Molecular dynamics simulation for hard sphere

Alessandro Orsini

May 12, 2024

Abstract

The aim of this proget is to study the *antikinetics* behavior for a hard sphere system and in order to obtain this we work at it through molecular dynamics simulation and we analyze the H function.

1 Introduction

To study the *kinetic* and *antikinetics* behaviour of our system, we analize the trend of the $H(t)$ function. It can be defined as it follows

$$H(t) = \int d\vec{p} f(\vec{p}, t) \ln[f(\vec{p}, t)]$$

where $f(\vec{p}, t)$ is the distribution of momenta. It can be proof that if $f(\vec{p}, t)$ is the Maxwell-Boltzmann solution, in other words this means that we react equilibrium, $H(t)$ is a monotonic decresing function. This is also called as *H-Theorem* that shows that if we use the molecular chaos approximation, or *Stosszahlansatz*, we obtain $\frac{dH(t)}{dt} \leq 0$ and this represents if we start from a correlated state during the simulation the two particles will be uncorrelate and the system tends to equilibrium thanks to interaction.

Soon after Boltzmann published his results, some physics, like Zermelo and Loschmidt, object Boltzmann thesis . The Zermelo paradox is out of our study, because it refers to returning time and Poincare recurrence time. It's known the solution of this problem reguards the difference in time scale between the returning time, find out from the Pincare theorem and duration of any observable. The Loschmidt's one is much complicated, from each microstate that approaches equilibrium (kinetic behaviour), another state can be obtained reversing all the molecular velocities and the system goes away from equilibrium. Does it true for every time? In other words, if we invert the velocities of each particle in different time, also once we reach equilibrium, the H function tends to its initial value? If we put also a small error in the velocity inversion what will change ? In order to do that we performed molecular dynamics simulation for an hard sphere system starting from a crystal described as follows.

2 Molecular dynamics for hard sphere

The molecular dynamics helps us to study the dynamics for a given system, and it consists in the solution of the equation of motion for each particles using determined programs, but in our example the scheme is a little different. During our simulation, when the distance for two particles becomes equal there is a discontinuity in our potential. To overcome this problem the principle aim of the simulation is to locate the time, collision parameter and all impact factors for each collusion occurring in the system in chronological order. Instead of a step by step simulation like in the case of soft potential we analize the "collision dynamics" and then search for the next collision. This scheme can be thought as it follows:

- locate next collision;
- move all particles forward until collision occurs;
- implement collision dynamics for the collinding part;
- calcolate the collision properties, ready for averaging, before return to first point;

To locate the collision time for hard sphere, means to find the solution of a quadratic equation given by the definition. Considering two particles, i and j, of diameter σ and whose positions are $r_i(t)$ and $r_j(t)$ and respective velocities v_i and v_j . When a collison occurs then at $t = t + t_{ij}$

$$|r_{ij}(t + t_{ij})| = |r_{ij} + v_{ij}t_{ij}| = \sigma$$

where $r_{ij} = r_i - r_j$ and $v_{ij} = v_i - v_j$. Defining $b_{ij} = r_{ij} \cdot v_{ij}$ then we obtain

$$v_{ij}^2 t_{ij}^2 + 2b_{ij}t_{ij} + r_{ij}^2 - \sigma = 0.$$

If $b_{ij} > 0$ then all the atoms are going away from each other, so no collision occurs; instead for $b_{ij} < 0$ but $b_{ij}^2 v_{ij}^2 (r_{ij}^2 - \sigma^2) < 0$ then we end up with complex solution and this means that we have no solution. Otherwize we have two positive roots from

$$t_{ij} = \frac{-b_{ij} - [b_{ij}^2 - v_{ij}^2(r_{ij}^2 - \sigma^2)]^{1/2}}{v_{ij}^2}$$

