Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Development of an Event-Based Simulator for Analysing Excluded Volume Effects in a Brownian Gas

Relatore:
Prof. Armando Bazzani

Presentata da:
Carlo Emilio Montanari

Correlatore:
Dott. Matteo Monti

Anno Accademico 2016/2017

Abstract

Il presente lavoro si pone come scopo lo sviluppo di un simulatore in C++ di dinamica molecolare utilizzando un approccio event-based, in grado di simulare la dinamica newtoniana semplice di molecole bidimensionali di forma arbitraria. Abbiamo utilizzato il simulatore NOCS per imbastire un primo tentativo di ricerca e di analisi degli effetti di volume escluso sul moto Browniano di molecole. In particolare si vogliono ricercare violazioni locali di isotropia nel moto Browniano. Nella parte teorica dell'elaborato, si analizzano gli strumenti matematici e statistici fondamentali della Kinetic Theory (teoria cinetica dei gas) ed i principali modelli della depletion force, uno dei fenomeni causati da potenziale di volume escluso.

Contents

Introduction

When studying suspensions of colloidal particles in a solvent, depletion forces and other effects caused by excluded-volume are of great interest in Chemical Physics, which studies systems with objects of different length scales and geometries.

Statistical Mechanics, where the average behaviour of a mechanical system is considered, might be modified by the particular shape of its elements.

Finite-size effects and size exclusion are known as important in many biological and molecular processes and are also suspected to be responsible for cellular organisation [7].

The most well-known effect caused by excluded volume is the break of pressure isotropy which causes in a suspension of large and small particles, an attractive force between the former caused by the pressure of the latter [9]. This phenomenon finds many practical consequences in biological scenarios such as phase separation of proteins in milk [5].

We want to analyse if some of these excluded volume effects are able to interfere within the isotropy of the Brownian motion of a molecule submerged into an hard-sphere gas. To do so, we decided to approach the problem directly by building an exact 2D gas dynamics simulator and observe from the data acquired if such violation takes place.

By using an event-based approach, we developed from scratch a an object-oriented library in C++ named NOCS (Not Only Colliding Spheres), capable to compute any Newtonian gas composed by arbitrary complex molecules.

NOCS presents optimisation tools that allows the user to obtain faster computations than a naive $\mathcal{O}(N^2)$ event-based simulation of N objects and, moreover, makes use of the advanced language tools from C++14 for offering the user easy data gathering tools.

With NOCS, we simulated the Brownian motion of a T-shaped molecule submerged into a gas of hard-spheres of different relative dimensions. We then gathered the dynamical information of every single collision that happened between the molecule and any sphere of the gas. Next, we compared that information with the data obtained from a control sample simulation in

which, instead of a T-shaped molecule, we have another hard sphere of same proportions.

After studying the local violation in isotropy found for the T-shaped molecule, we re-executed the simulations with the molecules' orientation locked down. Meaning that collisions were not able to change angular momentum but only the motion of the centre of mass. In that configuration, we tried to see if any of the local violations in isotropy observed before could lead to any relevant global violation in the Brownian motion.

Structure of the work

The present work is divided into 4 chapters.

In the first chapter, we give an overview of the main theoretical concepts of kinetic theory and some statistical and analysis tools like the Boltzmann equation and Fokker-Plank's equation [3, 6, 11]. Moreover, we give a brief analysis of Asakura-Oosawa's depletion force model [2, 1] and, as more precise alternative, we also briefly present B. Götzelmann work [4], based on density functionals.

In the second chapter, we introduce the main features and tools offered by our simulator NOCS, among with a brief conceptual analysis of its main algorithms for collision detection, collision computation and event-based integration of the system. Moreover, we present with some benchmarks how the introduction of a grid allows the user to reduce computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$ in the collision detection problem, where N is the number of simulated molecules. For accessing the source code and the complete documentation, the reader can consult [8].

In the third chapter, we present various simulations composed with NOCS and we briefly analyse the gathered results.

In the fourth chapter, we draw the final conclusions about the observed phenomenon and we discuss possible future applications for our simulator.

Chapter 1

Basic notions of kinetic theory and depletion forces

1.1 Kinetic Theory

1.1.1 Reaching the BBGKY equations

Probability distribution and Liouville's theorem

Let us consider the Hamiltonian of a system of N identical particles, where N is in the order of Avogadro's number: $N \sim 10^{23}$. We will have an Hamiltonian that is composed by three terms:

$$H = \frac{1}{2m}\sum_{i=1}^{N}\mathbf{p}_i^2 + \sum_{i=1}^{N}V(\mathbf{r}_i) + \sum_{i<j}U(\mathbf{r}_i - \mathbf{r}_j) \tag{1.1}$$

one term for the kinetic energies of the particles, one term for an external force $\mathbf{F} = -\nabla V$ that acts on the particles, and one last term for binary interactions among particles. From now on, we will refer to $\sum_i \mathbf{r}_i$ and $\sum_i \mathbf{p}_i$ with $\mathbf{v}_i$ and $\mathbf{p}_i$ for convenience.

As first step, we want to build a probability distribution of the system over the $6N$ phase space in the form $f(\mathbf{r}_i; \mathbf{p}_i; t)$, that could tell us the probability to find the system in the elementary volume dV at the point $(\mathbf{r}_i, \mathbf{p}_i)$. We want this function to be normalised as

$$\int f(\mathbf{r}_i; \mathbf{p}_i; t)dV = 1 \qquad \text{where} \qquad dV = \prod_{i=1}^{N} d^3r_i d^3p_i. \tag{1.2}$$

Furthermore, f have to respect local conservation of particles and obey

to a continuity equation. Considering now that Hamilton's equations

$$\frac{\partial \mathbf{p}_i}{\partial t} = -\frac{\partial H}{\partial \mathbf{r}_i} \quad , \quad \frac{\partial \mathbf{r}_i}{\partial t} = \frac{\partial H}{\partial \mathbf{p}_i} \tag{1.3}$$

we can use them to express f local conservation over the phase space and obtain

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial \mathbf{r}_i} \cdot \frac{\partial H}{\partial \mathbf{p}_i} - \frac{\partial f}{\partial \mathbf{p}_i} \cdot \frac{\partial H}{\partial \mathbf{r}_i} = 0 \tag{1.4}$$

which is the famous *Liouville's equation*, that states that $\frac{df}{dt} = 0$ i.e. the distribution function $f(\mathbf{r}_i; \mathbf{p}_i; t)$ has to be invariant along the phase flow of the system. If we switch to *Poisson* notation, we can rewrite Liouville's equation simply as:

$$\frac{\partial f}{\partial t} = \{H, f\}. \tag{1.5}$$

Now that we have defined the main characteristics of a probability distribution, we can use it to compute expectation values of any observable functions defined on the phase space in the form $A(\mathbf{r}_i, \mathbf{p}_i)$:

$$\langle A \rangle(t) = \int A(\mathbf{r}_i, \mathbf{p}_i) f(\mathbf{r}_i; \mathbf{p}_i; t) dV. \tag{1.6}$$

Let's say that f is an *equilibrium distribution* or, in other words, a distribution which has no explicit time dependence

$$\frac{\partial f}{\partial t} = 0 \tag{1.7}$$

that implies $\{H, f\} = 0$ and f is a first integral of motion. As a consenquence in the equilibrium state $\langle A \rangle$ will not change during time (that means that in equilibrium $\langle A \rangle$ is constant).

If we have instead explicit time dependence in the distribution, we have

$$\begin{aligned}
\frac{d\langle A \rangle}{dt} &= \int A \frac{\partial f}{\partial t}\, dV \\
&= \int A \left(-\frac{\partial f}{\partial \mathbf{r}_i} \cdot \frac{\partial H}{\partial \mathbf{p}_i} + \frac{\partial f}{\partial \mathbf{p}_i} \cdot \frac{\partial H}{\partial \mathbf{r}_i} \right) dV \\
&= \int \left(\frac{\partial A}{\partial \mathbf{r}_i} \cdot \frac{\partial H}{\partial \mathbf{p}_i} - \frac{\partial A}{\partial \mathbf{p}_i} \cdot \frac{\partial H}{\partial \mathbf{r}_i} \right) f\, dV \\
&= \int \{A, H\} f\, dV = \langle \{A, H\} \rangle
\end{aligned}$$

where we integrated by parts and used the fact that f is normalised and, therefore, $f = 0$ at the boundary condition.

The Hamiltonian character of the dynamics implies no relaxation process of the distribution function towards the equilibrium distribution and the existence of the time arrow is a open problem for isolated Hamiltonian statistical systems.

BBGKY hierarchy

Since it is impossible to compute directly a probability distribution that involves $\sim 10^{24}$ variables, we need to to proceed with an approximation procedure. Instead of working on a probability distribution for all N particles at the same time, we will introduce the one-particle distribution function $f_1(\mathbf{r};\mathbf{p};t)$, defined as

$$f_1(\mathbf{r};\mathbf{p};t) = N \int \prod_{i=2}^{N} d^3r_i d^3p_i \, f(\mathbf{r}, \mathbf{r}_2, \cdots, \mathbf{r}_N, \mathbf{p}, \mathbf{p}_2, \cdots, \mathbf{p}_N; t). \tag{1.8}$$

Since all N particles are identical, there is no reason to expect a different result if one change the choice of the first particle. The factor N on the right side has normalisation purposes since we have

$$\int f_1(\mathbf{r}, \mathbf{p}; t)\, d^3r d^3p = N. \tag{1.9}$$

Assuming that a single particle is representative of the whole system, the function f_1 allows us to easily gain many properties of the entire system (such as average density of particles in physical space, average velocity and energy flux) we want to obtain an equation governing f_1. If we derive partially for the time we get:

$$\frac{\partial f_1}{\partial t} = N \int \prod_{i=2}^{N} \frac{\partial f}{\partial t}\, d^3r_i d^3p_i = N \int \prod_{i=2}^{N} \{H, f\}\, d^3r_i d^3p_i \tag{1.10}$$

and, if we explicitate the Hamiltonian of equation (1.1) and then simplify the Hamiltonian's null derivatives,

$$\frac{\partial f_1}{\partial t} = N \int \prod_{i=2}^{N} \left(-\sum_{j=1}^{N} \frac{\mathbf{p}_j}{m} \cdot \frac{\partial f}{\partial \mathbf{r}_j} + \sum_{j=1}^{N} \frac{\partial V}{\partial \mathbf{p}_j} + \sum_{j=1}^{N} \sum_{k<l} \frac{\partial U(\mathbf{r}_k - \mathbf{r}_l)}{\partial \mathbf{r}_j} \cdot \frac{\partial f}{\partial \mathbf{p}_j} \right) d^3r_i d^3p_i \tag{1.11}$$

and, by integrating by parts we obtain zero in the first two terms for all the $j = 2, \ldots, N$ and we reach the final form

$$\frac{\partial f_1}{\partial t} = N\{H_1, f_1\} + N \int \prod_{i=2}^{N} d^3r_i d^3p_i \sum_{k=2} N \frac{\partial U(\mathbf{r} - \mathbf{r}_k)}{\partial \mathbf{r}} \cdot \frac{\partial f}{\partial \mathbf{p}} \tag{1.12}$$

where $\mathbf{r} \equiv \mathbf{r}_1$, $\mathbf{p} \equiv \mathbf{p}_1$ and $H_1 = \frac{p^2}{2m} + V(\mathbf{r})$ is the one-particle Hamiltonian.

That means that for $\frac{\partial f_1}{\partial t}$ satisfies to an equation that resembles Liouville's equation, with an extra term:

$$\frac{\partial f_1}{\partial t} = \{H_1, f_1\} + \left(\frac{\partial f_1}{\partial t}\right)_{\text{coll}} \tag{1.13}$$

where the first term is referred as *streaming term* and the second term as *collision integral* and it is defined by the second term in equation (1.12). Since we are dealing with a gas of identical particles we can reformulate the $(N-1)$ terms in the summation and obtain

$$\left(\frac{\partial f_1}{\partial t}\right)_{\text{coll}} = N(N-1)\int d^3r_2 d^3p_2 \, \frac{\partial U(\mathbf{r}-\mathbf{r}_2)}{\partial \mathbf{r}} \cdot \frac{\partial}{\partial \mathbf{p}} \int \prod_{i=3}^{N} d^3r_i d^3p_i \, f(\mathbf{r}, \ldots, \mathbf{p}, \ldots; t). \tag{1.14}$$

But, as we can immediately see, it is not possible to express the collision integral only in terms of the one-particle distribution function. This is quite obvious since, by definition, the one-particle distribution describes only the evolution of a single particle per time and the collision integral captures the binary intercations among the particles. We can deal with this problem by introducing the many-particle distributions.

If we define the *two-particle distribution function* as

$$f_2(\mathbf{r}_1, \mathbf{r}_2, \mathbf{p}_1, \mathbf{p}_2; t) \equiv N(N-1)\int \prod_{i=3}^{N} f(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{p}_1, \mathbf{p}_2, \ldots; t)\, d^3r_i d^3p_i \tag{1.15}$$

we can rewrite the collision integral as

$$\left(\frac{\partial f_1}{\partial t}\right)_{\text{coll}} = \int \frac{\partial U(\mathbf{r}-\mathbf{r}_2)}{\partial \mathbf{r}} \cdot \frac{\partial f_2}{\partial \mathbf{p}} \tag{1.16}$$

but, we can see that if we repeat on f_2 the calculations made on f_1 we will end up again by showing that f_2 evolves by a Liouville-like equation with an extra corrected term that, this time, depends on a *three-particle distribution function* f_3. Also f_3 will evolve in a Liouville-like manner, but with a correction term depending on an f_4 four-particle distribution, and so on with this recursive process.

In general, the n-particle distribution function defined as

$$f(\mathbf{r}, \cdots, \mathbf{r}_n, \mathbf{p}, \cdots, \mathbf{p}_n; t) = \frac{N!}{(N-n)!}\int \prod_{i=n+1}^{N} f(\mathbf{r}, \cdots, \mathbf{r}_N, \mathbf{p}, \cdots, \mathbf{p}_N; t) \tag{1.17}$$

obeys the equation

$$\frac{\partial f_n}{\partial t} = \{H_n, f_n\} + \sum_{i=1}^{n} \int \frac{\partial U(\mathbf{r}_i - \mathbf{r}_{n+1})}{\partial \mathbf{r}_i} \cdot \frac{\partial f_{n+1}}{\partial \mathbf{p}_i} \tag{1.18}$$

where we have defined an n-body Hamiltonian as

$$H_n = \sum_{i=1}^{n} \left(\frac{\mathbf{p}_i^2}{2m} + V(\mathbf{r}_i) \right) + \sum_{i<j\leq n} U(\mathbf{r}_i - \mathbf{r}_j) \tag{1.19}$$

The meaning of equations (1.18), known as **BBGKY hierarchy**, is that any group of n particles evolves in a Hamiltonian-like dynamic, with the correction of an interaction integral with one of the particles outside of the group. By reformulating the problem in these terms, we have, instead of the Liouville equation indicating a function f depending on $N \sim 10^{24}$ variables, a set of $N \sim 10^{24}$ coupled equations.

Apparently, this reformulation gives no benefits for solving completely the dynamic of the system, but it gives us a very practical approach that is philosophically analogous to a Taylor truncation. Since most interesting properties are isolated into f_1 and other lower f_n, we are authorised, depending on the problem's nature, to ignore the effects of higher terms and operate a truncation at a manageable level.

The most simple and famous of these truncations is the Boltzmann equation.

1.1.2 The Boltzmann equation

Let's consider a problem of colliding particles characterised by two time scales: one time scale τ for representing the average time between collisions, and the collision time scale τ_{coll}, which is the time it takes for a collision between particles to occur. Moreover, let's also assume that the variation of the external potential V can only be appreciated at macroscopic scales and neglected on the scale of atomic collisions.

