# MD in NVE and NVT ensembles

**Implementing different thermostats**

Fatemeh Tabatabaei, Marcello Sega*

1. 12. 2010

ICP, Uni Stuttgart

This tutorial aims at providing knowledge about Molecular Dynamics simulation techniques in different ensembles. In the following there is a short introduction about physical quantities in different ensembles and after that, with some programming, the theoretical concept will be applied. The tutorial main topics are Microcanonical and Canonical ensembles.

## Molecular Dynamics in NVE and NVT ensembles

If a system is ergodic, average quantities computed along the trajectory generated by a MD simulation (i.e. time averages) at constant number $N$ of particles, volume $V$ and total energy $E$, are equivalent to ensemble averages in the microcanonical ensemble. Since in the thermodynamic limit (of a large number of particles), measurements taken in every ensemble are equivalent, one could naively think that a $NVE$ MD simulation is all what one needs to simulate any thermodynamic point. In practice, however, simulated systems are usually too small for this to happen, and even in the case when they are big enough, it can be complicated to find out which energy value $E$ is the correct one to simulate the desired thermodynamic point. If, for example, we are interested in a given density and temperature, it is easier to simulate the coupling to an external thermal bath (hence implementing a canonical, $NVT$ ensemble), than guessing the correct energy. In other cases, some of the system parameters might not be known *a priori*. Many algorithms have been devised to simulate different ensembles like the isothermal-isobaric NpT one (used when the density of the system is unknown) or the grand-canonical ensemble ($\mu VT$) (where the chemical composition of the system is unknown).

In this tutorial you will learn how to use two different implementations of a thermostat for MD simulations, simulating two different systems. The first system represents a barrier potential. In this system we exercise two methods for thermalizing the system and consequently obtaining constant temperature ($NVT$). The first one is called the Andersen thermostat, in which a constant temperature can be achieved by coupling the system to a heat bath and the second is called the Nosé-Hoover thermostat. In this method, the motion of the particle is coupled to a chain of thermostats. The equations of motion are integrated using an explicit time-reversible algorithm that might look a little bit complicated at first sight. These two methods will be explained more in detail in the later parts of this tutorial.

The second system represents a Lennard-Jones liquid. With this system we exercise another method to obtain constant temperature, called Langevin thermostat. In summary, in this tutorial we discuss three different methods to sample the canonical ensemble.

## 1 Barrier Crossing

Consider the motion of a single particle in the following external potential ($\epsilon \geq 0$):

[*] **sega@icp.uni-stuttgart.de**

$$U(x) = \begin{cases} \epsilon A x^2 & x \leq 0 \\ \epsilon(1 - cos(2\pi x)) & 0 \leq x \leq 1 \\ \epsilon A(x-1)^2 & x \geq 1 \end{cases} \tag{1}$$

## Homework 1 (1 point)

1. As a first task, derive an expression for $A$, so that the force and its derivative to be continuous, and make a sketch of the energy landscape.

## 1.1 System Setup

A program is provided, which integrates the equations of motion of the particle starting at $x(t = 0) = 0$ using several methods while applying the "barrier potential" equation (1). First inspect the code `readdat.c`. This file contains all the information required for the generation of our system. A description of the input parameters is provided in the shell script `Run/run.sh`. You will have to change the setup parameters in that file in a later part of the tutorial.

The program provides the following methods including two different ensembles:

1. NVE ensemble which can be achieved with a "No thermostat" system.

2. The Andersen thermostat providing an NVT ensemble.

## 1.2 NVE ensemble

Using this method, a system with constant energy is provided. Try to predict what would you expect the phase space trajectories (i.e. curves in the 2D position-momentum space, in case of a single particle) to look like in the case of no thermostat.

## Homework 1 (1 point)

1. The integration steps are partially implemented in `integrate_nve.c`. Inspect the algorithm and make sure that you understand it. The code shuold be modified to update positions according to the Velocity-Verlet algorithm!

2. Compile the code and then change the path to folder `Run` and run the code launching `./run.sh` to get the data output for NVE ensemble.
   *Remark* : In order to switch the code for an NVE ensemble, change the value of "CHOICE" in the `run.sh` to `"1"`. The other "CHOICE" values for other methods are commented in `run.sh` as well.

