

Molecular Modeling: Project 4

Markus Rauhalahti, Alex Odiyo, Timo Pekkanen, Gaoyue Bi

May 8, 2017

1 Introduction

In this project, routines for the calculation of the diffusion coefficient and the pair correlation function were implemented to a molecular dynamics program. Pair correlation function was analyzed for a simulation of noble gases to examine the effect of density and temperature on the phase behavior. The actual selected parameters for simulation were Argon gas parameters.

Monte Carlo algorithm was implemented and used to compute the pair correlation function.

2 Theory and methods

2.1 Diffusion coefficient

The diffusion coefficient describes the propensity of particles to move from higher to lower chemical potential. The Einstein relation relates the diffusion coefficient to the mean-square displacement (MSD) of the particle [WZCJ08]

$$D = \lim_{t \rightarrow \infty} \frac{1}{6Nt} < \sum_{j=1}^N [r_j(t) - r_j(t_0)]^2 > \quad (1)$$

In practice this is done by calculating the MSD at specified intervals and by performing a linear regression [Kef01]. The system needs to be equilibrated in advance to the computation of the diffusion coefficient. The diffusion coefficient is then obtained as the slope

$$< \sum_{j=1}^N [r_j(t_0 + t) - r_j(t_0)]^2 > = b_0 + 2dDt, \quad (2)$$

where d is the dimensionality of the system, b_0 is the intercept, and t is the simulation time.

For ideal gases, the diffusion coefficient can be calculated from

$$D = \frac{1}{3} \lambda v_{\text{mean}} = \frac{2\mu RT}{3pM}, \quad (3)$$

where λ is the mean free path and v_{mean} is the mean speed, which have the definitions

$$\lambda = \frac{\mu}{p} \sqrt{\frac{\pi RT}{2M}} \quad \text{and} \quad (4)$$

$$v_{\text{mean}} = \sqrt{\frac{8RT}{\pi M}}. \quad (5)$$

It will be interesting to see how value of the diffusion coefficient we obtain from the computations compares to the analytical ideal gas diffusion coefficient. The diffusion coefficient can also be expressed in terms of density,

$$D = \frac{2\mu}{3\rho}, \quad (6)$$

since for ideal gases $p = \frac{\rho RT}{M}$. In the equation above μ is viscosity. The viscosity of Argon can be approximated from the expression¹

$$\mu = 2.125 \cdot 10^{-5} \cdot \frac{417.51 \text{ K}}{T + 144.4 \text{ K}} \left(\frac{T}{273.11 \text{ K}} \right)^{\frac{3}{2}} \text{ Pa s}. \quad (7)$$

2.2 Pair correlation function

Fluids that possess to a great order an average molecular organization, have well defined structural correlations. To obtain such information on real space correlation function, the quantity of interest is first calculated from several different starting configurations, calculating the correlation for function for each configuration and finally taking the average.

The pair correlation function $g(r)$, or the density density correlation, describes the radial variation of particle density. In the language of statistical mechanics it may be defined as

$$g^{(n)}(\mathbf{r}_1, \dots, \mathbf{r}_n) = \frac{V^n N!}{N^n (N-n)!} \cdot \frac{1}{Z_N} \int \dots \int e^{-\beta U_N} d\mathbf{r}_{n+1} \dots d\mathbf{r}_N \quad (8)$$

For a truly isotropic system, the pair correlation function's angular dependence averages out. The resultant function is simply the radial distribution function. Hence the interchangeable names.

In practice the following algorithm² is used. For each particle a histogram is formed describing the number of particles between $[r', r' + \Delta r]$, where $r' \in [0, r_c]$ extends to a cutoff of distance r_c . The results for all particles are added to a one histogram which is then normalized by dividing with the particle number N , the particle density ρ , and by the volume of a spherical shell, $4\pi r^2 \Delta r$.

$$g(r) = \frac{1}{N\rho 4\pi r^2 \Delta r} \sum \text{histograms} \quad (9)$$

2.3 Monte Carlo

The Monte Carlo (MC) algorithm is an optimization method, which is useful for sampling complex phase spaces. In molecular simulations it is performed by doing random displacements to a particle configuration.

MC is fast method of checking the outcome of a system without necessarily considering all the events that necessarily happens in between the start and end of the simulation. The time for execution of the MC method is almost independent of system size but simulation time may vary slightly depending on the number of simulation repetitions. More repetitions require more but generates better averages of the system.

The algorithm is straight forward to implement. If we have an initial configuration we need to perturb the system by some random displacement. If this random displacement lowers the energy of the system, the new configuration is adapted. If not, a random number is picked and compared to the Boltzmann factor $e^{\Delta U/kT}$. If the random number is larger than the Boltzmann factor, the system is moved to a new configuration. Then the processes of random perturbation and configuration adjustment is done recursively many times and the averages computed.

In this work the MC algorithm is applied by itself for the computation of the pair correlation function, and in combination to MM to obtain the diffusion constant.

¹F. M. White. *Viscous Fluid Flow*. McGraw-Hill, 2. edition, 1991.

²<http://www.physics.emory.edu/faculty/weeks/idl/gofr2.html>

2.4 Details on the implementation

The program was implemented using the Fortran language. Visualization and linear regression were done with python using the numpy, scipy, and matplotlib libraries.

The gas and Lennard Jones potential parameters used in the program are $\epsilon = 1.65 \cdot 10^{-21} \text{J}$, $\sigma = 3.4 \text{ nm}$, $\rho = 1.633 \cdot 10^{-3} \text{ g cm}^{-3}$, mass = 39.948 g. Other simulation parameters such as the cube length is fixed at $40 \cdot \sigma$ and the time step for MD $\delta t = 1 \cdot 10^{-8} \text{ s}$. These properties are global to both the MD and MC subroutines.