The program should store the earliest upcoming collision time for each time, and this list is called collision table. A method to handle with it consists into identify by the index i all the j particles that should collide with it, all the collision time will be saved in an array *coltime(i)* and all the j collision partenrs in *partners(i)*. Thanks the function MINLOC we can find the minimum collision time t_{ij}, and let evolve the system untill it reachest t_{ij} and the table of future collision times is adjusted accordingly. Now we focused on the second point of the program: the calculation of colliding dynamics. Considering elastic

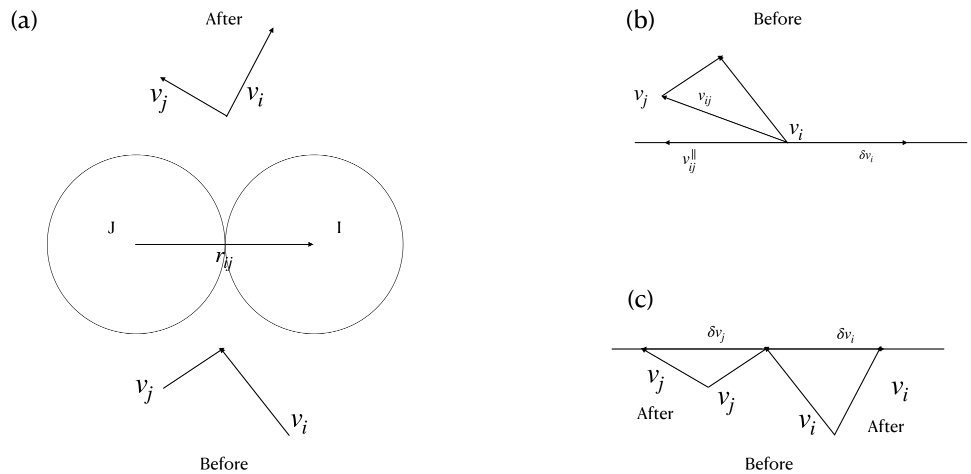

Figure 1: In (a) there is the illustration of a collision between i-th particle and j-th one. (b) vectorial construction of the velocities changing, (b) relation of initial and final velocities.

scattering betwen atoms, so energy and momentum are conserved, the change in velocity follows the relation

$$v_i(after) = v_i(before) + \delta v_i \quad v_j(after) = v_j(before) - \delta v_i$$

with $\delta v_i = -\frac{b_{ij}}{\sigma^2} r_{ij}$ evaluated during the impact. For construction, Fig 1, δv_i corresponds to the negative projection of v_{ij} along r_{ij} direction.

After calculating the collision properties, we can restart and do all the collision times again and this is very expensive from a computational point of view, but it's not important to do all again because the collision regard only two particles, i-th and j-th particles, and most of collision times and parteners will remain the same. Clearly, we must know all the next collision partners of i and j, then we have to examinate all the other atoms. Furthermore we have to look at those particles were due to collide with the particle we examine, if these two had not met each other first, for this we have to look for the partners with "higher" indices. A part from this all components of *coltime(i)* and *partners(i)* will be the same.

3 Results and conclusion

3.1 Study of distribution function of velocity and H function

Let's consider our sample as a 2D square lattice with one atom for unit cell of density 0.2 and all initial velocities have the same magnitude, for us is 1, and direction random therefore the total momentum equal 0. Our time interval is divided in blocks, similarly are made by time-step. Our purpose is to analyze the non-equilibrium proprieties of the system, and this requires that our number of time-step is one and fixed the number of block. From Fig [2] it's possible to see that after a certain time we reach equilibrium, and the distribution function is the well-known Maxwell- Boltzmann, and due to H-Theorem the H goes to his minimum value during the simulation unless it reaches the equilibrium.