If we have

$$\tau \gg \tau_{\text{coll}} \tag{1.20}$$

we can expect the first term of the BBGKY hierarchy f_1 to be a fine enough descriptor of the system, since it will simply follow an Hamiltonian evolution with perturbations due to binary collisions. For example, in a dilute enough gas expect to see this regime and it is extremely improbable to see a multiple collision.

As we said above, since we want to make a truncation of the BBGKY hierarchy at the first term, the equation have the form

$$\frac{\partial f_1}{\partial t} = \{H_1, f_1\} + \left(\frac{\partial f_1}{\partial t}\right)_{\text{coll}} \tag{1.21}$$

but, this time, we want to find an expression for the collision integral in terms of f_1. In this problem where (1.20) holds, the collisional term represents the change in momenta caused by two-particle scattering; we need to describe this phenomenon.

For simplicity, let us assume point-like particles that, when colliding, they are at the same position. If we consider a particle at position $(\mathbf{r}, \mathbf{p}_1)$ on the phase space colliding with another particle at position $(\mathbf{r}, \mathbf{p}_2)$, these two particle will emerge after the collision with momenta $\mathbf{p}_1'$ and $\mathbf{p}_1'$. Let us define the rate of this phenomenon as the number of collision per unit time in the position r

$$\text{Rate} = \omega(\mathbf{p}_1, \mathbf{p}_2|\mathbf{p}_1', \mathbf{p}_2') f_2(\mathbf{r}, \mathbf{r}, \mathbf{p}_1, \mathbf{p}_2)\, d^3p_2 d^3p_1' d^3p_2' \tag{1.22}$$

(time dependence is implicit for lighter notation). We have here a scattering function ω containing the information about the dynamic of the collision. With standard classical mechanic techniques, one can reformulate a given inter-atomic potential $U(\mathbf{r})$ into a scattering function ω. In this context, ω is proportional to the two-body distribution function f_2 and tells us the chance to find two starting particles in position $(\mathbf{r}, \mathbf{p}_1)$ and $(\mathbf{r}, \mathbf{p}_2)$.

Focusing now on the distribution of a test particle with specified momentum $\mathbf{p}$, we can take advantage on the fact that, in a collision, both momenta and energy are conserved, because we can neglect the external potential contribution

$$\mathbf{p} + \mathbf{p}_2 = \mathbf{p}_1' + \mathbf{p}_2' \tag{1.23}$$

$$\mathbf{p}'^2 + \mathbf{p}_2'^2 = \mathbf{p}_1'^2 + \mathbf{p}_2'^2 \tag{1.24}$$

these basic notions are helpful when we work on the scattering function.

Moreover, while collisions can change the momentum of a particle from $\mathbf{p}$ to another value, they can also deflect particles from other states to a state with momentum $\mathbf{p}$. If we consider these facts, we can formulate the collision integral in the two-terms form

$$\begin{aligned}\left(\frac{\partial f_1}{\partial t}\right)_{\text{coll}} = \int [&\omega(\mathbf{p}_1', \mathbf{p}_2'|\mathbf{p}, \mathbf{p}_1) f_2(\mathbf{r}, \mathbf{r}, \mathbf{p}_1', \mathbf{p}_2') \\ &- \omega(\mathbf{p}, \mathbf{p}_2|\mathbf{p}_1', \mathbf{p}_2') f_2(\mathbf{r}, \mathbf{r}, \mathbf{p}, \mathbf{p}_2)]\, d^3p_2 d^3p_1' d^3p_2'\end{aligned}$$

where the first term indicates scattering into the state $\mathbf{p}$, the second scattering out of state $\mathbf{p}$.

Now, we know that the scattering function has to respect some simple requirements:

1. ω is only non-vanishing for scattering events which satisfy equations (1.23) and (1.24).

2. Due to spacetime's discrete symmetries, ω is invariant under time reversal, therefore we have

$$\omega(\mathbf{p}, \mathbf{p}_2|\mathbf{p}_1', \mathbf{p}_2') = \omega(-\mathbf{p}_1', -\mathbf{p}_2'| - \mathbf{p}, -\mathbf{p}_2) \tag{1.25}$$

3. ω is invariant under parity $(\mathbf{r}, \mathbf{p}) \to (-\mathbf{r}, -\mathbf{p})$, which implies that any scattering process which is parity invariant obeys

$$\omega(\mathbf{p}, \mathbf{p}_2|\mathbf{p}_1', \mathbf{p}_2') = \omega(-\mathbf{p}, -\mathbf{p}_2| - \mathbf{p}_1', -\mathbf{p}_2') \tag{1.26}$$

4. As a result of equation (1.25) combined with (1.26), we have this other invariant fact

$$\omega(\mathbf{p}, \mathbf{p}_2|\mathbf{p}_1', \mathbf{p}_2') = \omega(\mathbf{p}_1', \mathbf{p}_2'|\mathbf{p}, \mathbf{p}_2) \tag{1.27}$$

5. Since space is isotropic, we assume also translational invariance for ω since $-\mathbf{r}$ is analogous to $\mathbf{r}$.

Equation (1.27) allows us to rewrite the collision integral as

$$\left(\frac{\partial f_1}{\partial t}\right)_{\text{coll}} = \int \omega(\mathbf{p}, \mathbf{p}_2|\mathbf{p}_1', \mathbf{p}_2')[f_2(\mathbf{r}, \mathbf{r}, \mathbf{p}_1', \mathbf{p}_2') - f_2(\mathbf{r}, \mathbf{r}, \mathbf{p}, \mathbf{p}_2)]\, d^3p_2 d^3p_1' d^3p_2' \tag{1.28}$$

which is a simplification of the initial expression but we have to express this integral in terms of f_1 rather than f_2. To do this, we have to make the important assumption of **molecular chaos**, which states that the velocities of two colliding particles are uncorrelated random variables before the collision. With this extremely important assumption we can write

$$f_2(\mathbf{r}, \mathbf{r}, \mathbf{p}, \mathbf{p}_2) = f_1(\mathbf{r}, \mathbf{p}) f_1(\mathbf{r}, \mathbf{p}_2) \tag{1.29}$$

This assumption has extremely dramatic implication in our treatment, the most immediate one is that we have introduced an arrow of time into the system since it introduces a mechanism to cancel the memory of the past dynamics. In other words, we somehow broke time isotropy and placed down the basis for the H-theorem that we will see in section 1.1.3.

With all that said, we finally reach our final expression of the collision integral, also known as **Boltzmann equation**

$$\left(\frac{\partial f_1}{\partial t}\right)_{\text{coll}} = \int \omega(\mathbf{p}, \mathbf{p}_2|\mathbf{p}_1', \mathbf{p}_2') \times \left[f_1(\mathbf{r}, \mathbf{p}_1')f_1(\mathbf{r}, \mathbf{p}_2') - f_1(\mathbf{r}, \mathbf{p})f_1(\mathbf{r}, \mathbf{p}_2)\right] d^3p_2 d^3p_1' d^3p_2' \quad (1.30)$$

and, as it was our initial purpose, we finally obtain a valid closed expression for the one-particle distribution function given by equation (1.21).

Equilibrium and detailed balance

Now that we have our expression for f_1, we want to explore the form of the equilibrium distribution obeying $\partial f^{\text{eq}}/\partial t = 0$. Considering equation (1.21), we know that $\{f, H_1\} = 0$ implies that f is an integral of motion so that if the only integral is the energy we expect $f = f(E)$. In the condition $V(r) = 0$, any function of momentum is an equilibrium distribution. If we are working in a scenario where $V(r) = 0$, any function of momentum Poisson commutes as we want, still, we have to look for any contributions from the collision integral.

If the distribution obeys the *detailed balance* condition

$$f_1^{\text{eq}}(\mathbf{r}, \mathbf{p}_1')f_1^{\text{eq}}(\mathbf{r}, \mathbf{p}_2') = f_1^{\text{eq}}(\mathbf{r}, \mathbf{p})f_1^{\text{eq}}(\mathbf{r}, \mathbf{p}_2) \quad (1.31)$$

or, alternatively,

$$\log(f_1^{\text{eq}}(\mathbf{r}, \mathbf{p}_1')) + \log(f_1^{\text{eq}}(\mathbf{r}, \mathbf{p}_2')) = \log(f_1^{\text{eq}}(\mathbf{r}, \mathbf{p})) + \log(f_1^{\text{eq}}(\mathbf{r}, \mathbf{p}_2)) \quad (1.32)$$

the collision integral vanishes and gives no contributions.

The right side terms are the momenta before the collision, the left side terms are the momenta after the collision. Therefore, we must have that this sum must be conserved during the collision. Since we already know that momentum and energy are conserved during a collision (equations (1.23) and (1.24)) we have the general form

$$\log(f_1^{\text{eq}}(\mathbf{r}, \mathbf{p})) = \beta(\mu - E(\mathbf{p}) + \mathbf{u} \cdot \mathbf{p}) \quad (1.33)$$

where $E(\mathbf{p}) = p^2/2m$ for non-relativistic frameworks, β and $\mathbf{u}$ are constants and μ is a normalisation constant to ensure that (1.9) holds. Writing $\mathbf{p} = m\mathbf{v}$, we obtain

$$f_1^{\text{eq}}(\mathbf{r}, \mathbf{p}) = \frac{N}{V}\left(\frac{\beta}{2\pi m}\right)^{3/2} e^{-\beta m(\mathbf{v}-\mathbf{u})^2/2} \quad (1.34)$$

which, if β is the inverse temperature, is the Maxwell-Boltzmann distribution. Therefore we discover that the introduction of the collision term inside the Liouville equation implies, under the assumption of molecular chaos, that the Boltzmann distribution tends to the equilibrium.

1.1.3 H-theorem

One of the fundamental observation of thermodynamics is that any system will eventually relaxes to equilibrium. This concept seems incompatible with time invariant classical mechanics, yet, we will see now how within the framework of Boltzmann equation, systems do settle down to equilibrium.

The starting point is equation (1.29), where we have introduced a time arrow by saying that particle velocities are uncorrelated before collisions. By using only this assumption we want to demonstrate the "H-theorem", named after the quantity H introduced by Boltzmann defined as

$$H(t) = \int f_1(\mathbf{r}, \mathbf{p}; t) \log(f_1(\mathbf{r}, \mathbf{p}; t))\, d^3r d^3p \tag{1.35}$$

The H-theorem states that H always decreases with time. Proof will follow below.

As seen before, when computing the variation of expectation values, only explicit time dependence is important, therefore

$$\frac{dH}{dt} = \int (\log f_1 + 1) \frac{\partial f_1}{\partial t}\, d^3r d^3p = \int \log f_1 \frac{\partial f_1}{\partial t}\, d^3r d^3p \tag{1.36}$$

where we dropped the $+1$ term because $\int f_1 = N$ does not change with time. If we insert Boltzmann equation (1.30) we obtain

$$\frac{dH}{dt} = \int \log f_1 \left(\frac{\partial V}{\partial \mathbf{r}} \cdot \frac{\partial f_1}{\partial \mathbf{p}} - \frac{\mathbf{p}}{m} \cdot \frac{\partial f_1}{\partial \mathbf{r}} + \left(\frac{\partial f_1}{\partial t} \right)_{\text{coll}} \right) d^3r d^3p \tag{1.37}$$

Where the first two terms inside the parenthesis vanish, thus we obtain that H variation is governed only by the collision terms

$$\begin{aligned} \frac{dH}{dt} &= \int \log f_1 \left(\frac{\partial f_1}{\partial t} \right)_{\text{coll}} d^3r d^3p \\ &= \int \omega(\mathbf{p}_1', \mathbf{p}_2' | \mathbf{p}_1, \mathbf{p}_2) \log f_1(\mathbf{p}_1) \\ &\quad \times [f_1(\mathbf{p}_1') f_1(\mathbf{p}_2') - f_1(\mathbf{p}_1 f_1(\mathbf{p}_2))]\, d^3r d^3p_1 d^3p_2 d^3p_1' d^3p_2' \end{aligned} \tag{1.38}$$

Note that in the explication we have suppressed $\mathbf{r}$ term and t terms since we are only interested with momenta. Now, considering that all momenta are dummy variables we can write

$$\begin{aligned} \frac{dH}{dt} &= \frac{1}{2} \int \omega(\mathbf{p}_1', \mathbf{p}_2' | \mathbf{p}_1, \mathbf{p}_2) \log[f_1(\mathbf{p}_1) f_1(\mathbf{p}_2)] \\ &\quad \times [f_1(\mathbf{p}_1') f_1(\mathbf{p}_2') - f_1(\mathbf{p}_1 f_1(\mathbf{p}_2))]\, d^3r d^3p_1 d^3p_2 d^3p_1' d^3p_2' \end{aligned} \tag{1.39}$$

Moreover, since all momenta are integrated, we can make another swap and make use of symmetry property (1.27). We get

$$\frac{dH}{dt} = -\frac{1}{2}\int \omega(\mathbf{p}_1', \mathbf{p}_2'|\mathbf{p}_1, \mathbf{p}_2)\log[f_1(\mathbf{p}_1')f_1(\mathbf{p}_2')] \times [f_1(\mathbf{p}_1')f_1(\mathbf{p}_2') - f_1(\mathbf{p}_1 f_1(\mathbf{p}_2))]\, d^3r d^3p_1 d^3p_2 d^3p_1' d^3p_2' \quad (1.40)$$

In the end, combining these last two equations, we get

$$\frac{dH}{dt} = -\frac{1}{4}\int \omega(\mathbf{p}_1', \mathbf{p}_2'|\mathbf{p}_1, \mathbf{p}_2) \times (\log[f_1(\mathbf{p}_1')f_1(\mathbf{p}_2')] - \log[f_1(\mathbf{p}_1)f_1(\mathbf{p}_2)]) \times [f_1(\mathbf{p}_1')f_1(\mathbf{p}_2') - f_1(\mathbf{p}_1 f_1(\mathbf{p}_2))]\, d^3r d^3p_1 d^3p_2 d^3p_1' d^3p_2' \quad (1.41)$$

The main part of this expression can be reformulated as $(\log x - \log y)(x - y)$ which is positive for every value of x and y. Since the scattering rate ω is always positive, we have obtained that

$$\frac{dH}{dt} \leq 0 \quad (1.42)$$

Thus, we have proved the H-theorem. Showing us that we have clearly "broken" time invariance with our assumption of molecular chaos (1.29).

1.2 Stochastic Processes

We will now focus on the phenomenon of the *Brownian motion*, or better, on the mathematical formalism used to model it and its emerging random phenomenon. The stochastic processes provide mathematical models to introduce the molecular chaos assumption in the microscopic dynamics of particles.

1.2.1 Langevin equation

Let's start by modelling a single particle submerged inside a background medium, as long as we know which forces F act on it, its motion is deterministic, governed by

$$m\ddot{\mathbf{x}} = -\gamma\dot{\mathbf{x}} + \mathbf{F} \quad (1.43)$$

where γ is a friction coefficient. This is not an Hamiltonian system. γ is related to the viscosity of the surrounded liquid η. This relation depends on the shape of the particle and, if we model the particle as a sphere of radius

a, we know from Stokes formula that $\gamma = 6\pi\eta a$. From now on γ will be a fixed parameter.

If F is time-independent, we have a naive steady-state solution $\ddot{\mathbf{x}} = 0$ with

$$\dot{\mathbf{x}} = \frac{1}{\gamma}\mathbf{F} \tag{1.44}$$

where $1/\gamma$ is usually called *mobility*.

