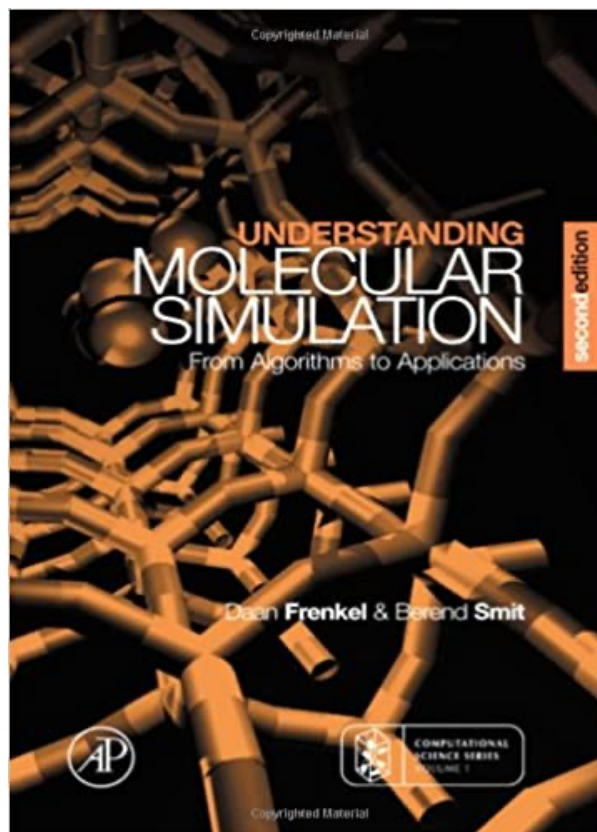# EPFL

CH-420 EXERCISE NOTES

Understanding Advanced Molecular Simulation

Berend Smit, Miriam Pougin, Sauradeep Majumdaar,
Elias Moubarak, Xiaoqi Zhang and Kevin Jablonka

Block 2

Basic Monte Carlo and Molecular Dynamics

2.1 Calculation of π

When we use Monte Carlo to perform a molecular simulation, we are actually evaluating a configurational integral or partition function. In mathematics, Monte Carlo is a common numerical algorithm to compute integrals (or the volume under a function), using random numbers instead of a regular grid. As a first example, we will use Monte Carlo to estimate π.

Consider a circle of diameter d surrounded by a square of length l ($l \geq d$). Random coordinates within the square are generated. The value of π can be calculated from the fraction of points that fall within the circle.
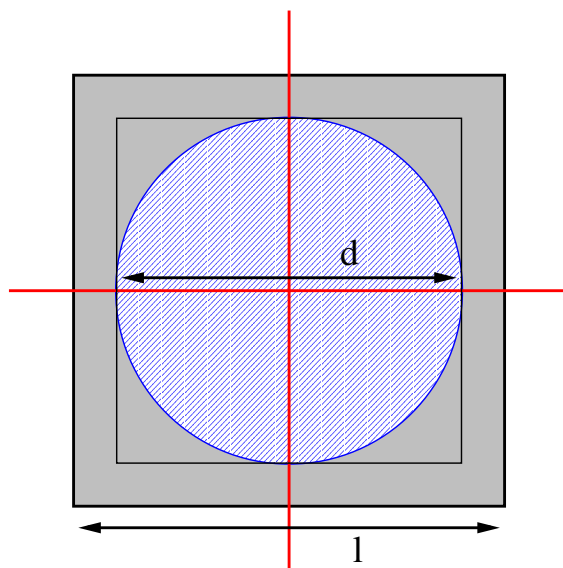

Figure 2.1: A circle of diameter d surrounded by a square of length l ($l \geq d$).

Questions:

1. How can π be calculated from the fraction of points that fall in the circle? Complete the small Monte Carlo program to calculate π using this method. Remark: the "exact" value

of π can be computed numerically using $\pi = 4 \times \arctan(1)$.

2. How would you expect that the accuracy of the result depends on the ratio l/d and the number of generated coordinates?

3. Derive a formula to calculate the relative standard deviation of the estimate of π. Hint: the distribution of successful hits is binomial. Is there an optimal ratio?