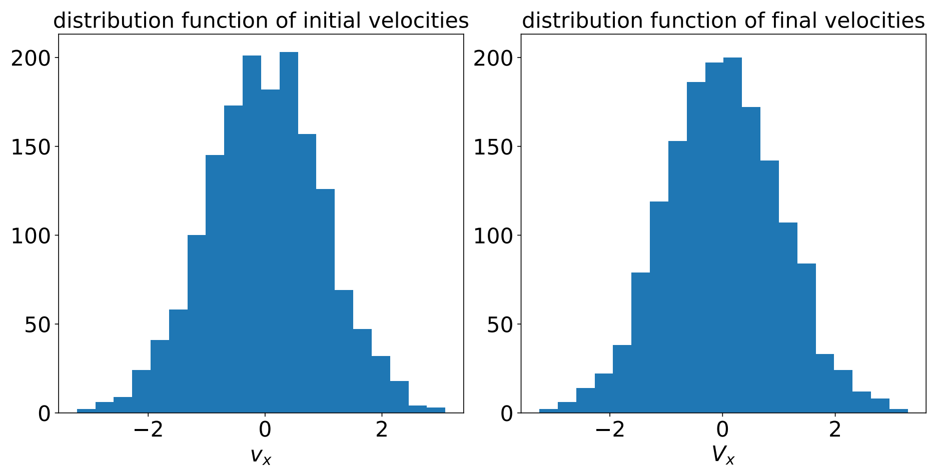

Figure 2: Distribution function of first component of velocities for the initial state and after the simulation,

The time intervall to get the equilibrium state is the 200 blocks, but for different *nblock* we invert all the velocities, studying the trend of the H function. So the scheme of this work is the following, once it's done the forward trajectory, after a certain time we invert all velocities and take as the new initial configuration. Then repeat the simulation for the same time intervall. In Fig [3] we underline that for a certain number of blocks H comes back to his maximum value, but once equilibrium is reached the function goes to a maximum value lower than the initial one. This shows that the anti-kinetics behaviour presents some kind of instability.

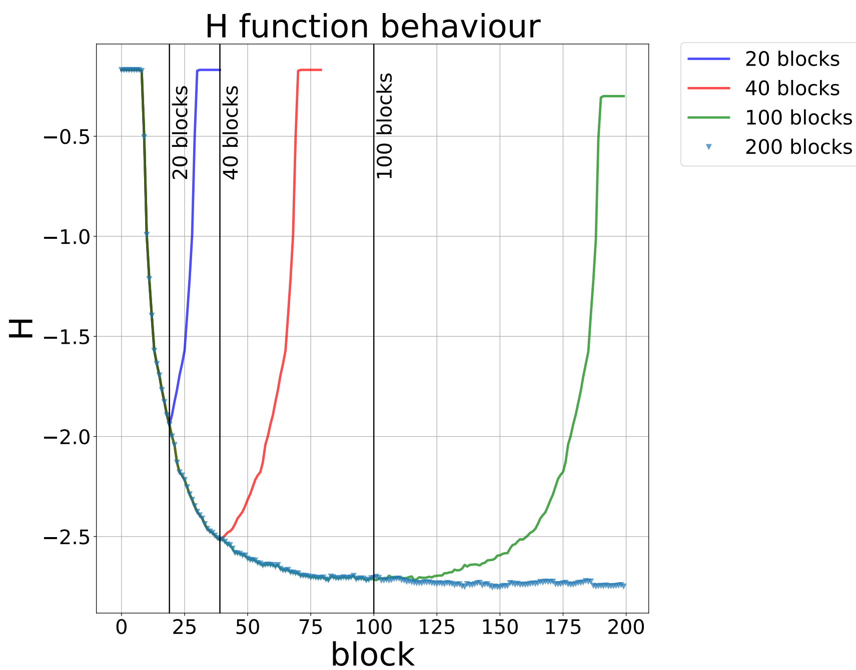

Figure 3: H function trend starting from a crystal and after with reversing of all atoms velocities for $nblock = 20, 40, 100$.

It's possible now to show directly the presition dependence of H putting some noise in the velocities inversion. Therefore we introduce an error ϵdirectly in $v = -v(1 + r\epsilon)$, where $\epsilon \in \left[-\frac{1}{2}, \frac{1}{2}\right]$ is a random number and this gives us a significant difference than before, and $r$goes from e^{-8} to e^{-2}, and this means as we decrese the r factor the inversion operation tends to the ideal one. The presence of this error in the velocities inversion, tends to simulate the instability of the anti-kinetic behaviour of H, and in particular it's underlined in Fig [4] that using different r factor the H function doesn't go to the maximum value we obtain before without ϵ.