## Homework 2 (1 points)

1. Set the temperature to $T = 0.05$. This is a low temperature and the system behaves like a harmonic oscillator. Explain why it is so.
   *Notice*: Think about the potential function.

2. Plot the phase space distribution. Explain why the phase-space distribution looks likes a circle.

3. Plot the phase-space trajectories at two other values of $T$ like $T = 1.5$ and $T = 4.0$. What do you observe? Explain.

## 1.3 Constant temperature ensemble

In the thermostat proposed by Andersen the system is coupled to a heat bath that keeps the system at the desired (average) temperature. The coupling to a heat bath is represented by stochastic impulsive forces that act occasionally on randomly selected particles. These stochastic collisions with the heat bath transport the system from one constant-energy shell to another. Between stochastic collisions, the system evolves at constant energy according to the normal Newtonian laws of motion. The stochastic collisions ensure that all accessible constant-energy shells are visited according to their Boltzmann weight.

All thermostats have in common that they transfer heat, sometimes momentum or angular momentum to or from the system. Obviously, the total energy then loses its property as a control parameter of the system (*i.e.* it is not anymore a conserved quantity). This means that it is no longer obvious whether our simulations is on the safe side on numerical grounds (Note that Nosé and Hoover formulated a method, the Nosé-Hoover thermostat, where the energy accounting of the heat bath is realized by one (or more) fictitious particle, which is propagated like the real particles of the system and this particle acts as a reservoir of the energy for the heat bath. Then, the total energy of the extended system is again a conserved quantity, and again a control parameter is available).

The integration of the Andersen thermostat in `integrate-and.c` is not implemented yet. You should implement the algorithm yourself according to the following steps:

1. Start with an initial set of positions and momenta $\{r^N(0), p^N(0)\}$ and integrate the equations of motion for a time $\Delta t$. For this you can use the Velocity-Verlet algorithm.

2. A particle is randomly selected to undergo a collision with the heat bath. The probability that a particle is selected in a time step of length $\Delta t$ is $\nu \Delta t$. Here $\nu$ is the collision frequency and specifies the strength of the coupling to the heat bath. Here you should use `RandomNumber()` from file `ran_uniform.c`.

3. If the particle has been selected to undergo a collision, its new velocity will be drawn from a Maxwell-Boltzmann distribution corresponding to the reservoir temperature $T$. For generating a Gaussian distributed random velocity look at the file `ran_uniform.c`. Use the function `RandomVelocity()` (To read more about the algorithm for Andersen thermostat see [1] ).

## Homework 3 (2 points)

1. Make plots for the phase space distribution at low temperature $T = 0.05$ for different values coupling parameters. What do you observe?

2. Compare the phase space distribution generated by the Andersen thermostat with the phase space distribution you obtained for $NVE$ ensemble in the last homework.

3. The relation between the reservoir temperature $T$ and the kinetic energy per particle is

$$k_B T = m \left\langle v_\alpha^2 \right\rangle, \tag{2}$$

   where $m$ is the mass of the particle and $v_\alpha$ is the $\alpha$-th component of its velocity. This is often used to measure the temperature in a microcanonical MD simulation. However, the condition of constant temperature is not equivalent to the condition that the kinetic energy per particle is constant. In a canonical ensemble of the finite system, the instantaneous kinetic temperature $T_k$ fluctuates. Running the code provides you the instantaneous kinetic temperature in every time step in `temp.dat`. Plot the data and explain what you see.

4. When running the code you will get the “Averaged temperature” as an output. Make a record of the average temperature for the two methods. Change the “NSTEP” parameter (number of time steps in the run) and see if any meaningful change happens towards getting a value closer to the reservoir temperature. Comment on this.

5. By changing the value of the temperature, investigate at which temperature the particle is able to cross the energy barrier in two methods. *Remark*: Crossing the barrier here means to find the density of the particle at the top of the barrier area to be non-zero.When you run the simulation you will get the density distribution output. So investigate the density distribution at different temperatures.

## 2 Lennard-Jones liquid

Now we consider the second system, a Lennard-Jones liquid (you should already be familiar with it from previous tutorials) that consists of $N$ particles located in a 3D cubic simulation box with edge length $L$, which interact with each other via the Lennard-Jones potential. In this case we employ the Langevin thermostat in order to simulate the canonical ensemble.