4. Is it a good idea to calculate many decimals of π using this method?

2.2 The photon gas

The average occupancy number of state j of the photon gas ($\langle n_j \rangle$) can be calculated analytically. From quantum mechanics we know that the total energy of the system (U) can be written as the sum of energies of harmonic oscillators:

$$U = \sum_{j=1}^{N} n_j \omega_j \hbar = \sum_{j=1}^{N} n_j \epsilon_j \tag{2.1}$$

in which ϵ_j is the characteristic energy of oscillator j, $n_j = 0, 1, 2, \cdots, \infty$ is the so called occupancy number of oscillator j, N is the number of oscillators, ω the frequency, and $2\pi\hbar$ Planck's constant. In this exercise we are going to compute this quantity, for one oscillator, using a Monte Carlo scheme. We will use the following procedure to calculate $\langle n_j \rangle$:

1. Start with an arbitrary n_j

2. Decide to perform a trial move to randomly increase or decrease n_j by 1.

3. Accept the trial move with probability

$$\text{acc}(o \rightarrow n) = \min\left(1, \exp\left[-\beta\left(U(n) - U(o)\right)\right]\right) \tag{2.2}$$

Of course n_j cannot become negative!

Questions:

1. How can this scheme obey detailed balance when $n_j = 0$? After all, n_j cannot become negative!

2. Is the algorithm still correct when trial moves are performed that change n_j with a random integer from the interval $[-5, 5]$? What happens when only trial moves are performed that change n_j with either -3 or $+3$?

3. Assume that $N = 1$ and $\epsilon_j = \epsilon$. Use the small Monte Carlo program and the "run_auto" script to calculate $\langle n_j \rangle$ as a function of $\beta\epsilon$. Compare your result with the analytical solution:

$$\langle n \rangle = \frac{1}{e^{\beta\epsilon} - 1} \tag{2.3}$$

4. Modify the program in such a way that the averages are updated only after an accepted trial move. Why does this lead to erroneous results? At which values of β does this error become more pronounced?

5. Modify the program in such a way that the distribution of n_j is calculated as well. Compare this distribution with the analytical distribution.
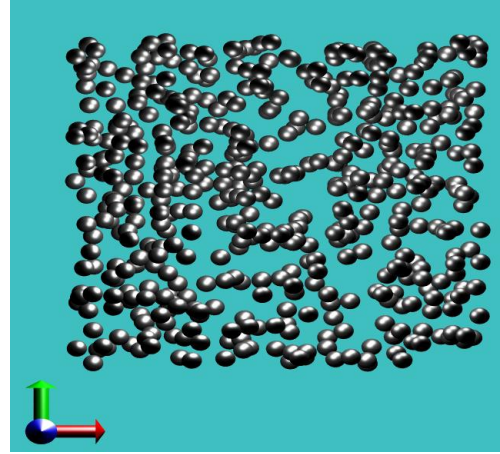
2.3 MC of a Lennard-Jones system

In this exercise, we will study a 3D Lennard-Jones NVT system. A cubic box of volume V encloses N particles at a given temperature T in any configuration allowed by the potential energy U. The potential is defined as

$$U(r) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] & r \leq r_c \\ 0 & r > r_c \end{cases} \tag{2.4}$$
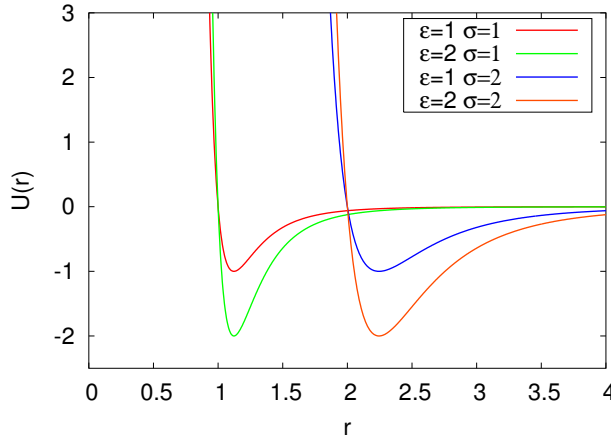
(a) The Lennard-Jones potential

(b) Snapshot from the Monte Carlo simulation.