The main problem with equation (1.43) is that describes a completely deterministic motion (as long as we know the nature of $\mathbf{F}$) where instead the Brownian motion presents random behaviour. We resolve this by introducing a randomic component inside the expression of $\mathbf{F}$

$$\mathbf{F} = -\nabla V + \mathbf{f}(t) \tag{1.45}$$

where V is the fixed background potential in which the particle is submerged and $\mathbf{f}(t)$ is a random force experienced by the particle, also referred as *noise*. The final form of equation (1.43) is the *Langevin equation*

$$m\ddot{\mathbf{x}} = -\gamma\dot{\mathbf{x}} - \nabla V + \mathbf{f}(t) \tag{1.46}$$

which is classified as a *stochastic differential equation*. Equations of this kind, in order to be solved, needs at least some information about the noise nature and form, information that can be extracted by elements like the average of $\mathbf{x}(t)$ and so on. Let's see some basic situations.

1.2.2 Diffusion in a very viscous fluid

To keep things simple and manageable, we consider a vanishing potential $V = 0$. If the Brownian motion happens in a very viscous fluid, we observe that the inertial term cease to have any effect and only the friction term dominates. We can simplify (1.46) to

$$\dot{\mathbf{x}}(t) = \frac{1}{\gamma}\mathbf{f}(t) \tag{1.47}$$

by just setting $m = 0$. This equation can be trivially integrated into

$$\mathbf{x}(t) = \mathbf{x}(0) + \frac{1}{\gamma}\int_0^t \mathbf{f}(t')\,dt' \tag{1.48}$$

but, at this point, we must have some information about the nature of the noise $\mathbf{f}(t)$, or at least make valid assumptions.

For example, when working with averages, if we assume that

$$\langle \mathbf{f}(t) \rangle = 0 \tag{1.49}$$

we can immediately obtain the well known result

$$\langle \mathbf{x}(t) \rangle = \mathbf{x}(0) \tag{1.50}$$

We can also look at the position variance

$$\left\langle (\mathbf{x}(t) - \mathbf{x}(0))^2 \right\rangle \tag{1.51}$$

in order to get information about particles' average speed. If we take equation (1.48) and, after squaring it, take its average, we obtain

$$\left\langle (\mathbf{x}(t) - \mathbf{x}(0))^2 \right\rangle = \frac{1}{\gamma^2} \int_0^t dt_1' \int_0^t dt_2' \, \langle \mathbf{f}(t_1') \cdot \mathbf{f}(t_2') \rangle \tag{1.52}$$

which requires us to have more information about noise correlation for given time deltas.

Now, what we know about Brownian motion is that the kicks given by the noise are fast and uncorrelated. We can formalise that by saying that, if a collision between our particle and an atom of the Brownian bath takes τ_{coll} seconds, we will have obvious correlation for time scales $t \ll \tau_{\text{coll}}$ while instead we will have

$$\langle f_i(t_1) f_j(t_2) \rangle = 0 \quad \text{when} \quad t_2 - t_1 \gg \tau_{\text{coll}} \tag{1.53}$$

where for the correlation function we are using index notation $i, j = 1, 2, 3$.

That means that we can only actually work with time scales $t_2 - t_1 \ll \tau_{\text{coll}}$ and take the limit $\tau_{\text{coll}} \to 0$. This limit allows us to reformulate the correlation function as

$$\langle f_i(t_1) f_j(t_2) \rangle = 2D\gamma^2 \delta_{ij} \delta(t_2 - t_1) \tag{1.54}$$

Where D is called *diffusion constant* and γ^2 was placed for convenience for working well with equation (1.52). Noises which obey (1.49) and (1.54) are also called *white noise*.

We obtain then that the variance of a white noise is

$$\left\langle (\mathbf{x}(t) - \mathbf{x}(0))^2 \right\rangle = 6Dt \tag{1.55}$$

Which implies that the root-mean square of the distance increases as $\sqrt{t}$ with time. This is typical diffusion phenomenon.

1.2.3 Diffusion in a less viscous liquid

This time we will consider (1.46) with also the inertia term, so $m \neq 0$. Still, we set $V = 0$.

If this were a standard differential equation, it would be straightforward to solve, if we multiply both sides for an integrating factor $e^{\gamma t/m}$ we have

$$\frac{d}{dt}\left(\dot{\mathbf{x}}e^{\gamma t/m}\right) = \frac{1}{m}\mathbf{f}(t)e^{\gamma t/m} \tag{1.56}$$

which can be integrated into

$$\dot{\mathbf{x}}(t) = \dot{\mathbf{x}}(0)e^{-\gamma t/m} + \frac{1}{m}\int_0^t \mathbf{f}(t')e^{\gamma(t'-t)/m}\,dt' \tag{1.57}$$

But from this point, since this is a stochastic equation, we can only work with (1.49) and (1.54).

Casting (1.49), we find immediately that the average velocity is the velocity of a particle in absence of noise

$$\langle\dot{\mathbf{x}}(t)\rangle = \dot{\mathbf{x}}(0)e^{-\gamma t/m} \tag{1.58}$$

Moreover, for the average position, we have

$$\langle\mathbf{x}(t)\rangle = \mathbf{x}(0) + \int_0^t \langle\dot{\mathbf{x}}(t')\rangle\,dt' \tag{1.59}$$

$$= \mathbf{x}(0) + \frac{m}{\gamma}\dot{\mathbf{x}}(0)e^{-\gamma t/m} \tag{1.60}$$

Which is still a predictable result.

When we work with quadratic quantities and (1.54), both for space and velocity. We have that

$$\begin{aligned}\langle\dot{x}_i(t_1)\dot{x}_j(t_2)\rangle &= \langle\dot{x}_i(t_1)\rangle\langle\dot{x}_j(t_2)\rangle \\ &+ \frac{1}{m^2}\int_0^{t_1} dt_1' \int_0^{t_2} dt_2'\,\langle f_i(t_1')f_j(t_2')\rangle\, e^{\gamma(t_1'+t_2'-t_1-t_2)/m}\end{aligned} \tag{1.61}$$

where we excluded linear terms for $\mathbf{f}$ because of $\langle\mathbf{f}(t)\rangle = 0$. Using at this point (1.54) and assuming $t_2 \geq t_1 > 0$ we obtain

$$\langle\dot{x}_i(t_1)\dot{x}_j(t_2)\rangle = \langle\dot{x}_i(t_1)\rangle\langle\dot{x}_j(t_2)\rangle + \frac{D\gamma}{m}\delta_{ij}(e^{-\gamma(t_2-t_1)/m} - e^{-\gamma(t_1+t_2)/m}) \tag{1.62}$$

For $t_1, t_2 \to \infty$, we can exclude the last term and also the average velocities. We learn finally that velocities correlation drops exponentially as

$$\langle\dot{x}_i(t_1)\dot{x}_j(t_2)\rangle \to \frac{D\gamma}{m}\delta_{ij}e^{-\gamma(t_2-t_1)/m} \tag{1.63}$$

We can then say that the particle has still some memory of its velocity at time t_1 for any time $t_2 < t_1 + m/\gamma$. After that amount of time, we lose any correlation.

With this result, we can also compute the average velocity-squared, obviously connected with the kinetic energy of the system, independent of time

$$\langle \dot{\mathbf{x}}(t) \cdot \dot{\mathbf{x}}(t) \rangle = \frac{3D\gamma}{m} \tag{1.64}$$

As for position correlation $\langle x_i(t_1)x_j(t_2)\rangle$ it is possible to prove that we get the same expression of the viscous liquid

$$\left\langle (\mathbf{x}(t) - \mathbf{x}(0))^2 \right\rangle = 6Dt \tag{1.65}$$

with the same consequential behaviours.

1.2.4 Einstein relation

As we saw in equation (1.64), the average kinetic energy of a particle is

$$E = \frac{1}{2} m \langle \dot{\mathbf{x}} \cdot \dot{\mathbf{x}} \rangle = \frac{3}{2} D\gamma \tag{1.66}$$

But we also know, from the *equipartition theorem*, that the average energy depends only on the system temperature

$$E = \frac{3}{2} k_B T \tag{1.67}$$

Therefore, we must have

$$D = \frac{k_B T}{\gamma} \tag{1.68}$$

Which is also known as *Einstein relation*.

1.3 Fokker-Plank Equation

In the previous sections, we treated position uncertainty of a particle working with correlation functions. Now we want to see if we can work the other way around: we ask what probability distribution $P(\mathbf{x}, t; \mathbf{x}_0, t_0)$ would return the same correlation function that would arose from a Langevin equation?

Instead of looking for the path $\mathbf{x}(t)$ the particle took, we try to express the probability that the particle sits at $\mathbf{x}$ at time t, regardless of the path it took. As for the formal definition of such distribution, we start by denoting

the solution to the Langevin equation for a given noise function $\mathbf{f}$ as $\mathbf{x}_f$, now, if the noise function is known, we have no uncertainty and clearly we have $P(\mathbf{x}, t) = \delta(\mathbf{x} - \mathbf{x}_f)$. From that consideration, we can consider the average of all possible noise and define the distribution as

$$P(\mathbf{x}, t) = \langle(\mathbf{x} - \mathbf{x}_f)\rangle \tag{1.69}$$

We shall now see how $P(\mathbf{x}, t)$ obeys a partial differential equation known as *Fokker-Plank equation*.

1.3.1 Diffusion equation

The stochastic process we saw in section 1.2.2 is the distribution of the particle displacements in a very viscous fluid with initial condition at $x = 0$. The probability distribution that reproduces its variance is the Gaussian:

$$P(\mathbf{x}, t) = \left(\frac{1}{4\pi D t}\right)^{3/2} e^{-\mathbf{x}^2/4Dt} \tag{1.70}$$

Which respects the normalisation requirement that

$$\int P(\mathbf{x}, t)\, d^3x = 1, \text{for every } t \tag{1.71}$$

We have that the probability distribution (1.70) obeys the *diffusion equation*

$$\frac{\partial P}{\partial t} = D\nabla^2 P \tag{1.72}$$

which is the simplest example of a Fokker-Planck equation. Obviously, for general Langevin equations, we will have to derive the probability distribution with more tools than just intuition.

1.3.2 General stochastic processes

Still working with Langevin equation in the viscous limit ($m = 0$), we have a first order equation

$$\gamma\dot{\mathbf{x}} = -\nabla V + \mathbf{f} \tag{1.73}$$

If V is quadratic it corresponds to a harmonic oscillator potential, we have a similar problem to the one in section 1.2.3, if V takes any other form it gives us a non-linear stochastic equation with no general solution available. Still, we are able to reshape it into a Fokker-Plank equation.

Firstly, we consider a particle sitting at some point $\mathbf{x}$ at time t. After a short time δt, we will have a small difference in position

$$\delta\mathbf{x} = \dot{\mathbf{x}}\delta t = -\frac{1}{\gamma}\nabla V\delta t + \frac{1}{\gamma}\int_t^{t+\delta t}\mathbf{f}(t')\,dt' \tag{1.74}$$

Where $\mathbf{f}$ in this case represents the average value of the noise function over the small time interval δt. If we assume that there are no strong pitfalls in ∇V, we have that $\delta\mathbf{x}$ is small enough so that ∇V can be evaluated at original position $\mathbf{x}$. Now, for computing the average, since we know that $\langle\mathbf{f}(t)\rangle = 0$, we have that

$$\langle\delta\mathbf{x}\rangle = -\frac{1}{\gamma}\nabla V\delta t \tag{1.75}$$

For the computation of $\langle\delta x_i\delta x_j\rangle$ we have,

$$\gamma^2\langle\delta x_i\delta x_j\rangle = \langle\partial_i V\partial_j V\rangle\delta t^2 - \delta t\int_t^{t+\delta t}\langle\partial_i V f_j(t') + \partial_j V f_i(t')\rangle\,dt' + \int_t^{t+\delta t}dt'\int_t^{t+\delta t}dt''\langle f_i(t')f_j(t'')\rangle \tag{1.76}$$

where the first two terms are of order δt^2 and the third term, thanks to equation (1.52), can have the integrals removed and made proportional to δt, than, by ignoring the terms of order δt^2, we get to

$$\langle\delta x_i\delta x_j\rangle = 2\delta_{ij}D\delta t + \mathcal{O}(\delta t^2) \tag{1.77}$$

We want now to write a probability distribution that can reproduce equations (1.75) and (1.77). Considering conditional probability $P(\mathbf{x}, t+\delta t; \mathbf{x}', t)$ that the particle will sit at $\mathbf{x}$ at time $t+\delta t$ after being at $\mathbf{x}'$ at time t. From equation (1.70) we can write

$$P(\mathbf{x}, t+\delta t; \mathbf{x}', t) = \langle\delta(\mathbf{x} - \mathbf{x}' - \delta\mathbf{x})\rangle \tag{1.78}$$

where δx is the random variable, the distance moved in time δt. Next, we Taylor expand the right term. Expanding a delta-function is in itself a big abuse of notation, however, in this context, can be tolerated as an abuse of notation.

$$P(\mathbf{x}, t+\delta t; \mathbf{x}', t) = \left(1 + \langle\delta x_i\rangle\frac{\partial}{\partial x_i'} + \frac{1}{2}\langle\delta x_i\delta x_j\rangle\frac{\partial^2}{\partial x_i'\partial x_j'} + \dots\right)\delta(\mathbf{x} - \mathbf{x}') \tag{1.79}$$

Now we have to consider one more thing to complete the calculation. Our interest is the evolution of the probability $P(\mathbf{x}, t; \mathbf{x}_0, t_0)$ given an arbitrary

starting position $\mathbf{x}(t = t_0) = \mathbf{x}_0$, but still, we also need P to respect the Chapman-Kolmogorov equation

$$P(\mathbf{x}, t; \mathbf{x}_0, t_0) = \int_{-\infty}^{+\infty} P(\mathbf{x}, t; \mathbf{x}', t') P(\mathbf{x}', t'; \mathbf{x}_0, t_0)\, d^3\mathbf{x}' \tag{1.80}$$

for every intermediate time $t_0 < t' < t$.

We can now combine equations (1.79) and (1.80) and reach the final result. By substituting t with $t + \delta t$, thanks to the delta function in (1.79), we obtain

$$P(\mathbf{x}, t + \delta t; \mathbf{x}_0, t_0) = P(\mathbf{x}, t; \mathbf{x}_0, t_0) - \frac{\partial}{\partial x_i} \left(\langle \delta x_i \rangle P(\mathbf{x}, t; x_0, t_0) \right) + \frac{1}{2} \langle \delta x_i \delta x_j \rangle \frac{\partial^2}{\partial x_i \partial x_j} P(\mathbf{x}, t; \mathbf{x}_0, t_0) + \dots \tag{1.81}$$

Then, using equation (1.75) and (1.77), we obtain

$$P(\mathbf{x}, t + \delta t; \mathbf{x}_0, t_0) = P(\mathbf{x}, t; \mathbf{x}_0, t_0) + \frac{1}{\gamma} \frac{1}{\partial x_i} \left(\frac{\partial V}{\partial x_i} P(\mathbf{x}, t; \mathbf{x}_0, t_0) \right) \delta t + D \frac{\partial^2}{\partial x^2} P(\mathbf{x}, t; \mathbf{x}_0, t_0) \delta t + \dots \tag{1.82}$$

Now, if we Taylor expand the left side of the equation

$$P(\mathbf{x}, t + \delta t; \mathbf{x}_0, t_0) = P(\mathbf{x}, t; \mathbf{x}_0, t_0) + \frac{\partial}{\partial t} P(\mathbf{x}, t; \mathbf{x}_0, t_0) \delta t + \dots \tag{1.83}$$

And, equating these two last equations, we obtain the final result

$$\frac{\partial P}{\partial t} = \frac{1}{\gamma} \nabla \cdot (P \nabla V) + D \nabla^2 P \tag{1.84}$$

Which is the *Fokker-Planck* equation.

1.3.3 Properties of Fokker-Plank equation

We can write (1.84) as a continuity equation

$$\frac{\partial P}{\partial t} = \nabla \cdot \mathbf{J} \tag{1.85}$$

where $\mathbf{J}$ is the *probability current*

$$\mathbf{J} = \frac{1}{\gamma} P \nabla V + D \nabla P \tag{1.86}$$

The first term is referred to as *drift*, the overall motion of the particle due to known background forces, the second term instead is due to diffusion.