### 2.1 Langevin Thermostat

In the Langevin thermostat, at each time step every particle is subject to a random (stochastic) force and to a friction (dissipative) force. There is a precise relation between the stochastic and dissipative terms, which comes from the so-called fluctuation-dissipation theorem, and ensures sampling of the canonical ensemble. The equation of motion of particle $i$ is thus modified:

$$m\mathbf{a}_i = \mathbf{F}_i - m\gamma\mathbf{v}_i + \mathbf{W}_i\left(t\right) \tag{3}$$

Here $\gamma$ is a friction coefficient with units of $t^{-1}$ and $W_i$ is a random force that is uncorrelated in time and between particles, and is characterized by its variance:

$$\left\langle \mathbf{W}_i\left(t\right) \cdot \mathbf{W}_j\left(t'\right)\right\rangle = \delta_{ij}\delta\left(t-t'\right)6k_B mT\gamma \tag{4}$$

### 2.2 Velocity-Verlet for Langevin thermostat

The modified Velocity-Verlet algorithm for a Langevin thermostat is as follows

$$x_i(t+\Delta t) = x_i(t) + \Delta t v_i(t)(1-\Delta t\frac{\gamma}{2}) + \frac{\Delta t^2}{2m}G_i(t) \tag{5}$$

$$v_i(t+\Delta t) = \frac{v_i(t)(1-\Delta t\frac{\gamma}{2}) + \frac{\Delta t}{2m}(G_i(t)+G_i(t+\Delta t))}{(1+\Delta t\frac{\gamma}{2})} \tag{6}$$

where the $G_i$ is the total force: $G_i = F_i + \mathbf{W}_i$.

### Homework 4 (1 points)

1. Derive Eq. (6) using Eq (3). Remember that, according to the Verlet algorithm (See Lennard-Jones liquid tutorial), we have $v(t+\Delta t) = v(t) + \frac{a(t)+a(t+\Delta t)}{2}\Delta t$.

### 2.3 System Setup

A program is provided (in folder `program2`) which integrates the equation of motion for the $N$ fluid particles. There are two major elements in this code. First is the force initialization at each time step that adds in the random forces, $\mathbf{W}_i$. The second is a slight modification to update equations in the integrator to include the effect of $\gamma$.

### Homework

You should modify the integration algorithm in such a way that gives us a constant temperature according to the Langevin dynamics, according to Eqs .5,6. The updates for the velocities should be done in two half integration steps like in the usual Velocity-Verlet. After finishing the implementation for the integration compile the code.
*Remark: you can compile the code using this syntax :* `gcc -O3 -o lan_thermo  lan_thermo.c -lm -lgsl`

### Homework 5 (2 points)

1. To get a quick help run `./lan_md -h`. Then you find out which arguments correspond to the parameters. For example, to specify the number of particles and the number of steps use:
   `./lan_thermo -N 256 -ns 10000`.

2. When running the code you will get an output file `data.txt`: the first 5 columns are `time-step`, `time`, `potential`, `kinetic` and `total energy`. Integrate 10000 steps with $\gamma = 1.0$ and $\gamma = 0.0$ Plot the potential, kinetic and total energy for these two systems as a function of time in a single figure. Explain what do you see. Which of them are constant and which of them fluctuates? Why ?

3. Make a plot of the kinetic temperature (with $\gamma$=1.0). That can be done by plotting the 7th column of the data output. Explain what do you see.

Since you have performed MD simulations, you also obtained information about the dynamics of the system. The next task is to analyse the velocity autocorrelation function, $C(\tau)$, (VACF), and the mean square displacement $\Delta r^2(t)$ (MSD) which are defined as follows:

$$C(\tau) = \langle \vec{v}(t) \cdot \vec{v}(t+\tau) \rangle \tag{7}$$

$$\Delta r^2(\tau) = \langle |\vec{r}(t+\tau) - \vec{r}(t)|^2 \rangle \tag{8}$$

The VACF shows the timespan of the correlation between the velocities of a particle. This information may be useful in interpreting the shapes of MSD($\tau$) curves. At very small time intervals $\tau$ the motion is ballistic, while after more random kicks from neighbours, particles lose the memory of their original direction of motion which is manifested by VACF($\tau$) $\approx 0$ and the motion becomes diffusive.