Figure 2.2: MC of a Lennard-Jones system.

The contributions of the particles beyond this cutoff is estimated with the usual tail corrections:

$$\frac{u^{\text{tail}}}{N} = \frac{8}{3}\pi\rho\epsilon\sigma^3 \left[\frac{1}{3}\left(\frac{\sigma}{r_c}\right)^9 - \left(\frac{\sigma}{r_c}\right)^3 \right] \tag{2.5}$$

$$p^{\text{tail}} = \frac{16}{3}\pi\rho^2\epsilon\sigma^3 \left[\frac{2}{3}\left(\frac{\sigma}{r_c}\right)^9 - \left(\frac{\sigma}{r_c}\right)^3 \right] \tag{2.6}$$

$$\mu^{\text{tail}} = \frac{1}{\beta}\frac{16}{3}\pi\rho\epsilon\sigma^3 \left[\frac{1}{3}\left(\frac{\sigma}{r_c}\right)^9 - \left(\frac{\sigma}{r_c}\right)^3 \right] \tag{2.7}$$

A randomly selected particle is randomly translated to generate a new particle configuration. If this move takes the particle outside the cube into a surrounding or *image* cube, application of periodic boundary conditions means that a new particle is inserted at the corresponding position in the original cube. Whether the new configuration is accepted depends on the Boltzmann factor $e^{-\beta\Delta U}$, with $\beta = 1/(k_B T)$ and ΔU the energy difference between after and before the move. The procedure to ensure that configurations that are accepted occur with a frequency according to their Boltzmann weight involves generating a random number and a comparison with the Boltzmann weight. If the random number is smaller than the Boltzmann weight

$$\text{RandomNumber} < e^{-\beta\left(U^{\text{new}} - U^{\text{old}}\right)} \tag{2.8}$$

the move is accepted, otherwise rejected, but still counted again. Then another move is randomly chosen. Classical phase space is directly sampled and averages of physical properties become arithmetic averages over their sampled values. The moves is in the Monte Carlo method are artificial rather than dynamical (in time).

Questions:

1. In the present code, the pressure of the system is not calculated. Modify the code in such a way that the average pressure can be calculated. You will only have to make some modifications in the subroutine *energyparticle.f* or *energy.c* to calculate the virial. Using this quantity, the pressure will then be calculated automatically.

2. Perform a simulation at $T = 2.0$ and various densities. Up to which density does the ideal gas law

$$\beta p = \rho \tag{2.9}$$

   hold?

3. The program produces a sequence of snapshots of the state of the system. Try to visualise these snapshots using the program *vmd*.

4. It takes twice as much heat to heat a kilogram of a material one degree Kelvin as it does to heat half a kilogram. That's fairly intuitive. So when we talk about heat capacities, how much material are we talking about? The simplest way is to talk about the heat capacity per gram. This is known as the specific heat and is usually denoted by a small c. Done at constant volume it is c_V. Specific heats are very practical since it is usually easy to weigh things. But the resulting values don't tell us much about what is really going on. For instance, a gram of aluminium contains almost exactly 7.3 times as many atoms as a gram of gold. What would be interesting is the per atom (or more practically, the per mole) value of the heat capacity. And this molar heat capacity is usually what is given in tables. To avoid confusion the symbol for it is a capital C, C_V. For the heat capacity at constant volume one can derive

$$C_V = \frac{1}{N}\left(\frac{\partial U}{\partial T}\right)_V = \frac{\langle U^2 \rangle - \langle U \rangle^2}{Nk_BT^2} \tag{2.10}$$

   in which U is the total energy of the system. This is the correction to the ideal-gas heat capacity of $3/2k_B$, resulting from particle-particle interactions. Derive a formula for the dimensionless heat capacity. Modify the program (only in *mc_nvt.f* or *mc_nvt.c*) in such a way that C_V is calculated.

5. Instead of performing a trial move in which only one particle is displaced, one can do a trial move in which all particles are displaced. Compare the maximum displacements of these moves when 50% of all displacements are accepted.

6. Instead of using a uniformly distributed displacement, one can also use a Gaussian displacement. Does this increase the efficiency of the simulation?