With the equation in this form we see that probability is conserved over time, meaning that if $\int P\, d^3x = 1$ at some time t_0, then it will remain so for all later times t. We can say this because

$$\frac{\partial}{\partial t}\int P\, d^3x = \int \frac{\partial P}{\partial t}\, d^3x = \int \nabla \cdot \mathbf{J}\, d^3x = 0 \tag{1.87}$$

Fokker-Planck equation describes the evolution of a system. It can be found in a naive form like the diffusion equation, in which there is no end point in the evolution, but for more generic potentials V we can find time-independent solution to the equation so that $\nabla \cdot \mathbf{J} = 0$. We call that equilibrium configurations, whose solution is found in the form

$$P(\mathbf{x}) \sim e^{-V(\mathbf{x})/\gamma D} \tag{1.88}$$

Using then equation (1.68), we find that it resembles a Boltzmann distribution for a particle with energy $V(\mathbf{x})$ in thermal equilibrium

$$P(\mathbf{x}) \sim e^{-V(\mathbf{x})/k_B T} \tag{1.89}$$

1.4 Depletion forces

When working with real suspensions of colloidal particles in a solvent, we might have included many objects of different length scales. For example, at the mesoscopic level, colloidal particles of the order of 100nm or more and, at the microscopic level, solvent molecules or dissolved ions of the order of a fraction of a nanometer.

It has been seen that, in a suspension of large and small particles, the pressure of the latter produces an attractive force between the former, which is referred as depletion force. In this chapter we will see the basic theoretical facts about this force and we will discuss how this force can interfere with local isotropy in brownian motions.

1.4.1 A simple example - two parallel plates

Let's consider a liquid of small hard spheres of radius R_S and two parallel hard plates within it. When the distance h between the two plates is $h < 2R_S$, the spheres are expelled from the gap between the plates. We call this effect a depletion, which leads inevitably to anisotropy of the local pressure around the plates, which implies an attractive depletion force between the plates.

This force was firstly predicted by Asakura and Oosawa, who predicted attractive force for $h < 2R_S$ and zero force for $h \geq 2R_S$. The force per unit area between two plates is connected to the osmotic pressure caused by the liquid of hard spheres, we can say then

$$\frac{\mathcal{F}}{A} = -\rho k_B T \Theta(2R_s - h) \tag{1.90}$$

where Θ is the Heaviside function.

We can say that the depletion force is purely an entropic force, since its properties are determined by the statistical tendency of the whole system to increase its entropy and are not determined by any underlying microscopic interaction. In fact, since the overlap of the restricted volume of the plates increases the volume accessible to the small hard-spheres, we have that their free energy $F_H = -TS$ decreases, increasing their entropy S.

This is an extreme ideal case, but we can observe a lot of real world examples in biology and chemistry, for example, depletion forces are responsible for phase separation in many cases of colloid particles immersed in a liquid of non-adsorbing polymers.

1.4.2 Asakura-Oosawa model

We will now consider Sho Asakura and Fumio Oosawa studies for the configuration of two hard spherical bodies of diameter $D = 2R_B$ immersed in a solution of hard spheres of diameter $d = 2R_S$. The treatment will be three dimensional from now on.

Around the large spheres there is a region unreachable for the small spheres because of the hard-sphere potential. The region volume of the two spheres is given by $V_E = \pi(D + d)^3/6$ but, when the large spheres start to get close enough (when $h < (D + d)/2$), the singular regions start to intersect and the unreachable region is reduced by the overlapping volume. The reduced excluded volume can be written as:

$$V'_E = V_E - \frac{2\pi l^2}{3}\left(\frac{3(D+d)}{2} - l\right) \tag{1.91}$$

where $l = (D+d)/2 - h/2$ is the width of the lens formed by the intersecting spherical caps.

Since the small spheres cannot penetrate into the excluded volume between the large spheres, we observe a phenomenon similar to the two large plates which results in an attractive force depending on the osmotic pressure of the solution.

We now try to analyse the system with some methods of statistical mechanics. Our main tools will be the partition function for a canonical ensemble $Q = (N!h_p^{3N})^{-1} \int e^{-\beta E}\, d\Gamma$, where $E = K + \mathcal{V}$ is sum of kinetic and potential terms, and statistical definition of Helmholtz's free energy $F_H = k_b T \ln Q$. We will have that that the force between two particles suspended in a solution is

$$\mathcal{F} = -\left(\frac{\partial F_H}{\partial h}\right)_T \tag{1.92}$$

where h is, again, the distance between the centres of the two large spheres. We want now to reach a valid expression for this force.

Potential energy can be expressed as

$$\mathcal{V} = \mathcal{V}_i + \mathcal{V}_e \tag{1.93}$$

where $\mathcal{V}_i$ is interparticle potential energy and $\mathcal{V}_e$ potential energy of interaction of small particles with an external field. If we work in a very dilute solution, we can approximate $\mathcal{V}_i = 0$. If the small particles are all identical, we can rewrite the partition function Q as

$$Q = \frac{1}{N!h_p^{3N}} \int \exp -\beta(T + \mathcal{V}_e)\, d\Gamma = \frac{1}{N!\Lambda^{3N}} \left(\int_V \exp -\beta \mathcal{V}_e(\mathbf{r}, h)\, d^3\mathbf{r}\right)^N \tag{1.94}$$

where N is the number of small spheres, V the total volume of the solution, $\Lambda = h/\sqrt{2\pi m k_B T}$ the de Broglie thermal wavelenght. In this context, $\mathcal{V}_e$ becomes $\mathcal{V}_e(\mathbf{r}, h)$ and represents the potential energy of a small sphere interacting with the two large spheres separated by h.

If we don't consider the interaction between the small spheres and the large spheres, we have that Q is completely related to the fraction of phase space available for the small spheres

$$Q = \frac{V_A^N}{N!\Lambda^{3N}} \tag{1.95}$$

where $V_A = V - V_E$ and represents the volume accessible to the centers of the small spheres. Free energy then becomes

$$F_H = -k_B T \ln\left(\frac{V_A^N}{N!\Lambda^{3N}}\right) = F_{id} - N k_B T \ln\left(\frac{V_A(h)}{V}\right) \tag{1.96}$$

where we used Stirling's approximation since $N \gg 1$ and where we defined $F_{id} = N k_B T(1 - \ln(N\Lambda^3/V))$ as the ideal contribution to free energy independent of h.

Now that we have an expression for the free energy, we can have two different scenarios: h can be big enough so that small spheres can penetrate between the two large spheres or it can be small enough so that this does not happen.

We have calculated the excluded volume for both cases in equation (1.91), which gives us

$$V_A = \begin{cases} V - V_E, & h \geq d + D \\ v - v_E + \pi/6(D + d - h)^2(D + d + h/2), & h < d + D \end{cases} \tag{1.97}$$

We can then linearize the logarithm in equation (1.96) because $V \gg V_E$. For $h < d + D$ we have

$$\ln\left(\frac{V_A}{V}\right) \approx -\frac{V_E}{V} + \frac{\pi}{6V}(D + d - h)^2(D + d + h/2) \tag{1.98}$$

Finally, we obtain

$$\mathcal{F} = \begin{cases} 0, & h \geq d + D \\ -\frac{N}{4v} k_B T \pi (D + d - h)(D + d + h) = -p_0 S, & h < d + D \end{cases} \tag{1.99}$$

Where p_0 is the osmotic pressure and S is the circular area of the overlapping volume of radius $r = \sqrt{(D + d)^2 - h^2}/2$The negative sign denotes the attractive force between the two large spheres, the depletion force is always attractive. Since we are working with low densities where we neglect interactions between small spheres, we have for the osmotic pressure $p_0 = \rho k_B T$ from van't Hoff's limit.

1.4.3 Improving the model

The calculations presented before gave us a good idea of the depletion force, but also makes very strong assumption about the homogeneity of local density of the small particles. We have assumed that density changes if and only if $h < D + d$, but in reality, near external potentials, such as a confining wall, homogeneities appear. If we want to improve the model, we need finer theories for considering them.

We have to include in the Hamiltonian term the potential energy of interaction between small particles and also consider the effects of excluded volumes of the small particles itself. Since we did not take them into account before, we got the equivalent of a first-order approximation for local density and, therefore, the pressure of an ideal gas.

B. Götzelmann et al. in [4] describes the same situation by using an external potential $V(\mathbf{R})$ composed of two contributions

$$V(\mathbf{R};h) = V_1(\mathbf{R}) + V_2[\mathbf{R} - (2R_b + h)\mathbf{e_z}], \quad h > 0 \tag{1.100}$$

where the second component is an hard-sphere potential

$$V_2(\mathbf{R}) = \begin{cases} \infty, & R < R_b + R_s \\ 0, & R > R_b + R_s \end{cases} \tag{1.101}$$

and the first component $V_1(\mathbf{R})$ is a potential that can represent any other fixed obstacle. In our case, another hard sphere such that $V_1(\mathbf{R}) = V_1(\mathbf{R})$, the problem is represented in Figure 1.1.

In this analysis we are concerned with force $\mathbf{F}(h)$ exerted by the fluid on the big sphere 2. For a generic obstacle V_1, we want to express the force just like we did before as

$$F_z(h) = -\left(\frac{\partial \Omega}{\partial h}\right)_{T,\mu} \tag{1.102}$$

And, by using density functional methods, B. Götzelmann et al.[4] derive the force in terms of the equilibrium number density profile $\rho(R)$:

$$\beta F_z(h) = \int d^3R\delta(|\mathbf{R}| - (R_b + R_s))(-z/R) \times \rho(\mathbf{R} + (2R_b + h)\mathbf{e}_z) \tag{1.103}$$

Where ρ is defined by

$$\rho(\mathbf{r}) = \left\langle \sum_{i=1}^{N} \delta(\mathbf{r} - \mathbf{r}_i) \right\rangle \tag{1.104}$$

The force can also be expressed as

$$\beta\mathbf{F}(h) = -\int_S dA\rho(\mathbf{R})\hat{\mathbf{n}} \tag{1.105}$$

where the integral is over the surface S of a sphere of radius $R_b + R_s$ centred at the center of sphere 2 and $\hat{\mathbf{n}}$ is the unit normal vector pointing outwards from the sphere.

This equation holds for any external potential $V_1(R)$, that means that obstacle 1 can have various shapes and can exert any arbitrary potential. Yet, if potential $V_1(R)$ is radially symmetric around z axis, as in our example case, we can find the same symmetry in the density profile and obtain

$$\beta F_z(h) = -2\pi(R_b + R_s)^2 \int_0^\pi d\omega \sin\omega\cos\omega\rho(\omega) \tag{1.106}$$

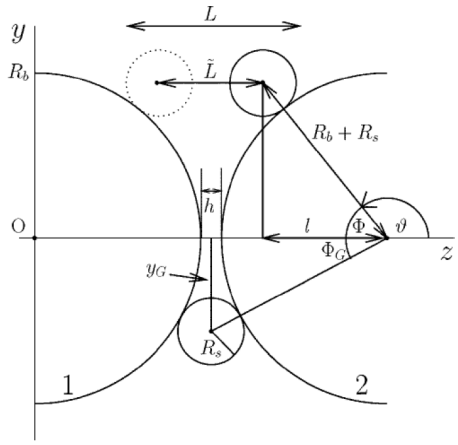


Figure 1.1: Our problem, two big spheres at distance h apart where we study the external potential acting upon a fluid of small spheres.

Where $\rho(\omega)$ is the contact density of the small hard-sphere fluid at the fixed big sphere 2 and ω the angle between the z axis and the axis connecting sphere 2 with the centre of the small sphere touching sphere 2. We also obviously have $F_x = F_y = 0$. We can alternatively write

$$\beta F_z(h) = 2\pi(R_b + R_s)^2 \int_{\pi/2}^{\pi} d\omega \sin\omega(-\cos\omega)\Delta\rho(\omega) \tag{1.107}$$

Where $\Delta\rho(\omega) = \rho(\omega) - \rho(\pi - \omega)$, with $\pi/2 < \omega < \pi$, is the contact density difference between the left ($\rho(\omega)$) and right ($\rho(\pi - \omega)$) hemisphere of sphere 2.

Now, for a fixed value of the y coordinate, we denote the distance in z direction a small sphere can travel before touching one of the two surfaces of the obstacles with $\tilde{L}$ (refer to Figure 1.1 for the notation). In our case, we have $\tilde{L} = 2R_b + h - 2l$. We can now change variables from ω to $\tilde{L}$ and obtain the exact equation

$$\beta F_z(h) = \pi\epsilon \int_h^{\infty} dL[(R_b + R_s) - \frac{1}{2}\epsilon(L - h)]\Delta\rho(L) \tag{1.108}$$

where we have defined $L \equiv \tilde{L} + 2R_s$ and where we have $\epsilon = 1$ for our sphere-sphere case and $\epsilon = 2$ instead for a sphere-wall case.

1.4.4 Depletion forces and Brownian isotropy

We have seen how excluded volumes and depletion forces cause an attractive force between certain objects. We now wonder, at an empirical level, if these

phenomenons can cause visible local violations of isotropy for the Brownian motion of a molecule with a particular geometry.

From the previous sections, we can be mathematically and statistically sure that the main macroscopic properties of the Brownian motion will be respected, whatever shape we take for the molecule.

Still, we can't say for example if the excluded regions of a "T" shaped molecule will cause collisions with certain characteristics to be more likely to happen than others, while for a normal sphere-shaped molecule such preferences are not present.

One thing that we can expect to see from a T-shaped molecule could be, for example, more collisions directed towards certain directions, a phenomenon which does not imply macroscopic violations at all but, still, can become of some interest in certain context of molecular dynamics and out-of-equilibrium thermodynamics.

In order to observe such local violation, we have chosen the direct approach to build an exact molecular dynamic simulator that could give us enough precise data to analyse, as it is reported in the next two chapters.

Chapter 2

NOCS, an event-based simulator

NOCS (Not Only Colliding Spheres) is an exact event-based 2D gas dynamics simulator written in C++. It implements the basic laws of Newton's dynamics and allows the user to create a simulator engine with any amount of hard spheres, immovable spheres, or arbitrarily complex molecules, made with multiple hard spheres of any mass and dimension.

Since the framework is event-based, there is no cumulative numerical error like a normal time-based simulation. In fact, every single event or collision is computed analytically, without interfering with the other elements of the system, and the user is therefore free to choose whether or not bring the whole system at a particular moment in time.

This chapter will expose the main features and characteristics of the framework, its main algorithms for the collision computation and also give a bird's-eye view of its structure. Next, some benchmarks will follow.

The entire work and the complete documentation for the project is available on GitHub under the GNU General Public License v3.0 at [8].

2.1 Main features

2.1.1 Assembling the simulation

With NOCS, the user is provided with an unitary square shaped region in which he can place 2 kind of elements: molecules, composed of hard spheres (from now on called "atoms") of arbitrary mass and dimensions, and immovable spheres (from now on called "bumpers"). This region can be subdivided into finer squares with a resizable grid, in order to improve computational

performances.

We impose periodic boundary conditions so that the region is a torus, that means that, if a molecule surpass the left bound of the square, it will teleport to the right bound without interference. This gives the possibility to analyse box-free thermodinamical scenarios and avoid biases caused by the presence of any wall. Still, if a wall is required, the user can of course initialise a column of bumpers as one.

It is also possible to use a "tag system" that allows the user to tag a particular molecule or a set of molecules with one or more "names". This system offers an extremely useful and immediate way to target a specific group of molecules when gathering data or manipulating the system.

For manipulating the physical rules of the system, it is possible at any moment in the program to change the elasticity coefficient of any collision (set at 1 by default). It can be a manipulation that involves all the collisions or only the collisions that involve molecules with particular tags.