In the `program2` folder you can find the `diffusion.c` code, which analyzes both the MSD (`-m` option) or the VACF (`-v` option) given a series of frames. The programs reads its data as from `stdin` and expects either only coordinates or velocities. Supposing you saved a set of configurations (`0.xyz`,...,`1000.xyz`,...), starting from an equilibrated one, you can provide the proper input to the analysis code with such a command line:

```
cat ?.xyz ??.xyz ???.xyz |  grep "^16" |  awk '{print $2,$3,$4}' |
          ./diffusion  -m -b 100 -n 216 -d 3  > msd.dat
```

the `xyz` files are first concatenated in temporal order, then `grep` is used to select the lines with information about position and velocities. Last, `awk` is used to print the 2nd, 3rd and 4th column, where the coordinates are stored. The output is then passed to the analysis code, specifying a window of 100 frames, 216 particles per frame, and 3 dimensions, and saved in the `msd.dat` file.

### 2.4 Homework (2 points)

1. Plot the MSDs and VACFs at two different densities (e.g., 0.5 and 0.8) As an example see figures 1 and 2. *Note that MSD should be plotted on a log-log scale.*

2. Explain the different behavior of the VACFs at the two densities.

3. Explain the different regimes in the MSD.

## References

[1] Molecular Simulation of Fluids, R. J. Sadus, Elsevier Science Publishers pp. 304 - 309

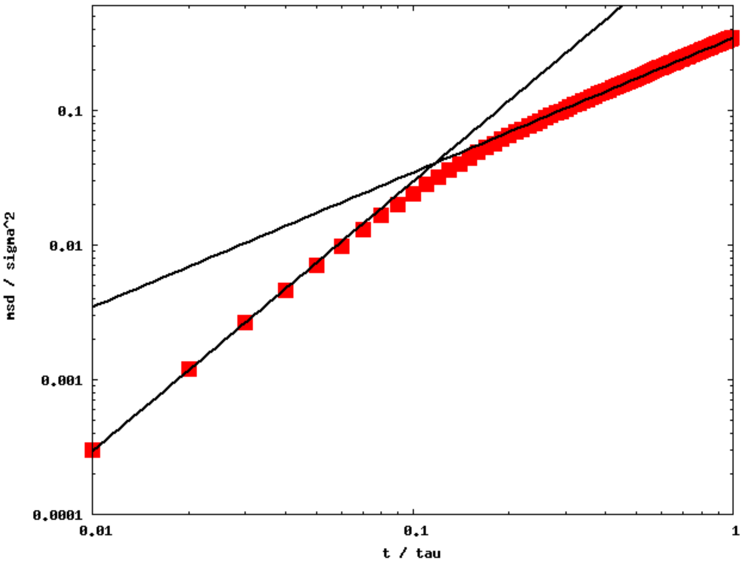

Figure 1: Typical mean square displacement of a Lennard-Jones fluid. The solid lines represent the balistic and diffusive behaviors.

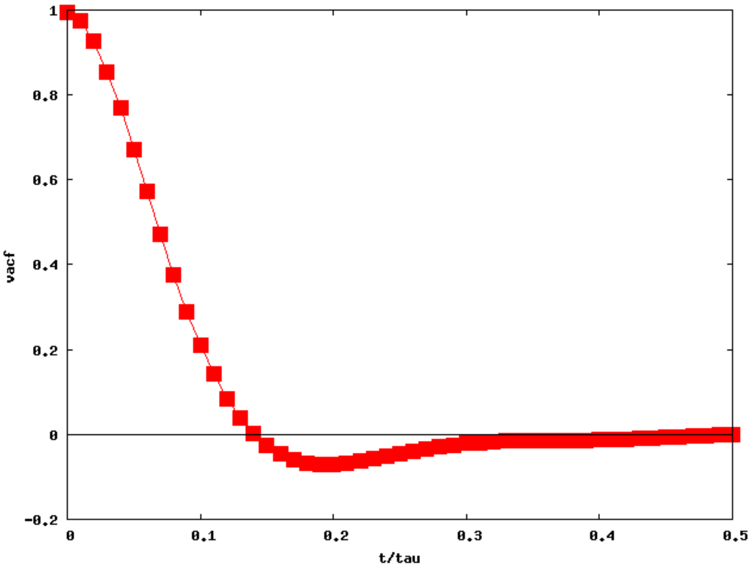

Figure 2: Typical velocity autocorrelation function of a Lennard-Jones fluid.