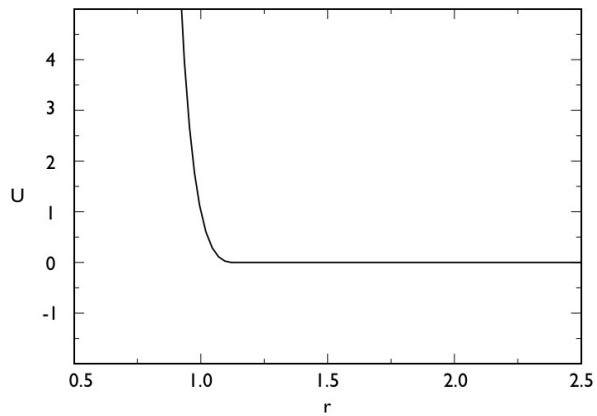
Structure and Simulation: Monte Carlo exercise Answers

31.5.2011

General notes on the program: force field

- we are simulating a Lennard-Jones system, e.g. liquid Argon
- periodic boundary conditions
- nearest image convention: $r_{\text{cut}} < L/2$
- Weeks-Chandler-Andersen (WCA) potential
  - special case of the Lennard-Jones (LJ) potential
  - truncated at minimum of LJ potential
  - minimum shifted to zero
  - so both energy and force are zero at the cutoff
  - purely repulsive



picture from `http://matdl.org/matdlwiki/index.php/Image:WCA_potential.jpg`

General notes on the program: simulation

- in the NVT ensemble, the following are constant throughout the simulation
  - number of particles (N)
  - volume (V)
  - temperature (T)
- we estimate ensemble averages via Metropolis Monte Carlo
- this program calculates:
  - potential energy per particle
  - pressure (by calculating the virial)
- there are no dynamics, therefore no explicit kinetic energy
- once again: ensemble averages are all that matter here

Short reminder: Ensemble averages

- NVT ensemble: we have many snapshots (microstates) of our system, each microstate has a probability
  - ensemble: Ω
  - snapshot / microstate: ω
  - probability of a microstate in the ensemble: $p(\omega)$
- given some physical property of the system, e.g. potential energy, pressure, etc.
  - physical property of a snapshot: $A(\omega)$
- then the expected value of A averaged over all of Ω is

$$\langle A \rangle = \sum_{\omega \in \Omega} p(\omega) A(\omega)$$

or

$$\langle A \rangle = \int_{\Omega} p(\omega) A(\omega) d\omega$$

Short reminder: Monte Carlo integration

Problems of naive summation or integration

- usually can't exhaustively sum a huge number of states
- direct numerical integration often impossible for high-dimensional systems (such as ours)

Monte Carlo integration

- take random snapshots, calculate physical property, weigh with snapshot probability, and average this
- Problem: we need to know the partition function (Z) to know the probability of a state in the NVT ensemble

$$p(\omega) = \frac{1}{Z} e^{-E(\omega)/kT}$$

$$Z = \sum_{\omega \in \Omega} e^{-E(\omega)/kT}$$

or

$$Z = \int_{\Omega} e^{-E(\omega)/kT} d\omega$$

Short reminder: Monte Carlo integration (contd.)

More problems of Monte Carlo integration (for our system)

- at liquid densities there will be
  - many many high-energy conformations with a tiny tiny probability
  - few low-energy conformations with a comparably high probability
  - ... but these low-energy conformations will dominate the ensemble averages
- Monte Carlo integration samples the microstates uniformly, then weighs them with their probability
- it would be good if we didn't waste so much time with low-probability states that don't contribute to the average

Short reminder: Metropolis Monte Carlo

Metropolis Monte Carlo to the rescue

- Metropolis Monte Carlo constructs a Markov chain of states
- the Markov chain visits states according to their probabilities
  - without having to know the partition function beforehand !
- a uniform (unweighted) average gives us the ensemble average
- same result as Monte Carlo integration, but much faster

Short reminder: Metropolis Monte Carlo (contd.)

The Markov chain in Metropolis Monte Carlo

- we generate a new state by making a move from an old state
- move probability consists of trial probability and acceptance probability
- acceptance probability depends only on energy difference

Careful:

- detailed balance for trial probability guarantees desired equilibrium probability distribution of generated states
- without detailed balance it most probably won't work

Back to our program...

Metropolis Monte Carlo quite simple to use in practise

- from old state make a trial move
  - our program tries to move each particle in turn
  - all directions are equally likely, so detailed balance holds
- accept move depending on energy difference between new and old state
- after each trial move, we need to only recalculate the interactions of the particle we attempted to move
  - this is done by the `calc_i()` function in the program

General comments on the exercise

- graphs must have
  - labels on axes
  - units on axes (here: “reduced units”)
  - a title or caption to state what is shown
- look at both pressure and potential energy
- look at averages
  - some things are hard to see from instantaneous values alone
  - it is the averages we are interested in

How to make proper graphs with gnuplot

... maybe this should have been in the exercise notes ...

```
set title 'Average pressure comparison'
set xlabel 'step'
set ylabel 'pressure (reduced units)'
plot "mc.res" us 1:2 t 'monte',
    "mc_2.res" us 1:2 t 'monte_2'
```

Part I: Convergence of averages

- the simulation starts in a cubic lattice
  - often partially filled (depends on $N_{\text{particles}}$)
  - that state is probably very un-typical
  - we generate new states from old ones, so the first states will all be unrealistic
- ignore the beginning of the simulation
  - it would be better if the computed averages did this too
- wait until averages have converged
- how long in general?
  - depends on force field, density, temperature
- could be treacherous
  - metastable states give a false sense of convergence
  - our system should be safe from that if we keep density and temperature in a friendly range (density not too high, temperature not too low)

Part II: Matching averages

- number of particles should have little impact on the averages
  - remember that potential energy is calculated per particle
  - pressure is an intensive property
- qualitatively, higher temperature leads to
  - higher average potential energy per particle
  - higher average pressure
- qualitatively, higher density leads to
  - lower average potential energy per particle, then higher again at high densities
  - higher average pressure
- can search by trial and error
- or write a script to search the parameter space

Part III: Miscalculated averages

- particles only moving in one direction
- detailed balance is violated
- instantaneous values look similar
- need to look at averages to see the difference