For creating situations of non-equilibrium thermodynamics, the user can at any time execute an energy "reset" of a molecule (or a group of molecules with the help of the tag system). An energy reset takes the molecule's velocity and angular velocity and re-scales them in order to get the target energy requested by the user. This allows the creation inside the system of plausible thermostats, like always-hot and always-cold molecules. Therefore it is possible to manipulate the energy of any molecule but it is **not** possible to manipulate the velocity's direction and the proportions between translational energy and rotational energy[1].

2.1.2 Computing the simulation

The main idea behind NOCS's computation is to minimize as much as possible the numerical error that derives from excessive integrations. Therefore, NOCS tries to compute only the collisions that happens between the elements of the system and always with analytical precision. This approach is called event-based and it implies that time is not the main pillar of the simulation, but events are. More details on this are available at paragraph 2.2.3.

When it comes to floating point calculations, NOCS always works with C++'s double numbers. This could of course be incremented to a long double or an even more precise number, but, for the purposes of this thesis, double precision numbers were a right compromise between speed and precision.

Due to the program's architecture, the user can freely choose at what

[1] we decided to do so in order to avoid bugs that could be caused by excessive freedom in the manipulations.

times he wants to take an instant photo to the system, knowing that he will get the best performances only if he does them not too close to each other (that would be equal to force the program to work as a time-based simulator).

2.1.3 Gathering the data

A simulation made with NOCS has two states: a **standby state**, in which the whole system is at a defined time and we have an instant photo of the situation, and a **working state**, in which the simulation is bringing the system forward in time, event after event.

When the simulation is at standby state, we have a still photo of the system where the user can just peek into and gather all the properties of every molecule. To do this, NOCS offers the possibility to build customised lambda function to execute over one or more molecules (which can be chosen with the tag system).

Instead, when the simulation is at working state, the user can gather data from the events that are computed among the integration. To do so in an easy and intuitive way, NOCS offers a "subscription" system for the events, which allows the user to specify what kind of events he wants to gather data from and, most importantly, to specify what molecule's tags has to be present beneath the involved molecules. By passing customised lambda function to the subscritpion system, the user is able to pinpoint the exact data he desires, without wasting computational time or lines of code.

2.1.4 Graphics

When the simulation is at standby state, it is possible to draw a representation of the system inside a graphical window, in order to visualise the position of every desired element. The graphical part of the program relies completely on the C++ library "Passe_par_tout" made by Graziano Servizi [10].

2.2 Brief analysis of the main algorithms

2.2.1 Detecting a collision between molecules

Since we want to work with complex construction of spheres in rotation, the detection algorithm is not naive at all. Given any two molecules a and b in roto-translation, we want to efficiently determine whether or not the two molecules will collide and, if the answer is yes, which atoms from the two molecules will collide and when.

It is necessary to execute processes of skimming at many incremental level of precision, in order to quickly reject obvious cases of molecule that will not collide (for example, two molecules that are moving away or two molecules that will never come enough close).

Important note: as it's said further in paragraph 2.2.3, each molecule has its own "subjective time", therefore, for easier analysis, it is necessary to bring them at the same subjective time or, in other words, equalise them at the same moment (algorithm 1).

Data: molecule a, molecule b
Result: molecule a, molecule b with same subjective time
1 **if** *molecule a time* ! = *molecule b time* **then**
2 | equalise(molecule a, molecule b);
3 | **return** molecule a, molecule b;
4 **else**
5 | **return** molecule a, molecule b;
6 **end**

Algorithm 1: Paring the subjective time of two molecules for simpler computations

Skimming

Firstly, we can simplify the problem by taking the two molecules and considering around each of them the smallest circumference with the center in the molecule's center of mass that contains them. Then, we consider only the rectilinear motion of the two molecules and we determine if the two circumferences are or will start intersecting and, if so, when they will stop intersecting.

Considering the distance of the two molecules' center of mass in function of time, it is possible to treat this problem as finding the zeros of a parabola $y = a \times x^2 + b \times x + c$ with coefficients:

$$a = (\mathbf{v}_a - \mathbf{v}_b)^2 \tag{2.1}$$
$$b = 2 \cdot (\mathbf{x}_a - \mathbf{x}_b) \cdot (\mathbf{v}_a - \mathbf{v}_b) \tag{2.2}$$
$$c = (\mathbf{x}_a - \mathbf{x}_b)^2 - (R_a + R_b)^2 \tag{2.3}$$

where R_a and R_b are respectively the radius of the circumference of the first and second molecule. In this way the zeros of the parabola, if presents, correspond to the point where the two circumferences are tangent.

With some help from the Newton-Raphson algorithm, we can easily find the zeros of the parabola and, therefore, the time interval $[t_{\text{begin}}, t_{\text{end}}]$ in which the two molecules could actually collide.

The pseudo-code is reported into algorithm 2, for the sake of comprehension of this problem reformulation, we remand to multiple figure 2.1.

```
   Data: molecule a, molecule b
   Result: if present, t_begin, t_end
 1 for each molecule do
 2 |  build the smallest circle that contains it;
 3 end
 4 if the two circles are already touching then
 5 |  begin time = molecule's time;
 6 else
 7 |  if the two circles are not approaching then
 8 |  |  return "no collision detected";
 9 |  else
10 |  |  compute circles' meeting time;
11 |  |  begin time = meeting time;
12 |  end
13 end
14 compute circles' departure time;
15 end time = departure time;
16 return begin time and end time
```

Algorithm 2: First skimming for collision detection

Cropping and searching

After that, it is necessary to analyse the motion of every single couple of atoms taken respectively from the two molecules, in order to determine which couple will collide first, if it will. In other words, we need to consider the complex distance over time of multiple couples of objects with both an orbital motion and a rectilinear motion.

We already know that such collision can happen only in a restricted time interval that we computed before. We can still consider the problem as finding the zeros of a time function but, instead of having just a well-known parabola, we have a parabola summed with two orbital motions, which gives us an extremely complex object to derive directly (figure 2.2(a) shows an example of such function). Therefore, a direct approach with Newton-Raphson

algorithm is impossible, since computing the first derivative of the distance function would be extremely complex and inefficient.

Instead, we can make use of the Golden Section Search (GSS) algorithm and the Secant Method (SM) algorithm. If we crop the distance function of two atoms fine enough to have only time intervals small enough to contain one and only one local maxima or minima of the function (extremes of the interval excluded), we can immediately apply a GSS to find them with high efficiency. Then, when we find inside the same interval a local maxima higher than zero and a local minima lower than zero, we can be sure that an SM execution in that interval will find a zero of the function.

This reformulation stands on the assumption that two stable orbital motions will somehow maintain a form of periodicity in the distances, even when the rectilinear motion of the two molecules changes it a lot. With that said, we consider this function for the cropping fineness, even thought it still can be improved with deeper analysis of the complete analytic expression of the distances:

$$\Delta t = \frac{1}{2} \cdot min\left(\frac{\pi}{|\omega_a + \omega_b|}, \frac{\pi}{|\omega_a - \omega_b|}, \frac{\pi}{2 \cdot \omega_a}, \frac{\pi}{2 \cdot \omega_b}\right) \tag{2.4}$$

where ω_a and ω_b are the angular velocities of molecule a and b. For the sake of comprehension, we remand to multiple figure 2.2, where we plot an example distance function over time of two atoms from two approaching rotating cross shaped molecules (i.e. two molecules like the ones in figures 2.1).

At the end of this phase, we are capable to say if the two molecules will actually collide and, if so, which atoms exactly will and in which position and time. The pseudo-code is reported into algorithm 3.

After this general formulation of the algorithm, it's easy to extrapolate a simplified and optimised version for a collision between a molecule and a bumper.

2.2.2 Resolving a collision

In NOCS, collisions are defined classically as an immediate exchange of momentum. Every collision is by default completely elastic but, as we will see further, we also offer the possibility to place inelastic and superelastic collision with a customisable constant E.

Computing the effects of such collision, after we have found its time of happening, is a trivial operation. Since we know the collision time and which atoms are involved, we integrate the two molecules to that time and then we look for the position of the two atoms involved.

Data: molecule a, molecule b, time begin, time end
Result: if present, time collision and involved atoms

time of impact = NaN;
involved atom a = none;
involved atom b = none;
crop time [begin, end] in segments of lenght $\frac{1}{2} \cdot min\left(\frac{\pi}{|\omega_a+\omega_b|}, \frac{\pi}{|\omega_a-\omega_b|}, \frac{\pi}{2\cdot\omega_a}, \frac{\pi}{2\cdot\omega_b}\right)$;
foreach *segment in [begin, end]* **do**
  foreach *atom in molecule a* **do**
    foreach *atom in molecule b* **do**
      find minima with GSS (inside segment);
      if *minima < 0* **then**
        no collision, go to next loop cycle;
      else
        find maxima with GSS (from segment begin to minima);
        find zero with SM (from maxima to minima);
        time of impact = zero time;
        involved atom a = current atom from a;
        involved atom b = current atom from b;
        break and exit all loop cycles;
      end
    end
  end
end
if *time of impact = NaN* **then**
  return "there was no collision";
else
  return time of impact, involved atom a, involved atom b;
end

Algorithm 3: Cropping and searching for a collision of atoms.

Next, by working with the two atoms' centre and radius, we find the application point and the direction of the impulse **J**, acting on the first molecule during the collision. After that, we can compute the module of the impulse by imposing momentum conservation, energy conservation, and basic rotational dynamic rules. Because of Newton's third law, we automatically find everything about the impulse $\mathbf{J'} = -\mathbf{J}$ acting on the second molecule.

We define our impulse **J** with

$$m_1(\mathbf{v}_1' - \mathbf{v}_1) = \mathbf{J} \tag{2.5}$$

And, by knowing that

$$\begin{cases} m_1(\mathbf{v}_1' - \mathbf{v}_1) = m_2(\mathbf{v}_2 - \mathbf{v}_2') = \mathbf{J} \\ I_1(\omega_1' - \omega_1) = \mathbf{r_1} \times \mathbf{J} \\ I_2(\omega_2 - \omega_2') = \mathbf{r_2} \times \mathbf{J} \\ m_1\mathbf{v}_1^2 + m_2\mathbf{v}_2^2 + I_1\omega_1^2 + I_2\omega_2^2 = m_1{\mathbf{v}_1'}^2 + m_2{\mathbf{v}_2'}^2 + I_1{\omega_1'}^2 + I_2{\omega_2'}^2 \end{cases} \tag{2.6}$$

Where **r** is the position vector between the collision point and the centre of the molecule. We can simplify our calculations by writing

$$\begin{cases} (\mathbf{p}_1' - \mathbf{p}_1) = (\mathbf{p}_2 - \mathbf{p}_2') = \mathbf{J} \\ (L_1' - L_1) = \mathbf{r_1} \times \mathbf{J} \\ (L_2 - L_2') = \mathbf{r_2} \times \mathbf{J} \\ p_1^2/m_1 + p_2^2/m_2 + L_1^2/I_1 + L_2^2/I_2 = {p_1'}^2/m_1 + {p_2'}^2/m_2 + {L_1'}^2/I_1 + {L_2'}^2/I_2 \end{cases} \tag{2.7}$$

We can then reformulate energy conservation as

$$\begin{aligned} &\frac{p_1^2}{m_1} + \frac{p_2^2}{m_2} + \frac{L_1^2}{I_1} + \frac{L_2^2}{I_2} = \frac{J^2 + p_1^2 + 2\mathbf{J}\cdot\mathbf{p}_1}{m_1} + \frac{J^2 + p_2^2 - 2\mathbf{J}\cdot\mathbf{p}_2}{m_2} \\ &+ \frac{(\mathbf{r_1}\times\mathbf{J})^2 + L_1^2 + 2L_1\cdot(\mathbf{r_1}\times\mathbf{J})}{I_1} + \frac{(\mathbf{r_2}\times\mathbf{J})^2 + L_2^2 - 2L_2\cdot(\mathbf{r_2}\times\mathbf{J})}{I_2} \end{aligned} \tag{2.8}$$

By extracting the impulse module $|J|$ and simplifying, the calculation gives us a long but simple formula for computing the impulse module

$$|J| = (1+E) \times \frac{-\frac{\mathbf{p}_1\cdot\mathbf{n}}{m_1} + \frac{\mathbf{p}_2\cdot\mathbf{n}}{m_2} - \frac{L_1\cdot(\mathbf{r}_1\times\mathbf{n})}{I_1} + \frac{L_2\cdot(\mathbf{r}_2\times\mathbf{n})}{I_2}}{\frac{1}{m_1} + \frac{1}{m_2} + \frac{(\mathbf{r}_1\times\mathbf{n})\cdot(\mathbf{r}_1\times\mathbf{n})}{I_1} + \frac{(\mathbf{r}_2\times n)\cdot(\mathbf{r}_2\times\mathbf{n})}{I_2}} \tag{2.9}$$

Where **p** indicates the momentum of a molecule, **L** the angular momentum, **n** the versor of the impulse **J**, **r** the position vector between the collision point and the centre of the molecule and, finally, E is an elasticity modifier added by us and set to 1 by default (therefore $1 + E$ is 2 by default), which and can be changed by the user in order to gain inelastic or superelastic collisions (see paragraph 2.1.1).

Locked rotations

In certain points in our analysis, we want molecules unable to rotate ($\omega_1 = \omega_2 = 0$), therefore we ask the simulator not to consider the rotational dynamic in the collision resolutions. This leads to equations

$$\begin{cases} (\mathbf{p}_1' - \mathbf{p}_1) = (\mathbf{p}_2 - \mathbf{p}_2') = \mathbf{J} \\ p_1^2/m_1 + p_2^2/m_2 = {p_1'}^2/m_1 + {p_2'}^2/m_2 \end{cases} \tag{2.10}$$

Following to the reformulation

$$\frac{p_1^2}{m_1} + \frac{p_2^2}{m_2} = \frac{J^2 + p_1^2 + 2\mathbf{J}\cdot\mathbf{p}_1}{m_1} + \frac{J^2 + p_2^2 - 2\mathbf{J}\cdot\mathbf{p}_2}{m_2} \tag{2.11}$$

And the final result

$$|J| = (1+E) \times \frac{-\frac{\mathbf{p}_1\cdot\mathbf{n}}{m_1} + \frac{\mathbf{p}_2\cdot\mathbf{n}}{m_2}}{\frac{1}{m_1} + \frac{1}{m_2}} \tag{2.12}$$

Where we still consider the customisable elastic constant E equal to one by default.

2.2.3 Engine's event system

As mentioned earlier, NOCS is event-based, that means that the simulation's engine does not make a parallel time integration of every element of the system, but instead integrates separately only those elements whose events are first to come.

In other words, we have that each object in the system has its own "subjective time", that changes and increases when we resolve an event that involves that object, and a "version number" that keeps track of the number of changes occurred because of an event.

For example, let's say that we have three molecules a, b and c, at the beginning of the simulation we have an "objective time" for the whole system that is 0 seconds. That means that each molecule has a "subjective time" of 0 seconds. They also all have a version number equal to zero.
But now, we ask the engine to bring the entire simulation to the objective time of 10 seconds. The engine will then start to analyse every couple of the three molecule to see if they will collide or not.

Let's say that it finds out that a and b will collide at time 2.5s, b and c at time 3.0s, and memorises them in order of time taking also note of the molecule's version number. The engine is than sure that the collision between

a and b is the first to happen, so it resolves it, bringing the subjective time of a and b to 2.5s and the version number to 1.

Then, the engine recalculates all possible events that involve molecules a and b. This process is called "refreshing" and allows the engine to find how the molecule will interact with the system after the collision. Let's say that it finds out that a will collide with c at time 3.5s, memorise the event and takes down the version numbers.

