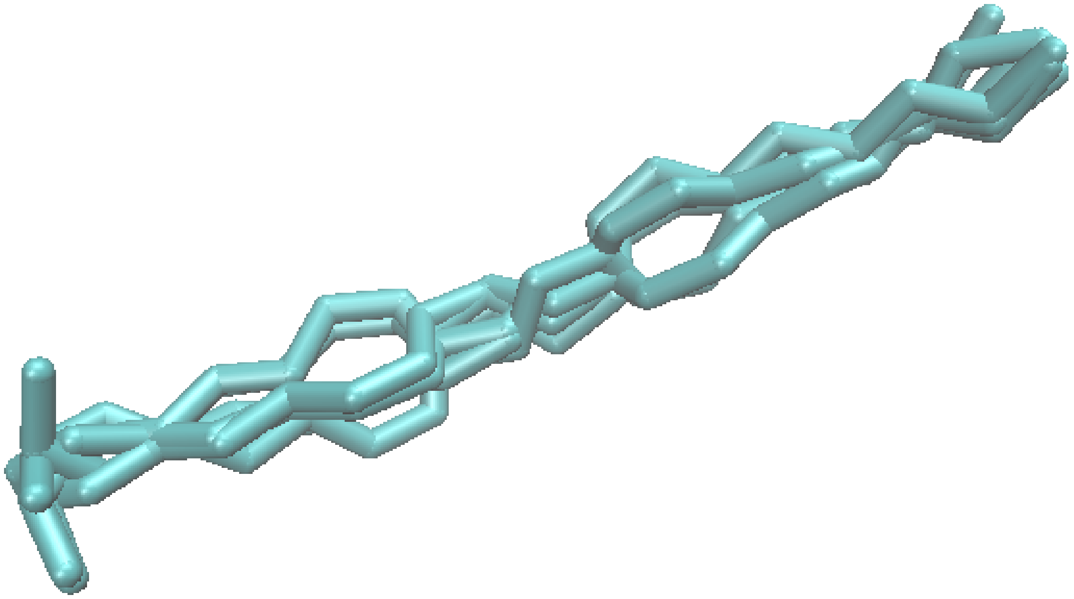

Figure 2: These figure shows the last five frames of the simulation. It is clear that the polymer freeze and fluctuates non physically

5 Appendix A: berendsen.mdp commented

Here is presented the commented .mdp file for the berendsen thermostat:

```
title           = small water box


; Run parameters
integrator      = md        ; Molecular dynamics integrator (leap-frog algorithm)
nsteps          = 10000     ; Number of steps to simulate (total time = nsteps × dt)
dt              = 0.002     ; Time step in picoseconds (2 fs per step)

; Output control
nstxout         = 500       ; Write coordinates to trajectory every 500 steps
nstvout         = 500       ; Write velocities to trajectory every 500 steps
nstenergy       = 500       ; Write energies to energy file every 500 steps
nstlog          = 500       ; Write log file output every 500 steps


; Bond parameters
continuation    = no        ; Simulation does not continue from a previous run (new velocities generated )
constraint_algorithm  = lincs     ; Use LINCS algorithm to constrain bonds
constraints     = h-bonds   ; Only constrain bonds involving hydrogen atoms
lincs_iter      = 1         ; Number of LINCS iterations (1 is default)
lincs_order     = 4         ; Order of the expansion in LINCS (4 is usually sufficient)


; Nonbonded settings
cutoff-scheme   = Verlet    ; Use Verlet scheme for neighbor searching (for GPU support and better performance)
ns_type         = grid      ; Use a grid-based method to construct neighbor lists
nstlist         = 10        ; Neighbor list updated every 10 steps
rcoulomb        = 1.0       ; Coulomb (electrostatic) cutoff distance (in nm)
rvdw            = 1.0       ; van der Waals cutoff distance (in nm)
DispCorr        = EnerPres  ; Apply long-range dispersion correction to energy and pressure


; Electrostatics
coulombtype     = PME       ; Use Particle-Mesh Ewald method for long-range electrostatics
pme_order       = 4         ; Interpolation order for PME (4 is typical, i.e. cubic interpolation)
fourierspacing  = 0.16      ; Grid spacing for PME Fourier mesh (smaller → more accurate, but slower)


; Temperature coupling is on
tcoupl = Berendsen ; Use Berendsen thermostat to maintain temperature (not for runs due to poor ensemble sampling)
tc-grps      = system    ; Define temperature-coupling group (here the entire system is coupled as one group)
tau_t        = 0.1       ; Time constant (ps) for coupling (how quickly temperature is adjusted)
ref_t        = 300       ; Reference temperature (K) for the system


; Pressure coupling is off
pcoupl       = no        ; Do not use pressure coupling (i.e., constant volume simulation, NVT ensemble)


; Periodic boundary conditions
```

```
pbc          = xyz       ; Apply periodic boundary conditions in all three directions (x, y, z)


; Velocity generation
gen_vel      = yes       ; Generate initial velocities according to Maxwell distribution
gen_temp     = 300       ; Temperature for velocity generation (K)
gen_seed     = -1        ; Random seed for velocity generation (-1 means random seed based on system time)
```