2.4 MD of a Lennard-Jones system

Enclosed is a Molecular Dynamics (MD) program for a Lennard-Jones fluid in the NVE ensemble. Unfortunately, the program does not conserve the total energy because it contains three errors.

Questions:

1. Find the three errors in the code. The person that spots the errors first will receive a small prize Hint: there are two errors in *integrate.f* (*integrate.c*) and one in *force.f* (*force.c*). See the file *system.inc* (*system.h*) for documentation about some of the variables used in this code.

2. How is one able to control the temperature in this program? After all, the total energy of the system should be constant (not the temperature).

3. To test the energy drift ΔE of the numerical integration algorithm for a given time step Δt after N integration steps, one usually computes

$$\Delta E \left(\Delta t \right) = \frac{1}{N} \sum_{i=1}^{i=N} \left| \frac{E\left(0\right) - E\left(i\Delta t\right)}{E\left(0\right)} \right| \tag{2.11}$$

In this equation, $E\left(t\right)$ is the total energy (kinetic+potential) of the system at time t. Change the program, only in *mdloop.f* (*mdloop.c*), in such a way that ΔE is computed and make a plot of ΔE as a function of the time step. To maintain a certain energy drift, how does the time step have to change as a function of the temperature and density?

4. As you might have noticed in the code, the implementation of the periodic boundary conditions is often done as follows:

$$x = x - \text{box} * \text{nint}\left(x * \text{ibox}\right) \tag{2.12}$$

where ibox is used instead of $1/\text{box}$. Why would one do this?

5. An important quantity of a liquid or gas is the so-called self diffusivity D. There are two methods to calculate D:

    (a) by integrating the velocity autocorrelation function:

    $$\begin{aligned} D &= \frac{1}{3} \int_0^\infty \left\langle \mathbf{v}\left(t\right) \cdot \mathbf{v}\left(t + t'\right) \right\rangle dt' \\ &= \frac{\int_0^\infty \sum_{i=1}^{i=N} \left\langle \mathbf{v}\left(i,t\right) \cdot \mathbf{v}\left(i,t+t'\right) \right\rangle dt'}{3N} \end{aligned} \tag{2.13}$$

    in which N is the number of particles and $\mathbf{v}\left(i,t\right)$ is the velocity of particle i at time t. One should choose t in such a way that independent time origins are taken, *i.e.* $t = ia\Delta t$, $i = 1, 2, \cdots, \infty$ and $\left\langle \mathbf{v}\left(t\right) \cdot \mathbf{v}\left(t + a\Delta t\right) \right\rangle \approx \mathbf{0}$ (why?).

    (b) by calculating the mean square displacement:

    $$D = \lim_{t' \to \infty} \frac{\left\langle \left| \mathbf{x}\left(t + t'\right) - \mathbf{x}\left(t\right) \right|^2 \right\rangle}{6t'} \tag{2.14}$$

    One should be very careful with the mean square displacement calculation when particles are always translated back to the central box – why?

Modify the program in such a way that the self diffusivity can be calculated using both methods. Only modifications in subroutine *sample_diff.f* (*sample_diff.c*) are needed. Why is it important to use only independent time origins for the calculation of the mean square displacement and the velocity autocorrelation function? What is the unit of D in SI units? How can one transform D into dimensionless units?

6. For Lennard-Jones liquids, Naghizadeh and Rice report the following equation for the self diffusivity (dimensionless units, $T^* < 1.0$ and $p^* < 3.0$)

$$^{10}\log\left(D^*\right) = 0.05 + 0.07p^* - \frac{1.04 + 0.1p^*}{T^*} \qquad (2.15)$$

Try to confirm this equation with simulations.

7. Instead of calculating the average potential $\langle U \rangle$ directly, one can use the radial distribution function $g(r)$. Derive an expression for $\langle U \rangle$ using $g(r)$. Compare this calculation with a direct calculation of the average energy. A similar method can be used to compute the average pressure.

8. In the current version of the code, the equation of motion are integrated by the Verlet algorithm. Make a plot of the energy drift ΔU for the following integration algorithms

   - Euler (never use this one except here !!!)
   - Verlet
   - Velocity Verlet