Simple Discrete Simulations

Software project May 2013



# 1 State of the programs

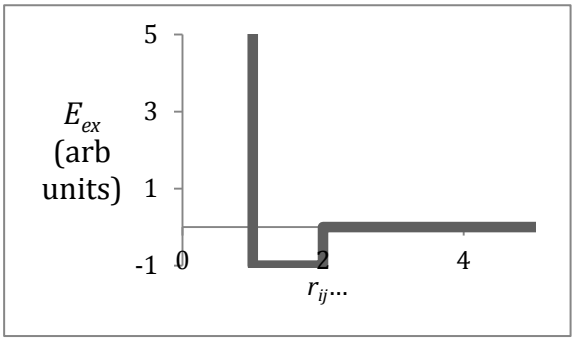
Do not go forward unless you are happy with your programs. You must be able to

- put down particles randomly
- put down particles in the first 10% of the boxin

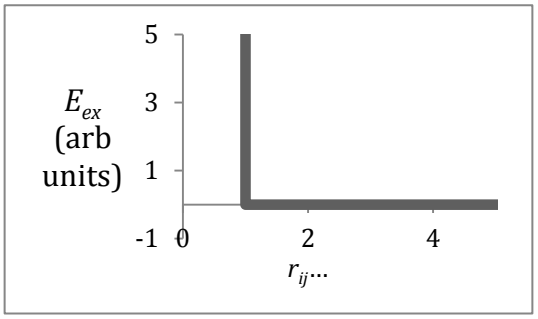
# 2 Discuss at start

In the original handout, there were several issues which left slightly open.

- How did you implement the options (number of particles, size, how often to write out coordinates) ?
- Simulating: What is the most elegant way to implement




and




within one program ?

- How did you implement running with an electrostatic term ($^{q_1 q_2}/_{cr_{12}}$) and switching it on and off ?
- The original notes said you should be able to write out coordinates every *n* steps for analysis. What format have you decided on ? There are many sensible options. It would be useful if we could agree on a format, so your program can read coordinates written by someone else's program.
- Do periodic boundary conditions work or have you had problems ?

## 3 Equilibration and walls

The initial coordinates that you generate are not typical of an equilibrium state. They would be representative of a high energy state. In most molecular simulation programs, you

- run for $n_{start}$ steps
- throw these away
- run for $n_{sim}$ step and keep the results for analysis.

This will be the plan here. There is a complication. We would like to be able to let the system equilibrate with the restriction that the particles just distribute themselves over the first 10% of the box. You might be tempted to add another option to the program, but it is probably already becoming overloaded. Let us say, we have

- set $x_{max}$ to 10% of the value you want
- simulate for $n_{start}$ steps
- write the coordinates to a file
- use these as starting coordinates for a real simulation