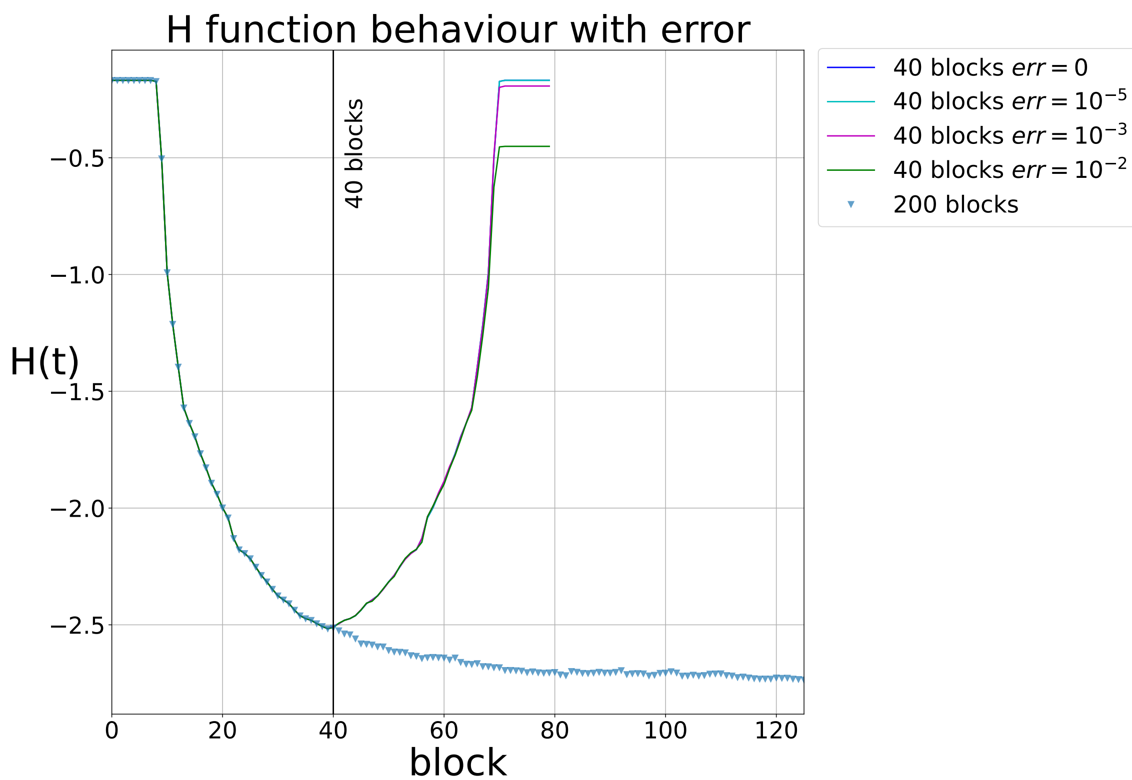

Figure 4: H function at $nbloc = 40$ it's performed the velocities inversion for different r-factor.

3.2 Study of distribution function of coordinates and mean square displacement

We also study the coordinates distribution function and in particular the mean square displacement. During the simulation we consider the periodic boundary condition, and this make the trajectory with discontineous point, therefore our first step is to correct this in order to obtain a contineous evolution for coordinates. Once this problem is resolved, we compute the difference for each time between the position and the initial one , $\langle \Delta r(t) \rangle = \langle r(t) - r_0 \rangle$, and then make the averege of the square of Δr.

