Computer simulation of physical systems I

Task VI: Monte-Carlo simulation of Lennard-Jones liquid

In this task, we will study the Lennard-Jones liquid, as in Task II, but this time using the Monte-Carlo (Metropolis) method. The code uses again LJ units; for reminder, see the additional notes at the end of Task II.

The parameters for the programs and other detailed instructions

Please go through the provided python scripts trying to understand what are the meaning of each parameter in the scripts. The parameters are specified in the begining of the python scripts.

Usages of $\boldsymbol{mc\_sc.py}$ and $\boldsymbol{mc\_argon.py}$
$-----------$
$parameters$:
eps, sig - LJ potential parameters
mass - Mass of particles in LJ units
nequil - Number of equilibration cycles
lmax - Number of production cycle
nsamp - Steps between property sampling
dmax - maximal displacement
ndispl - Number of displacement attempts per cycle
npart - Number of particles
temp - Temperature
rho - Particle density (not mass density)
nbins - number of points to sample g(r)

In addition to the standard output, the program $mc\_sc.py$ also creates two other output files:
$energy$ - The energy property sampled during the run
$pressure$ - The pressure property sampled during the run

Usage of $blocking.py$
$-----------$
$parameters$:
fn - file name of data file to be read
ntrans - number of steps of block averaging

The program is run by giving the name of the data-file in the script. In our case, one needs to run blocking averaging for two properties: the energy and the pressure. The output then includes the estimates for the standard deviation and an estimate for its error, first for energy and then for pressure, when the size of the block is varied (as 2^n) (the blocking transformation of Flyvbjerg and Petersen) The first line gives the standard deviation without block averaging.

1. We will simulate a system of 200 particles. Let us choose temperature $T = 2.0$ and density $\rho = 0.5$. You can run a simulation by executing the script as $./mc\_sc.py$ >out. The output is written to the file "out". Examine the output of the Monte-Carlo simulation program. Find out how many cycles (also how many displacement attempts and how many displacements) are needed to obtain convergence in both the energy and the pressure; and plot them. Observe how the step length is adjusted during the run in order to attain the (most efficient) acceptance rate of about 50%.

Calculate the $T = 2$ isotherm (pressure vs. density) of the LJ liquid for particle densities ranging from 0.1 to 0.9. Set a sufficient number of equilibration steps to the input file. Run the code for different values of density ρ and plot the results. Compare to Figure 3.5a in Frenkel-Smit.

You can also try to reproduce the $T = 0.9$ isotherm of Fig 3.5b. Is there something wrong with our simulated results?

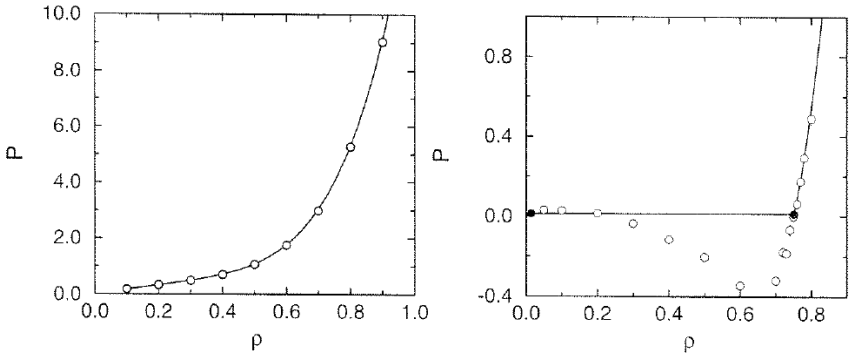

Figure 3.5: Equation of state of the Lennard-Jones fluid. (left) Isotherm at T = 2.0. (right) Isotherm below the critical temperature (T = 0.9); the horizontal line is the saturated vapor pressure and the filled circles indicate the densities of the coexisting vapor and liquid phases. The solid curve represents the equation of state of Johnson *et al.* [62] and the circles are the results of the simulations (N = 500). The errors are smaller than the symbol size.

2. Study of correlation. Block averaging can be used to obtain a more reliable error estimate when the sampling is correlated, like in the case of Metropolis sampling.

Continuing with the system of 200 particles, $T = 2.0$ and $\rho = 0.5$ (Remember to do the equilibration), plot the evolution of the block-averaging errors of energy and pressure as the block size increases. Execute the script as $./blocking.py$. Increase the number of cycles until a formation of a plateau can be observed. In this case the blocks becomes larger than the correlation length. Compare the magnitude of errors when calculated with and without block averaging. Naturally, if you increase the number of steps between samplings, you should find out that the samples are less correlated. Compare the correlation length to the number of equilibration steps.

3. Calculation of $g(r)$ and $S(k)$ for argon.

That is, we have a system of 864 argon atoms, for which we fix $\sigma = 3.4$ Å and $\varepsilon/k_B = 120$ K. Perform the simulation temperature in $T = 94$ K (0.7833 in LJ units) and density of $\rho = 1.374$ g·cm^{-3} (0.8072 particle density in LJ units). Mass of argon atom is about

$6.69 \cdot 10^{-26}$ kg. Find out how to obtain the temperature and particle density in the used LJ units by yourself. Execute the script as $./mc_argon.py >$out2 and two data files will be generated: $gr.dat$ and $sk.dat$.

You should set the number of steps between the sampling of $g(r)$ to something similar to the correlation length as obtained from the block averaging. You should also set the sampling frequency nstep to something bigger than 1, because calculating these functions can take quite a lot of time. A good choice could be a value close to the "correlation length" as obtained from the block size in Step2. Actually, after the equilibration, $g(r)$ does not change much i.e., you don't need many time-steps. Remember, that you could start from a configuration that you know is already equilibrated. The plots of the average of the sampled $g(r)$ and $S(k)$ are presented in $gofr.pdf$ and $sofk.pdf$. You can generate your own plots. Compare them to that obtained in Task II and in Rahman's paper.