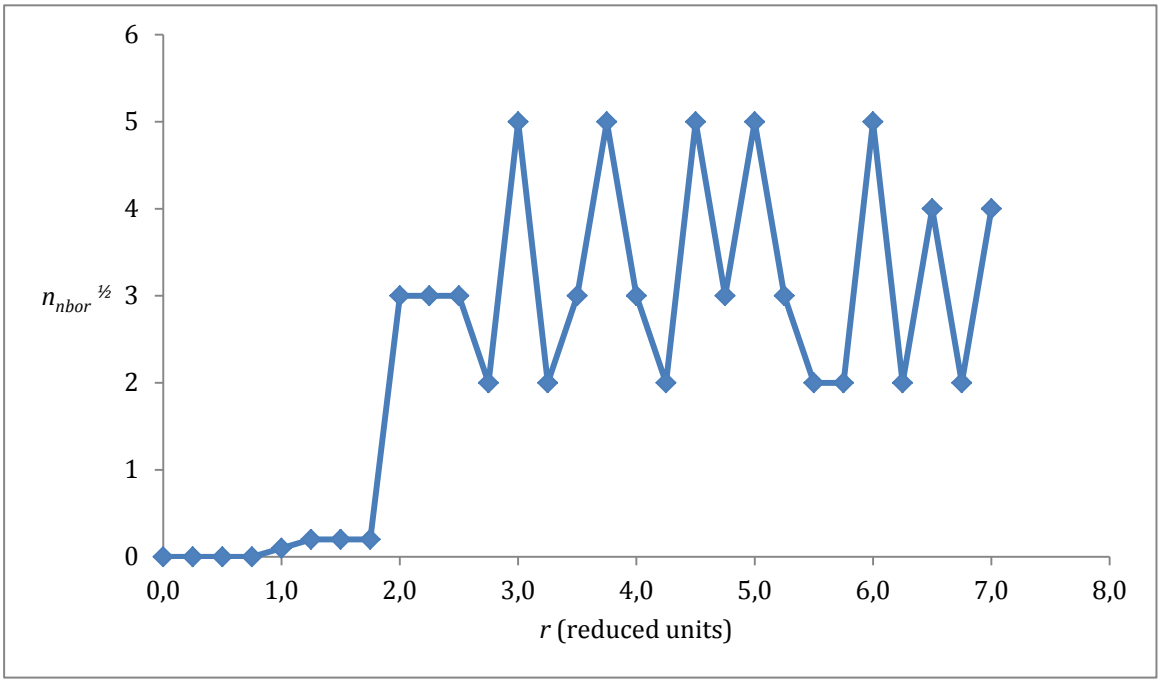
## 4 Analysis

You might now want to do some simulations and establish values for parameters. It is not that easy. If you want to look at parameters, you have to be able to see their effect. There are two kinds of properties

1. Kinetic/dynamic – all properties with time dependence
2. Structural – at equilibrium, structural properties are time independent

To look at properties of energy functions, you normally look at structural properties. We are interested in the size of energy wells/valleys and these are reflected in distances between particles. The most important structural property is the number of neighbours a particle has, as a function of distance.

Consider an infinite system. The number of neighbours will grow with $r^2$ where $r$ is the distance from the particle. Remember the particles have a radius of 1.0 length units. What you want is a plot of the form.




Your plot will not look like this. It should be smoother and make sense in terms of the physics. Note that one plots the square root of the number of neighbours.

### 4.1 Implementation

This is for you to work out.

You must be able to read input coordinates. You must be able to write *x*, *y* data, suitable for a plotting program. Try to plan so you can expand your program later, to read several sets of coordinates. You do not want the statistics for one particle. You want the averages over all particles. This means, for each particle *i*, you have to visit all of the neighbours *j*

```
set up an array for the counts of neighbours
for (i = 0; i < n−1; i++)
   for (j = i + 1; j < n; j++)
      r = distance (i,j)
      increment the counter for the  bin for r
```

But when you print out the data, print out the $\left(n_{neighbour}\right)^{\frac{1}{2}}$.

# 5 Parameters

Note that in physics, it usually only makes sense to talk about energies as big or small when you consider temperature. Think about the Boltzmann weight, $e^{-E/kT}$. If you scale both energy and temperature, the system looks the same. In the text below, we scale the energy.

## 5.1 First version

You want to be able to find reasonable values for temperature, *T* and the well depth. As a start, use the option which randomly places particles in the full box. Start with 100 or 200 particles. You could use a bigger number, but then it is hard to check what is happening by hand. Estimate the box size you would need if you have the particles on a grid or lattice. You know they have a radius of 1 unit and will pack with a spacing of two length units. Make your initial box size three times this size in both *x* and *y* directions. You now have a system of medium density. If the depth of the energy well (*c* in the previous handout) is big, the system should freeze rather easily. If *c* small, the system will never freeze.

Your mission sounds easy. Try to find values for *c* where the system seems structured. There may not be so easy.

Use simulations of $10^5$ or $10^6$ steps. This should be quite fast. When you are happy, try scaling your system. Increase the *x* and *y* lengths and the number of particles. This should not change and results.

## 5.2 Second version

Keep the number of particles fixed, but make the box five times larger on each side. The value for *c* which leads to a phase transition should change. Why ?