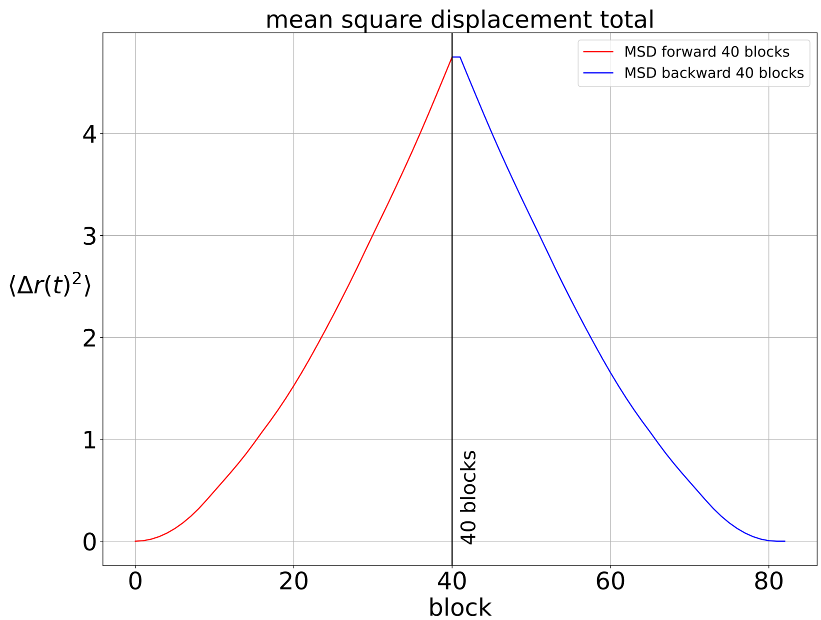

Figure 5: Plot the mean square displacement for the kinetic (blue) and anti-kinetic behaviour where the velocity inversion take place at $nblock = 40$

The same procedure we did before, or rather calculate the H function defined as before but instead of $f(\vec{p}, t)$ we used the distribution function of positions and emegers an instability for anti-kinetic behaviour, and in particular when the velocity inversion is performed at 100 blocks H doesn't go in its initial value.

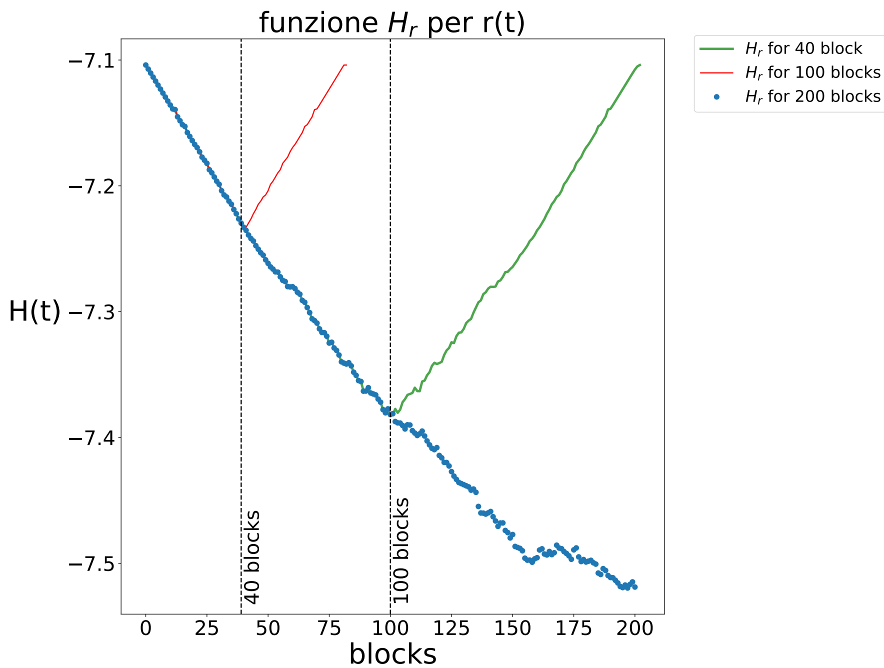

Figure 6: Plot of H with time (block number) showing the kinetic and antikinetic behaviour for velocity inversion taking place at 40 and 100 blocks.