Now, the engine sees that the next collision even would be the collision between b and c at time 3.0s, but when it checks the event number taken down, the engine sees that one of the two molecules has now a different version number, so it discards the event and proceeds with the next one.

This process of checking and refreshing continues when the engine reaches an event with a time of happening superior than the time it has to reach (in this case, 10 seconds). At that point the engine executes a simple integration for every single molecule. That brings the subjective time of each molecule to the same value and allows a correct data gathering.

The pseudo-code of this process is reported into algorithm 4.

2.2.4 A remark about the grid

Since analysing every possible couple of molecules for finding future events implies an $O(n^2)$ computational difficulty, where n is the number of molecules in the system, we need to do something extra to avoid useless computations.

We know that every molecule inside a gas has a mean free path that depends from molecule density and can be easily estimated. Now, since we know it is extremely unlikely that a molecule will freely run for much longer distances, we can say that it is useless to compute an event that involve two molecules distant something like 10 times the mean free path. Because, almost certainly, one of the two molecules will have a collision before with another nearer molecule.

Therefore, in order to save computational time, the user can set a subdivision grid of arbitrary fineness of the simulation space into smaller squares. Thanks to this grid, we can ask the engine not to compute every possible collision between any possible couple of molecules, but only the possible collision between molecules in the same grid's region or in adjacent regions.

The user can choose the grid's fineness at his own risk and danger, because if the fineness is too high, there could be two colliding molecules with their centre of mass placed into two non adjacent regions. That inevitably causes the event to be overlooked.

Default fineness is set to 1, that implies that there is no subdivision and every couple of molecules is analysed.

```
   Data: system at starting time, target time
   Result: system at target time
 1 foreach couple of elements do
 2 |   check for future event;
 3 |   if event happens then
 4 |   |   save element's version;
 5 |   |   save event;
 6 |   end
 7 end
 8 while there are events before target time do
 9 |   take closest event;
10 |   if elements in the event have different version then
11 |   |   discard event;
12 |   |   continue loop;
13 |   end
14 |   resolve event;
15 |   increase version of the elements in the event;
16 |   foreach element in event do
17 |   |   check for future event;
18 |   |   if event happens then
19 |   |   |   save element's version;
20 |   |   |   save event;
21 |   |   end
22 |   end
23 end
24 foreach element do
25 |   integrate motion to target time;
26 end
```

Algorithm 4: Loop cycle for engine's event-based integration

2.3 Benchmarks

The event-based architecture of the simulator makes the whole computational process single-threaded. To test the performances of the engine's event system and the performances of the collision detector algorithm, we run two increasingly difficult scenarios on a laptop computer provided with an Intel® Core™i5-7200U in standard conditions.

2.3.1 Test 1 - Many colliding spheres

Scenario

We consider N identical spherical molecules of radius 0.001 space units with unitary mass with 1 energy unit each and velocity direction assigned randomly. We ask the simulator to compute the system forward in time for 100 time units. We execute this scenario with two different grid setups: one with grid fineness set to 1, which means no grid subdivision at all, the other one with grid fineness set to the integer closest to $\sqrt{N}$, which means a number of region subdivision equal to the number of spherical molecules.

Results

Without grid optimisation. Results can be seen at figure 2.3(a), where the plotted performance are fitted with a standard quadratic function $f(x) = p_2 \times x^2 + p_1 \times x$. As expected, we can clearly see that increasing the number of molecules inside the simulator implies a quadratic increase in difficult (it is an $O(N^2)$ problem comparing every possible couple of molecules). This is exactly what we expected from what is indeed a naive-approach to an event-based simulation problem.

With grid optimisation. Results can be seen at figure 2.3(b), where the plotted performance are fitted with an exponential function $f(x) = \exp k_0 + k_1 x$. This might seem counterintuitve, but by comparing closely the performance reported in Table 2.1, we can understand that the exponential component $\mathcal{O}(e^N)$ is inevitably given by the fact that an high amount of spheres implies an higher density and, therefore, exponentially more collisions to compute. Still, if we limit our analysis to values of $N < 1000$, we can see how we reduced the specific problem of detecting collisions from a $\mathcal{O}(N^2)$ problem to a $\mathcal{O}(N)$ problem.

N	without grid	with grid
50	$2.015s$	$0.343s$
100	$8.921s$	$1.0s$
500	$428.391s$	$10.640s$
1000	$2395.67s$	$35.437s$
2000	$> 2h$	$313.641s$

Table 2.1: Different execution times of Test 1 problems of dimension N.

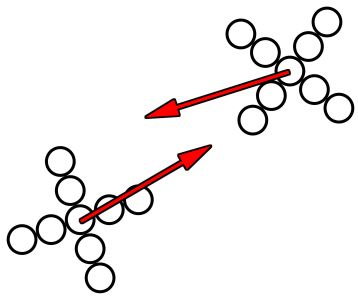
2.3.2 Test 2 - 2 increasingly complex colliding particles

Scenario

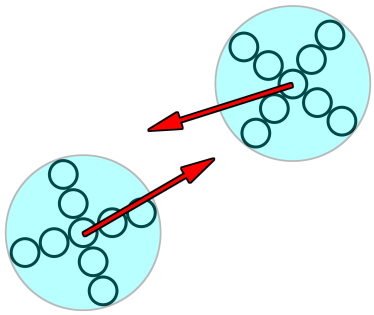
We consider 2 identical molecules of unitary mass composed by N atoms placed linearly of radius 0.001 space units. Each of the two molecules has 1 energy unit and assigned velocity direction so that they immediately start with a collision between themselves. We ask the simulator to compute the system forward in time for 100 time units without interruptions. In this case, grid fineness is set to 1, since this is not a case in which a finer grid would optimise our calculations.

Results

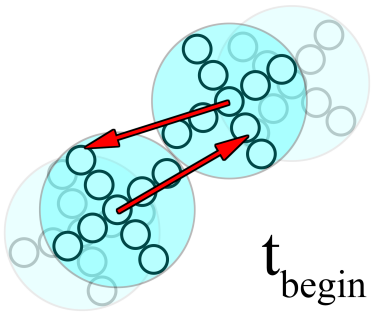
Results can be seen at figure 2.4, where the plotted performance are fitted with a standard quadratic function $f(x) = p_2 \times x^2 + p_1 \times x$. Even this time, we can prove that the detection algorithm works with an $O(N^2)$ efficiency when it comes to completely compute a collision. This is unfortunately the best we can obtain because, when it comes to fully detect the time and position of a collision, we are forced to inspect every possible couple of atoms from the two colliding molecules. The best we can do is to skim the passages and make minor optimisations.



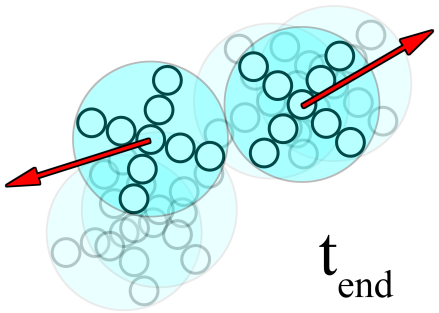
(a) The problem: two cross shaped molecules are approaching, we want to know whether they will collide or not.



(b) By building the smallest circles which contain the particles, we can reformulate the skimming problem as a parabola using equations (2.3).

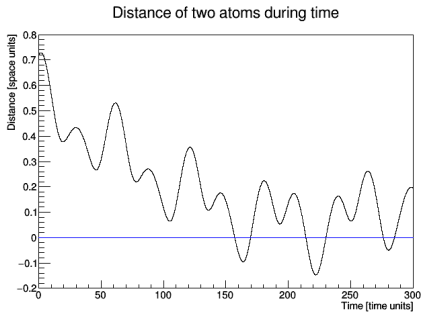


(c) If present, the first zero of the parabola is t_{begin} and corresponds to sphere encounter time. If not present, the molecules will not collide.

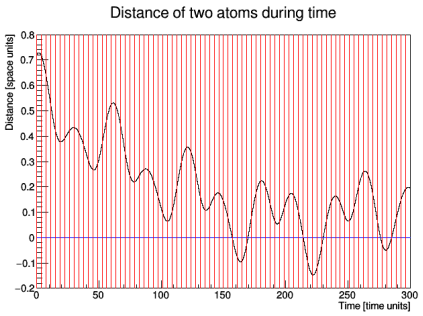


(d) The second zero of the parabola is t_{end}, time of spheres detachment. We can be sure that, if present, molecule collision will happen at time $t_{\text{begin}} \leq t < t_{\text{end}}$.

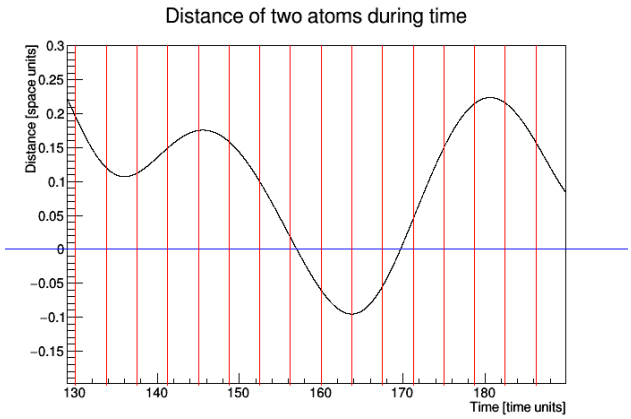
Figure 2.1: Graphical representation of the problem reformulation executed for skimming non colliding molecules.



(a) Plot of an example of distance over time of two atoms from two rotating approaching molecules, we want to crop it such that there is at most one extremum point in every interval.

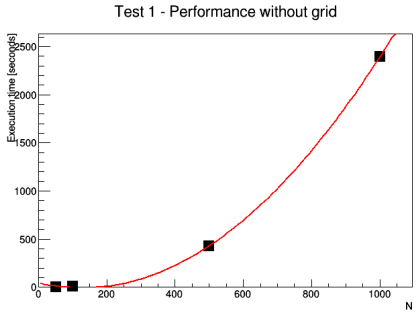


(b) Execution of a cropping with Δt computed with equation (2.4). The high fineness of the cropping assures us that in each interval there will be no more than one maximum.

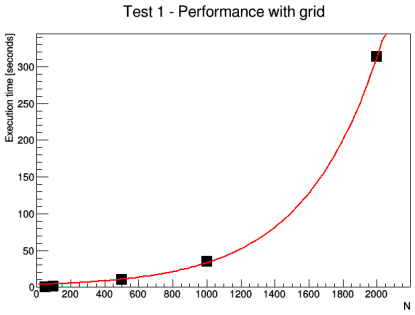


(c) Zoom over the first zero of the function, we can see how the fine cropping gives enough securities over the presence of maximum points. An application of GSS and SM over the eighth interval can find the zero with high efficiency.

Figure 2.2: Cropping application over the plot of a typical distance function over time of two atoms from two rotating approaching molecules.



(a) Test without grid. $p_1 = -0.80 \pm 0.09$ and $p_2 = (315 \pm 9) \times 10^{-5}$.



(b) Test with grid, asymptotic exponential difficult emerges. $k_0 = 1.27 \pm 0.16$ and $k_1 = (223 \pm 8) \times 10^{-6}$.

Figure 2.3: Measured performances for Test 1, dots are measures, red is fit with $f(x)$

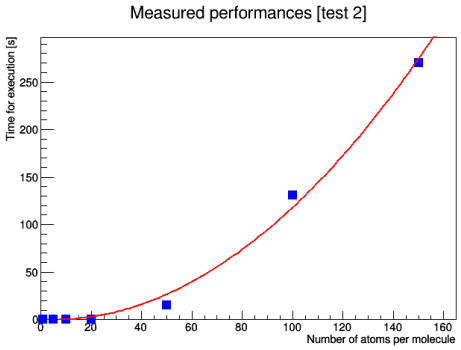


Figure 2.4: Measured performances for Test 2, dots are measures, red is fit with $f(x)$. We have $p_1 = -0.1 \pm 0.2$ and $p_2 = 0.013 \pm 0.001$).

Chapter 3

Analysing local isotropy in Brownian motion

In this chapter we shall analyse the obtained results from NOCS applications into Brownian gas simulation. The present work has to be seen as a first attempt to approach the problem of detecting and quantifying the local and eventual global isotropic violations in Brownian motion caused by excluded volume potentials. A more rigorous work would require much more different scenarios and more theoretical bases to evaluate.

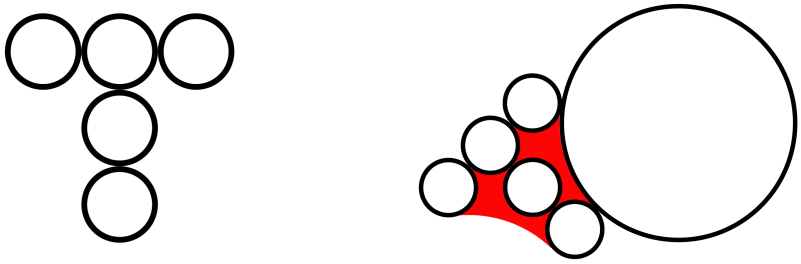
For analysing the effects of excluded volume potential, we wanted a molecule with the most simple geometry, capable to exclude parts of its atoms to hard spheres, bigger than the hard spheres which the molecule itself is composed.

We opted for a small T-shaped molecule, composed by 5 hard spheres (see figure 3.1). With this simple shape, we can already have some excluded zones that might be source of excluded volume potential. Moreover, its axial symmetry allows us to focus our analysis of the effects on one axis only, since every effect on the other axis will be compensated by the molecule's inner symmetries.

3.1 The experiments

3.1.1 Standard conditions

In standard condition we have one "T" shaped molecule composed by 5 atoms with 2 units of mass each. The molecule is submerged in a thermal gas composed by hard spheres of radius 0.01 space units of unitary mass with one unit of energy each and random starting direction. The 50% of the space



(a) Simple scheme of the molecule.

(b) Representation of the excluded volume of the molecule with an hard sphere 4 times bigger than a molecule's atom.

Figure 3.1: Pictures of the analysed T-shaped molecule.

is occupied by the hard spheres. The "T" shaped molecule is free to rotate.

Moreover, we also placed a small bumper of diameter 0.005 space units as wind breaker, in order to avoid motion biases which can be caused by an unfortunate combination of random initial velocities and toroidal space. Moreover, grid fineness is set to 50. (See figure 3.2(a) for a representation)

We run 2 different kind of simulations, in which each one has different relative dimensions between the hard-spheres and the molecule's atoms:

1. One where the molecule's atoms are half the size of the hard spheres (relative size 0.5).

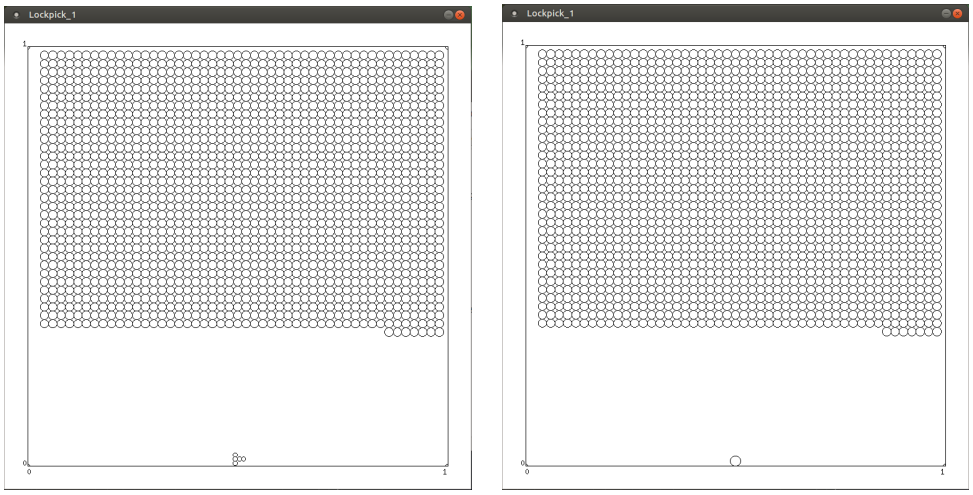
2. One where the molecule's atoms are a quarter of the size of the hard spheres (relative size 0.25).

