universität innsbruck
Institut für Theoretische Physik

Prof. Dr. T. Franosch

Dr. Michele Caraglio

Institut für Theoretische Physik

WiSe 2021 19.11.2021

Computer Simulations in Statistical Physics

Prof. Dr. T. Franosch

Problem set 5

Proseminar

Problem 5.1 *Characterizing Structural Properties: The Radial Distribution Function*

A very important observable is the radial distribution function (RDF) $g(r)$ which describes the structure around the particles [1]. For a mono-atomic liquid it is defined as,

$$g(r) = \frac{1}{N\varrho V(r)} \sum_i \sum_j \delta(r_{ij} - r), \tag{1}$$

with $V(r) = \frac{4}{3}\pi((r+\Delta r)^3 - r^3)$ (the size of the spherical shell in which we count). Here, Δr is the size of the bin in the histogram with which you discretize the delta function (choose $\Delta r = L/(2N_{\text{hist}})$ with $N_{\text{hist}} = 200$). The density is $\varrho = 6\varphi/\pi$. For $r \to \infty$ the radial distribution function is supposed to converge, $\lim_{r\to\infty} g(r) = 1.0$.

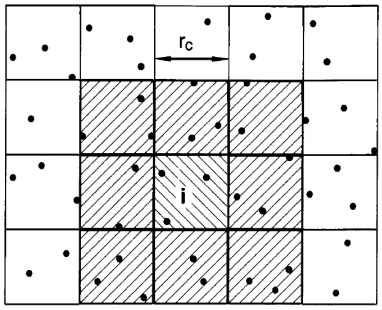
a) Write a function "calc_print_gr" that determines and print on a file the RDF. Loop over all pairs of particles and create a histogram with N_{hist} bins. In bin i you store how many particles are within a range $r \in [r_i, r_i + \Delta r]$ of each other, with $r_i = i\Delta r$.

b) Determine the RDF for $\varphi = 0.4$ and $N = 10^3$ directly after initialization and after equilibration. What do you observe?

[1] The RDF will therefore obtain $g(r) = 1\,\forall r$ in an ideal gas.

Problem 5.2 *Saving CPU Time: Cell Lists*

The most significant problem of the algorithm you have used so far is that it scales with $\mathcal{O}(N^2)$, where N is the number of particles. Since we aim at simulating $10^5 - 10^6$ particles, the simulation time will explode if the number particles becomes too large. Physically, however, every particle will only interact with its nearest neighbors, which should allow to construct much more efficient algorithm. The easiest - and to the same time already very effective - solution is including cell lists. To create these lists, the whole simulation box is divided into domains (domain decomposition). Each domain has a size that is larger than the range of the longest interaction. This automatically means, that a particle i in one domain/cell can only interact with particles in neighboring cells (see sketch underneath this paragraph). Since the cell size is independent of the system size, the algorithm scales like $\mathcal{O}(N)$, which is a huge improvement.



a) Write a function "create_cell_lists()" which loops over all particles and allocates them to cell lists. These cell lists are two dimensional arrays. In the first dimension the index of the lists is stored, in the second dimension the indices of the particles that will be included into this cell. The second dimension thus needs $N_{\max}$ storage, where $N_{\max}$ denotes the maximum number of particles that can be compressed into one cell (in our case with $r_c = 1.5\sigma$, $N_{\max} \approx 10$). Do not forget to store for each particle which cell list it was assigned to.

b) Write a function "determine_overlap(...)" which uses the cell lists to determine, for a particle i, whether it overlaps with another particle. For this operation the index of its cell lists has to be determined. Afterwards, you just have to loop over all particles within the neighbouring cell lists.

Tipp: Cell lists do not affect which particles interact with each other. Use the same seed for the random number generator and compare your simulation results with and without cell lists. The numbers have to be *exactly* the same. Otherwise, continue debugging.

c) Use the two above functions to perform simulations for $\varphi = 0.4$ with and without cell lists for $N = 10^3$. Compare the simulation time. Ideally, vary the system size between $N = 5^3$ and $N = 50^3$ (you just have to do very short simulations) and analyse the scaling with the particle number N.

d) Initialize systems with $\varphi = 0.48$, $\varphi = 0.52$ and $\varphi = 0.55$ and $N > 15^3$. Simulate the system for at least 5000 MC steps and determine the radial distribution function. Discuss your observations using also the visualization of snapshots. Answer the question: Can hard spheres crystallize and if "yes" why?