The MD code utilizes periodic boundary condition and a linked list system. The MC section employs the Metropolis algorithm. Although a random particle is chosen, its maximum displacement is restricted not to over exceed more than half the cube length. A good statistics level has been ensured through the usage of a minimum of 100000 steps. There's an additional code to compute the diffusion coefficients and radial functions.

3 Results

For various temperature changes, assuming that no particles escape from the simulation environment, only MD shows phase change due to temperature. The MC plot does not show any phase change.

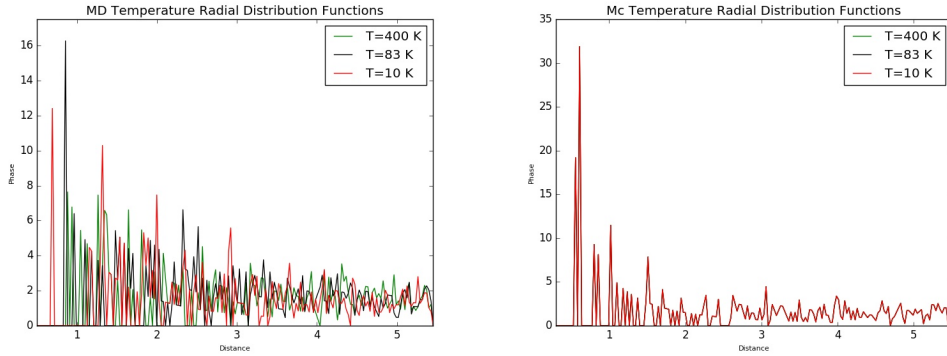


Figure 1: Radial distribution function vs temperature plot

To test the effect of density, Volume is fixed (cube dimensions being constant) and simulations are done for varying number of particles at constant temperature. The results shows that both MD and MC simulation are sensitive to sample density variation

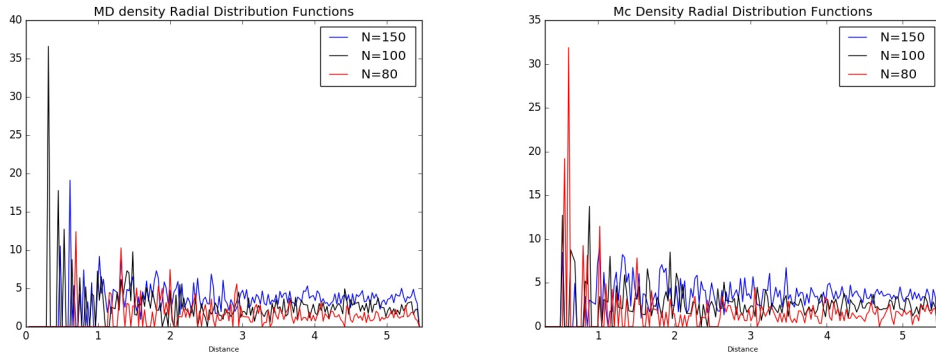


Figure 2: Radial distribution function vs number density plot.

The MC simulation procedure involving identical simulation steps takes a much shorter time to execute than an optimized MD code. The actual simulation times are tabulated in table 1 below.

N	Temp (K)	mdstime (s)	mcstime (s)
200	10.000	258.392000	1.360000
150	10.000	264.200000	1.036000
100	10.000	246.372000	0.748000
80	10.000	289.712000	0.608000
60	10.000	280.300000	0.492000
50	10.000	347.424000	0.380000

The MSD as a function of step is shown in figure 3.

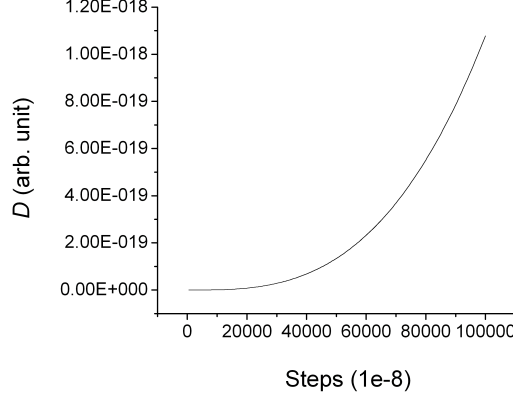


Figure 3: MSD as a function of steps for a MD simulation of 20 argon atoms for 100000 timesteps

As can be seen, the graph above is not reliably linear at any point, so a linear fit was not made to obtain the diffusion coefficient.

4 Conclusions

As expected, the Monte Carlo method was far more efficient in calculating the pair correlation function. The Monte Carlo simulations were several orders of magnitude faster than the corresponding molecular dynamics simulations. The determination of the pair correlation function was relatively successful in this project.

On the other hand, we had great difficulty determining the diffusion coefficient, As can be seen from the graph above, the MSD vs, step function is not reliably linear at any points. This suggests that the system is not properly equilibrated. If the system is not equilibrated, it is not possible to determine the diffusion coefficient from equation 2. The energy profiles of the system very strongly suggested that the system is not close to equilibrium.

Despite our efforts, we were not able to find the reason for this problem. We are therefore unable to say if the function that calculates the diffusion coefficient is working properly.

More time is needed to fix the bugs in the code.

References

- [Kef01] David Keffer. The working man's guide to obtaining self diffusion coefficients from molecular dynamics simulations. *Department of Chemical Engineering, University of Tennessee, Knoxville*, 2001.
- [WZCJ08] Li Wei-Zhong, Chen Cong, and Yang Jian. Molecular dynamics simulation of self-diffusion coefficient and its relation with temperature using simple lennard-jones potential. *Heat Transfer—Asian Research*, 37(2):86–93, 2008.