In this way we hope to observe and quantify different intensities of isotropy violation caused by different portions of excluded volume.

The simulation is than integrated from 0 to 5000 units of time, with standbys every unit of time for data gathering, in order to gain every possible statistical information.

3.1.2 Control sample

For each computed simulation with "T" shaped molecule, we also compute a "control sample" simulation where we replace the molecule with an hard sphere of equal area. In this way, we are able to distinguish actual phenomenon caused by the molecule geometry from standard statistical biases. (See multiple figure 3.2(b) for a representation)



(a) Standard simulation at time 0. (b) Control group simulation at time 0.

Figure 3.2: Picture of two simulations made with NOCS

3.1.3 Locked rotation

For this case, we replicate in every aspect the standard conditions, except for one thing: we lock the "T" shaped molecule orientation to one fixed value. With some modifications in the collision resolution algorithm (see section 2.2.2), we just don't consider the torque caused by an impact. With this modification we hope to see strong global violations of Brownian isotropy. For each of the 3 standard cases, we run 2 different simulations:

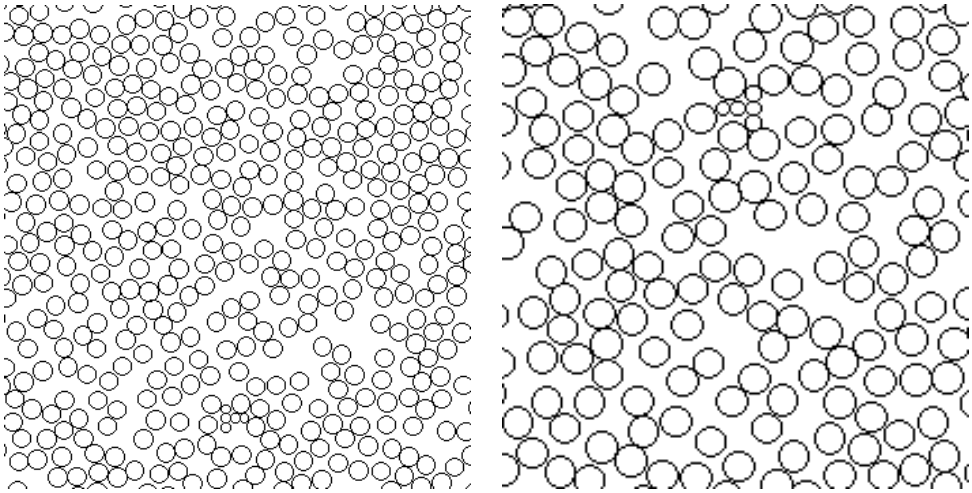
1. One in which the molecule has orientation locked at 0 radiant.

2. One in which the molecule has orientation locked at π radiant.

In this way, we are able to distinguish actual isotropic violations caused by the molecule geometry from other statistical biases. (See multiple figure 3.3 for a representation)

3.2 Data acquired

For each simulation, we consider directly a reference system relative to the T-shaped molecule (see figure 3.4) we will refer to it as (x', y'), for a sphere from a control group we consider directly parallel axis to the global space. We report:



(a) Simulation with molecule's rotation locked at 0 radians.

(b) Simulation with molecule's rotation locked at π radians.

Figure 3.3: Picture of two simulations made with NOCS with rotational dynamics deactivated

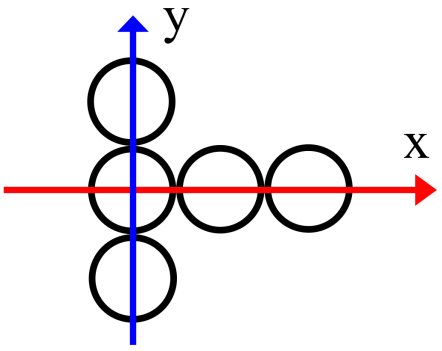


Figure 3.4: Reference system used over the T-shaped molecule

1. A cumulative histogram of all the single vectors $\Delta\mathbf{p}$ that the analysed molecule received from every collision happened during the simulation. Every $\Delta\mathbf{p}$ is sorted depending from the vector direction, considering the molecule's reference system. Then, we consider the simulation time that occurred (5000 time units) and we normalise the whole histogram so that we obtain a time average of the received momentum in function of the orientation. In this way, we hope to find preferential directions for the collision. We also report a polar graph transposition of the histogram for a better graphical visualisation for the reader.

2. In order to verify the characteristics of the brownian motion of the spheres from the control group, an histogram that takes every single $\Delta\mathbf{p}$ the molecule received separately, projects it over molecule's x' axis, and then classifies it depending on its module over the x' axis. Then, by normalising the histogram in order to gain a probability distribution, we expect to see a Gaussian curve.

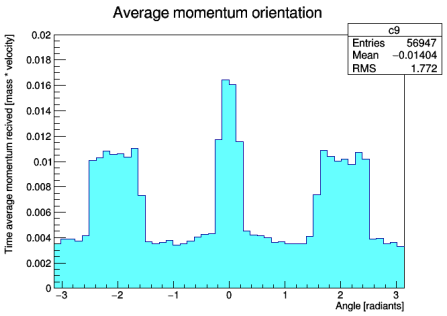
3. A graph which plots the progressive sum, time averaged, of the momentum p'_x that the analysed molecule has among its own x' axis. We expect to see white noise in the control groop and at least some form of minor correlation for the T-shaped molecules.

We considered only the time interval $[100, 5000]$, since the first time units of the system are characterised by a rapid diffusion of the hard spheres over the empty zones, among with a relocation of the starting velocities until a Boltzmann distribution is reached.

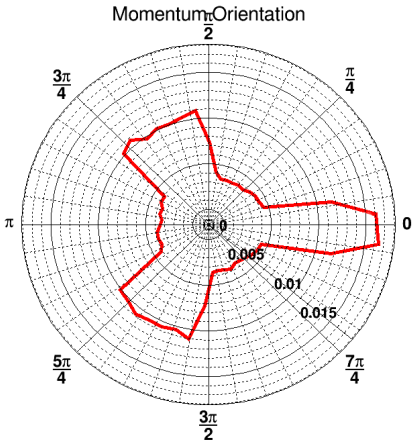
Standard conditions. Graphs (1) can be found at multiple figure 3.5, graphs (2) and (3) at multiple figure 3.6.

Control group. Graphs (1) can be found at multiple figure 3.7, graphs (2) and (3) at multiple figure 3.8.

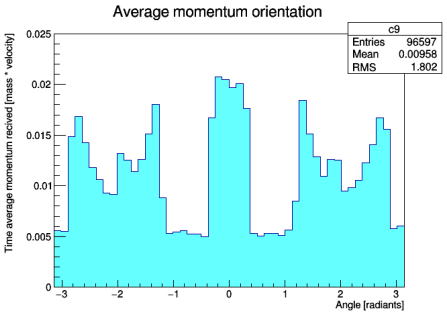
Locked rotation. Graphs (1) can be found at multiple figure 3.9 and multiple figure 3.10, graphs (2) and (3) at multiple figure 3.11.



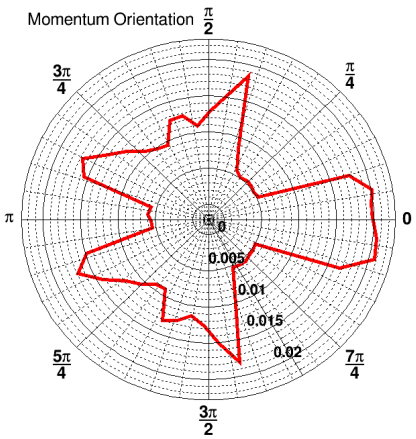
(a) Average momentum orientation for a molecule with atoms of relative size 0.25.



(b) Polar view of histogram (a).

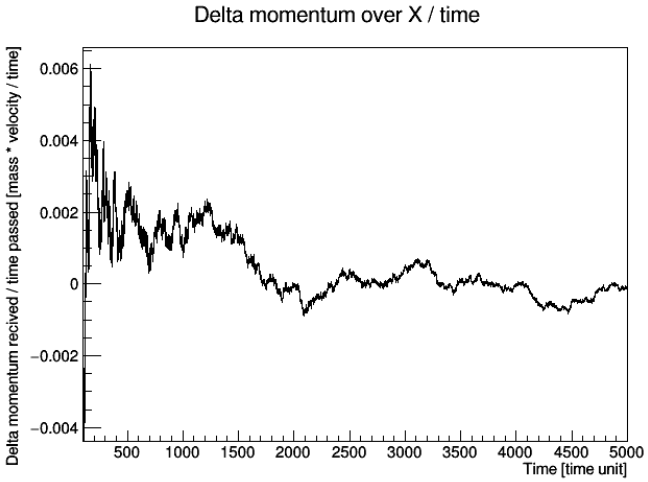


(c) Average momentum orientation for a molecule with atoms of relative size 0.50.

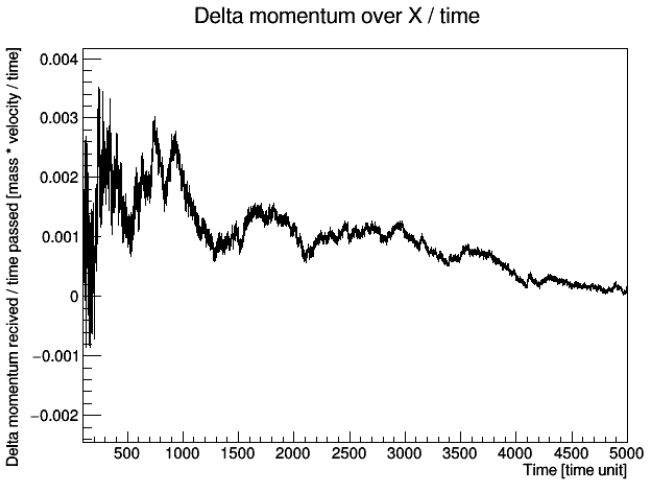


(d) Polar view of histogram (c).

Figure 3.5: Average momentum orientation for standard simulations.

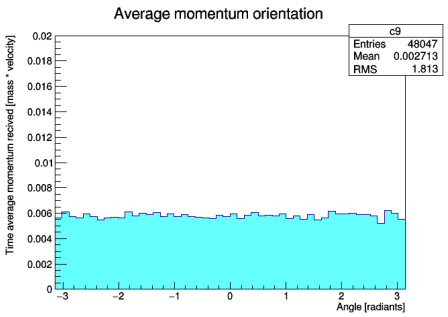


(a) Sum of $\Delta\mathbf{p}$ projected on molecule's x axis averaged over time. Molecule has atoms of relative size 0.25.

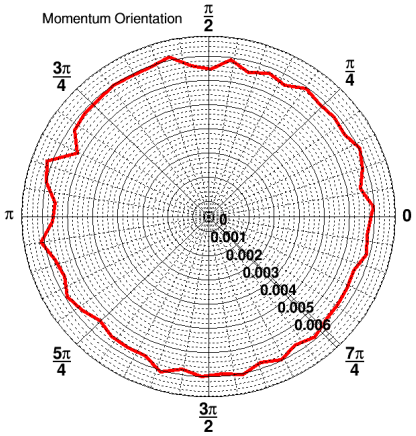


(b) Sum of $\Delta\mathbf{p}$ projected on molecule's x axis averaged over time. Molecule has atoms of relative size 0.50.

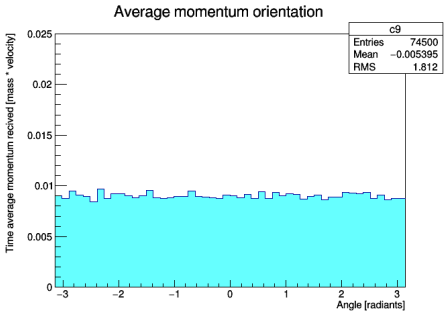
Figure 3.6: Analysis of $\Delta\mathbf{p}$ projections over x for standard simulations.



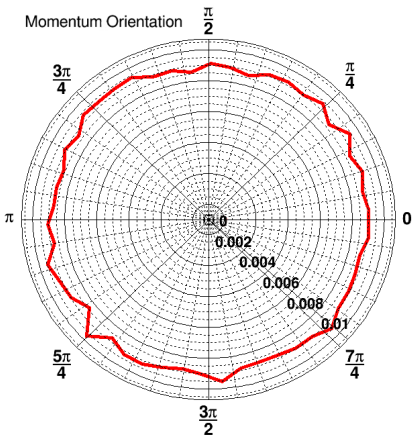
(a) Average momentum orientation for an hard sphere with the same area of a molecule made with atoms of relative size 0.25.



(b) Polar view of histogram (a).

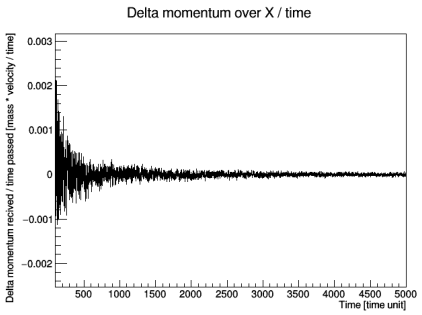


(c) Average momentum orientation for an hard sphere with the same area of a molecule made with atoms of relative size 0.50.

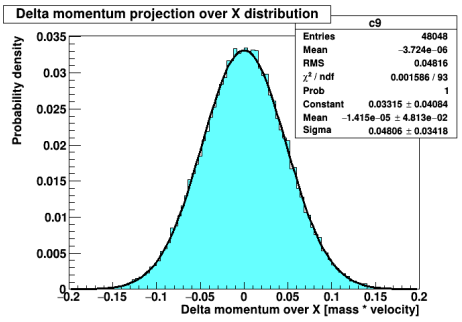


(d) Polar view of histogram (c).

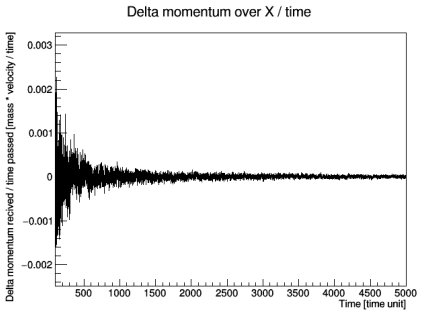
Figure 3.7: Average momentum orientation for control group simulations.



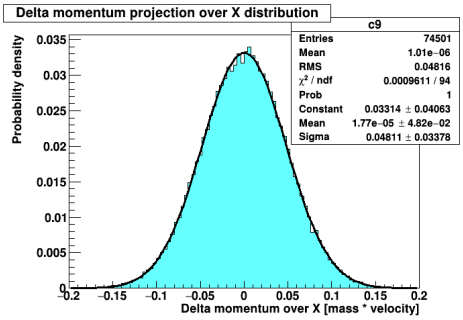
(a) Sum of $\Delta\mathbf{p}$ projected on sphere's x axis averaged over time. Sphere has same area of a molecule with atoms of relative size 0.25.



(b) Average distribution of $\Delta\mathbf{p}$ projected on sphere's x over time. Sphere has same area of a molecule with atoms of relative size 0.25.

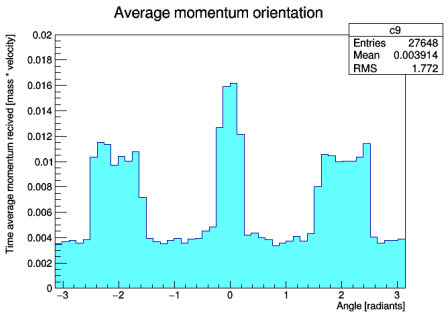


(c) Sum of $\Delta\mathbf{p}$ projected on sphere's x axis averaged over time. Sphere has same area of a molecule with atoms of relative size 0.50.

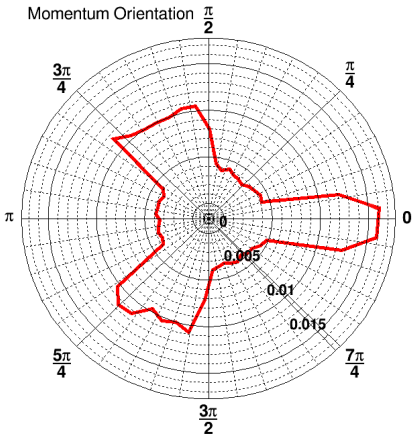


(d) Distribution of $\Delta\mathbf{p}$ projected on sphere's x over time. Sphere has same area of a molecule with atoms of relative size 0.50.

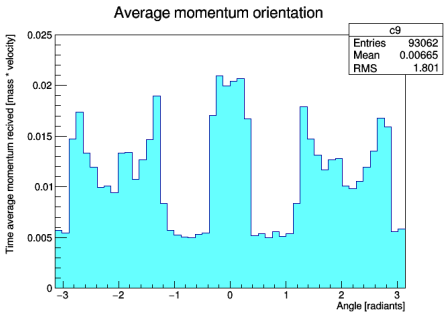
Figure 3.8: Analysis of $\Delta\mathbf{p}$ projections for control group simulations.



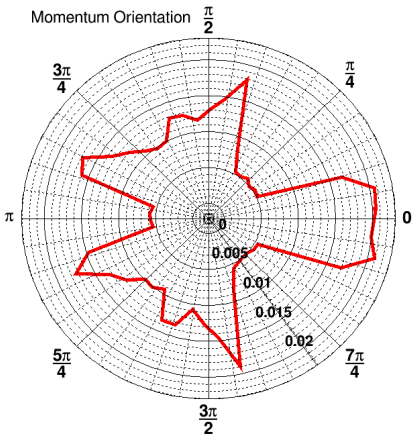
(a) Average momentum orientation for a molecule with atoms of relative size 0.25, with orientation locked on 0.



(b) Polar view of histogram (a).

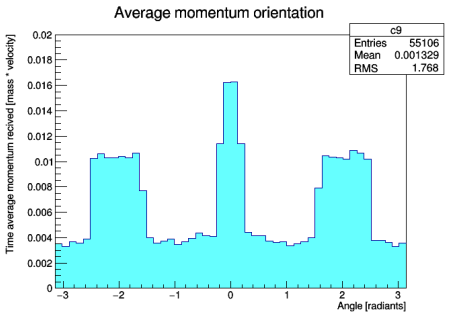


(c) Average momentum orientation for a molecule with atoms of relative size 0.50, with orientation locked on 0.

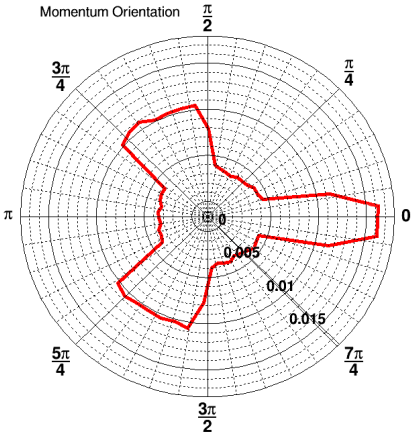


(d) Polar view of histogram (c).

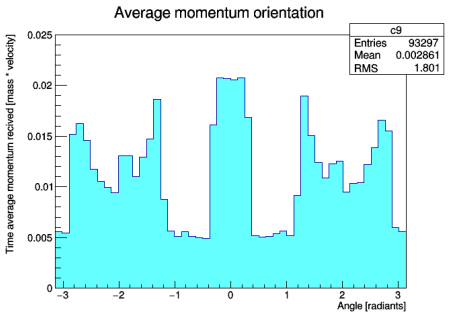
Figure 3.9: Average momentum orientation for simulations with rotation locked on 0.



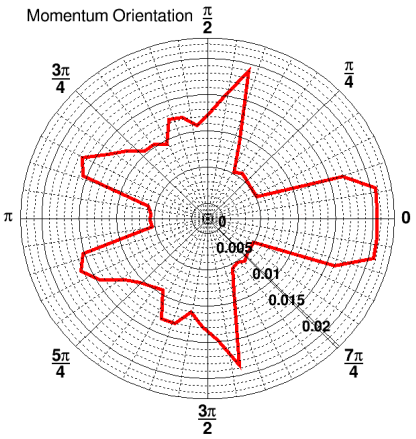
(a) Average momentum orientation for a molecule with atoms of relative size 0.25, with orientation locked on π.



(b) Polar view of histogram (a).

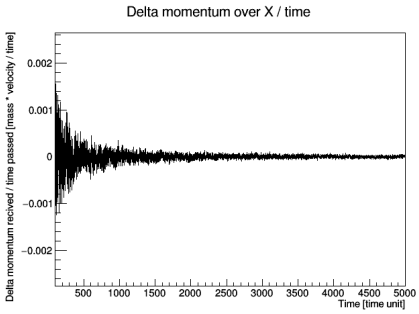


(c) Average momentum orientation for a molecule with atoms of relative size 0.50, with orientation locked on π.

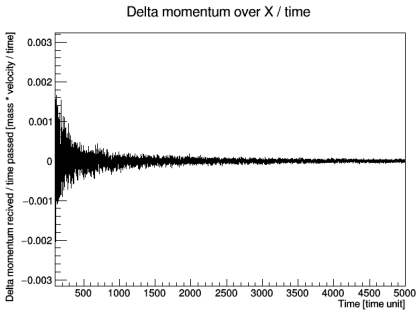


(d) Polar view of histogram (c).

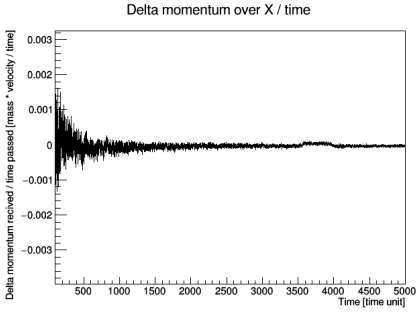
Figure 3.10: Average momentum orientation for simulations with rotation locked on π.



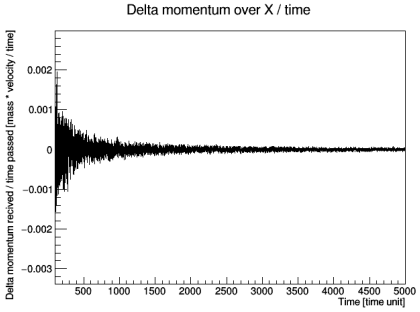
(a) Sum of $\Delta\mathbf{p}$ projected on molecule's x axis averaged over time. Molecule has atoms of relative size 0.25 and has rotation locked at 0.



(b) Sum of $\Delta\mathbf{p}$ projected on molecule's x axis averaged over time. Molecule has atoms of relative size 0.50 and has rotation locked at 0.



(c) Sum of $\Delta\mathbf{p}$ projected on molecule's x axis averaged over time. Molecule has atoms of relative size 0.25 and has rotation locked at π.



(d) Sum of $\Delta\mathbf{p}$ projected on molecule's x axis averaged over time. Molecule has atoms of relative size 0.50 and has rotation locked at π.

Figure 3.11: Analysis of $\Delta\mathbf{p}$ projections over x for simulation with rotation locked on 0.

Chapter 4

Analysis of the results and future research

4.1 Excluded volume and Brownian isotropy

As for the Brownian behaviour of the control group, everything is just as expected: as we can see in multiple figures 3.7 and 3.8, an hard sphere presents complete isotropy for the momentum orientation, and the delta momentum received over the x axis can be perfectly described as a white noise whose average fades over time.

Instead, for a T-shaped molecule free to rotate, we can see from multiple figure 3.5 that the geometry of the molecule causes huge preferential directions for the collision directions, that implies huge differences in the direction distribution of delta momentum. We can also see from multiple figure 3.6 that, instead of a white noise similar to the control group, we have an extremely small bias that suggests a preferential motion direction for the T-shaped molecule, along the T's rod. This is a local violation of symmetry caused by the molecule's geometry and excluded volume zones. Of course, this does not lead to any global violation of Brownian isotropy, because the molecule is free to rotate and, therefore, to change its motion direction in the global space.

By locking the molecule rotation, we hoped to see heavier violations of isotropy in the Brownian motion. Unfortunately, as we can see from multiple figure 3.11, there isn't any behaviour that differs from the one of a standard Brownian hard sphere (as seen from the control group). Still, we can observe from multiple figure 3.9 and 3.10 that the momentum orientation distribution still presents high preferential angles.

Probably, the effects of isotropy violations caused by excluded volume are

so small that they require much more precise and studied experiments and simulations in order to be detected. One other possibility is that what we want to observe takes place only in out-of-equilibrium contexts and not in isolated systems like the ones we simulated.

Still, this analysis has to be considered as an initial approach to the problem, made more as a trial stage for our simulator NOCS. Many more tests and different simulation approaches are necessary in order to declare some consistent results for this problem.

4.2 Future application for NOCS

We have seen how the versatility of NOCS allows various analysis with objects of many orders of magnitude. Simulators of this kind fit well when precise physical analysis of systems are needed and there is no strong mathematical theory to support the gathering.

The process, still, is single threaded, that means that at the moment we cannot rely on current developments in heavily paralleled computing and, as inevitable consequence, it is completely impossible to work with realistic amounts of molecules ($\sim 10^{23}$). Parallel computing, in the field of event-based simulators, is not contemplated at the moment but probably will be a possible direction of research and development for NOCS.

At its current state, NOCS allows a complete view of any dynamical system with $\sim 10^5$ objects in a reasonable amount of time, and it will be used at its current state to prosecute the research for excluded volume phenomenon. Next, when such phenomenon will be more classified in a context of equilibrium thermodynamics, we will try to observe the possible interference of excluded volume potentials in contexts of out-of-equilibrium thermodynamics, which can be easily recreated with NOCS tools for energy reset system (see section 2.1.1).

Moreover, possible developments of NOCS into event-based 3-dimensional simulators or event-based simulators capable of containing objects with freely hinged links might find even more interest for chemical physics and biophysics studies.

Appendix A

Important quoted algorithms

A.1 Newton-Raphson method

Newton-Raphson method is a method for finding successively better approximation to the roots of a real-valued function $f(x)$. The method in one variable starts by taking the function f, defined over real numbers x, the function's derivative f', and an initial guess x_0 for a root of the function f. If the function satisfies the assumptions made in the derivation of the formula and the initial guess is close, then we have that a better approximation of the root x_1 is

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_1)} \tag{A.1}$$

the process is then repeated as

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{A.2}$$

until a sufficiently accurate value is reached (in our case, five times the machine epsilon).

A.2 Golden section search

The golden section search is a method for finding the extremum of a strictly unimodal function $f(x)$ by recursively narrowing the range of values inside which the extremum is known to exist. This algorithm maintains the function values for triples of points whose distances form a golden ratio. The algorithm is iterative and it is executed until a sufficiently accurate value is reached (in our case, five times the machine epsilon). Pseudocode is reported in algorithm 5.

Data: function f, interval $[a, b]$, tol = machine epsilon $\times$ 5
Result: extremum point x

```
ψ = (1 + √5)/2;
c = b - (b - a) / ψ;
d = a - (a - b) / ψ;
while |c − d| > tol do
    if f(c) < f(d) then
        b = d;
    else
        a = c;
    end
    c = b - (b - a) / ψ;
    d = a - (a - b) / ψ;
end
```

Algorithm 5: Golden section search algorithm

A.3 Secant method

Secant method is a root-finding algorithm that works with succession of roots of secant lines to better approximate a root of a function f. Its performances are not as good as Newton-Raphson method, but it's a valid alternative when it is extremely difficult to obtain a valid computational expression of f'. Given two starting points x_1 and x_2, among which we know there is a root of the function that, we have the recurrence relation

$$x_n = x_{n-1} - f(x_{n-1})\frac{x_{n-1} - x_{n-2}}{f(x_{n-1} - f(x_{n-2}))} = \frac{x_{n-2}f(x_{n-1}) - x_{n-1}f(x_{n-2})}{f(x_{n-1}) - f(x_{n-2})} \tag{A.3}$$

Which has an order of convergence of $(1+\sqrt{5})/2 \approx 1.618$, if the initial values are "close enough" to the root. There is no actual definition of how much close is "close enough", but, for our usage, there are no cases of divergence. As above, tolerance is set to five times the machine epsilon.

Bibliography

[1] Sho Asakura and Fumio Oosawa. "Interaction between particles suspended in solutions of macromolecules". In: *Journal of Polymer Science* 33.126 (1958), pp. 183–192. ISSN: 1542-6238. DOI: 10.1002/pol.1958.1203312618. URL: http://dx.doi.org/10.1002/pol.1958.1203312618.

[2] Sho Asakura and Fumio Oosawa. "On Interaction between Two Bodies Immersed in a Solution of Macromolecules". In: *The Journal of Chemical Physics* 22.7 (1954), pp. 1255–1256. DOI: 10.1063/1.1740347. eprint: http://dx.doi.org/10.1063/1.1740347. URL: http://dx.doi.org/10.1063/1.1740347.

[3] Armando Bazzani. *Fisica dei Sistemi Complessi - More is Better*. Dipartimento di Fisica e Astronomia dell'UniversitÃ di Bologna, Jan. 2017. URL: http://www.physycom.unibo.it/pagina_web_bazzani.html/bazzani_web_dir/sistemi_complessi/definitivo.pdf.

[4] B. Götzelmann, R. Evans, and S. Dietrich. "Depletion forces in fluids". In: *Phys. Rev. E* 57 (6 June 1998), pp. 6785–6800. DOI: 10.1103/PhysRevE.57.6785. URL: https://link.aps.org/doi/10.1103/PhysRevE.57.6785.

[5] K. Higaki et al. "Physical analyses of gel-like behavior of binary mixtures of high- and low-melting fats". In: *Journal of the American Oil Chemists' Society* 80.3 (Mar. 2003), pp. 263–270. ISSN: 1558-9331. DOI: 10.1007/s11746-003-0687-z. URL: https://doi.org/10.1007/s11746-003-0687-z.

[6] Lev D. Landau and Evgeny M. Lifshitz. *Course of Theoretical Physics, volume 10, Physical Kinetics*. 1981.

[7] Davide Marenduzzo, Kieran Finan, and Peter R. Cook. "The depletion attraction: an underappreciated force driving cellular organization". In: *The Journal of Cell Biology* 175.5 (2006), pp. 681–686. ISSN: 0021-9525. DOI: 10.1083/jcb.200609066. eprint: http://jcb.rupress.org/

content/175/5/681.full.pdf. URL: http://jcb.rupress.org/content/175/5/681.

[8] Matteo Monti and Carlo Emilio Montanari. *NOCS (Not Only Colliding Spheres) exact 2D gas dynamics framework*. 2017. URL: https://github.com/matteomonti/nocs.

[9] Marco Mravlak. *Depletion force*. Faculty of Mathematics and Physics of University of Ljubljana, May 2008. URL: http://mc2tar.phys.uniroma1.it/~fs/didattica/LIQUIDI/LIQUIDI/depletion.pdf.

[10] Graziano Servizi. *passe_par_tout library*. 2017. URL: http://www.physycom.unibo.it/labinfo/passe_par_tout.php.

[11] David Tong. *Kinetic Theory*. University of Canbridge Graduate Course. 2012. URL: http://www.damtp.cam.ac.uk/user/tong/kintheory/kt.pdf.