INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
ROUEN NORMANDIE

INSA de Rouen

Internship report

Cardiff University

School of Mathematics

CARDIFF UNIVERSITY

Senghennydd Rd, Cardiff CF24 4AG

Author :
ANQUETIL Thomas, GM3

Supervisors :
Dr. BEN-ARTZI Jonathan
Dr. MORISSE Baptiste

18/06/2018 - 07/09/2018

« Physics does not even try to give us complete information about the events around us, it gives us correlations between those events. »

- Eugene Wigner, *1963 Nobel Prize in physics for his work on atomic nucleus.*

Résumé (français)

Ce rapport constitue le compte rendu du stage que j'ai effectué pendant l'été 2018 dans le cadre de la validation de ma troisième année à l'INSA de Rouen, au département Génie Mathématique. Du 18 Juin au 7 Septembre, j'ai intégré l'Université de Cardiff (Pays de Galle, Royaume-Uni), et plus précisément, l'École de Mathématiques de cette université. J'ai été accueilli par Jonathan Ben-Artzi, maître de conférence et chercheur en mathématiques à l'université de Cardiff, principalement en analyse et équations aux dérivées partielles. J'ai travaillé avec lui et l'un de ses collègues également chercheur, Baptiste Morisse.

Le sujet d'étude de ce stage porte sur la Théorie cinétique des gaz, et plus particulièrement, le comportement des plasmas. C'est un sujet très vaste et potentiellement complexe, et surtout, sur lequel la recherche est toujours très active. Son application en est toujours au stade théorique et experimental. Cela sera discuté plus précisément par la suite. Compte tenu de l'étendu du sujet et de la durée relativement courte du stage, le but n'est pas de travailler formellement sur quelque chose de nouveau, ni de contribuer à un projet en particulier; mais de travailler sur la Théorie cinétique des gaz dans son ensemble et d'en découvrir certains aspects. Le but de ce rapport est donc de synthétiser l'ensemble des travaux que j'ai été amené à étudier.

La suite de ce rapport est rédigée en anglais.

Acknowledgements

First and foremost I would like to grant all my gratitude to my supervisor Jonathan Ben-Artzi, who agreed to receive me for this internship. I thank him a lot for his welcome, the help he brought me for work, from providing general guidelines to helping to solve mathematics prolems. Apart from that, for all the summer he showed kindness, goodwill, and was always willing to provide any help.

Then, I would like to offer special thanks to Baptiste Morisse, researcher colleague of Jonathan, who in the same way, helped me a lot by always showing himself available, giving many tips and advices in the solution of several problems.

Finally, I offer a particuliar word to Martin Vivier, a friend of mine studying at Magistère de Physique fondamentale at Paris-Sud University, who provided help in the solution of some problems.

Contents

Overview of the report

This report is the result of the internship I performed during summer 2018, in the scope of my engineering school course. From the 18^{th} of June to the 7^{th} of September, I entered the School of Mathematics of Cardiff University in Wales (U.K.) to work by searchers' side. This internship comes as a validation of my third year of further studies at INSA Rouen Normandie. It has a double objective : to put knowledge I learnt during my year in Génie Mathématique into practice, and at the same time, to provide an enriching professional experience.

Cardiff is the capital of Wales, country both part of the United Kingdom and British Isles. Cardiff University was founded in 1883 as the *University College of South of Wales and Monmouthshire*, but received its independance and own-degree awarding in 1997, to adopt the public name of *Cardiff University* in 1999. It is the only Welsh university that is part of the Russel Group (an association gathering several british universities with a global view to leading and promoting research efforts in U.K.). It offers a wide set of courses, from psychology to mathematics and physics, by way of chemistry, litterature, art, history, laws and politics, business.

In 2016-2017, the university received 31,595 students for 5,230 staff employees. It had a budget of £503.8 million, of which 57% was dedicated to the staff, and 20% to the research (see [6]). Cardiff University's world rank was 99 in 2017, and in top 150 in 2016 (according to the ARWU).

I was welcomed by Dr. Jonathan Ben-Artzi, senior lecturer at Cardiff University , and member of the Mathematical Analysis group. His main topic of research are analysis and partial differential equations. I worked with him and one of his collegue, Dr. Baptiste Morrise, researcher in the same university. Since the topic was totally new to me, they both helped me to have a better understanding of it.

The topic of this internship was the study of some aspects of Kinetic theory of gases. Its aim is to explain the macroscopic behavior of gases. This domain of study arose during the 18^{th} century, in Bernouilli's *Hydrodynamica*, and was later formalized by Maxwell and Boltzmann. Since this field of research is still topical, and regarding its potential complexity according to how deeply one would want to study it, the aim of this internship was neither to work on something new, nor throwing myself into a totally new project; but to explore different aspects and problems of Kinetic theory, through both their physical and mathematical facet.

Abstract

The Kinetic theory of gases seeks to explain the behavior of gases in term of the averaged motion of their particles. In this study, we focus on a specific aspect of Kinetic theory : the behavior of particles in a *plasma*. Plasma is the fourth fundamental state of matter. It is basically a ionized gas. It can be generated by heating or by subjecting a gas to an intense electromagnetic field to the point that it becomes electrically conductive. Therefore, the behavior of particles is mostly ruled by the long-range electromagnetic field. Due to these specific conditions, plasma does not show up naturally on Earth. Yet, as it is the main component of stars, it remains the most abundant form of matter in the universe.

This theory is based on several assumptions :

- We consider the gas as a particle cloud, in which the average distance between particles is larger than their size.
- All particles in the gas have the same mass.
- The number of particles in the cloud is very large. This allows to treat particles statistically.
- Relativisitc and quantum effects are negligible. That means that particles are treated with classical mechanics laws.

Vlasov's equation is a differential equation which describes the time evolution of the distribution function of particles within a plasma. It considers only long-range interactions between particles, and collisions are neglected. It is named after the russian theoretical physicist Anatoly Alexandrovich Vlasov who first suggested it in 1938.

Poisson's equation is a partial differential equation, which arises for instance to describe the potential field caused by a given charge or mass density distribution.

The Vlasov-Poisson system describes the evolution of the distribution function for a gas subjected to an electric field, without magnetic field.

$$\begin{cases} \partial_t f + \frac{\mathbf{p}}{m} \cdot \nabla_{\mathbf{x}} f + \frac{q\mathbf{E}}{m} \cdot \nabla_{\mathbf{p}} f = 0 \\ \nabla_{\mathbf{x}} \cdot \mathbf{E} = -4\pi\rho \end{cases} \tag{1}$$

In the scope of this study, we can get rid of some parameter. Our previous assumptions say that m is constant. Plus, since we will work considering only electric field, then only electrons distribution will change. Therefore, q is constant as well. As constant, we can get rid of m and q by setting them to 1. Results will only differ by a multiplying factor. For the same reason, we can remove the 4π from Poisson's equation. Thus, the system is

$$\begin{cases} \partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \mathbf{E} \cdot \nabla_{\mathbf{v}} f = 0 \\ \nabla_{\mathbf{x}} \cdot \mathbf{E} = -\rho \end{cases} \tag{2}$$

Notations

For the whole report, following notations will be used :

- $f = f(t, \mathbf{x}, \mathbf{v})$ is the particles distribution function, $\rho(t, \mathbf{x}) = \int f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}$ is the number density.
- $t \geq 0$ is time, $d \in \mathbb{N}$ is the dimension (usually $d = 3$ but some changes can be specified)
- $\mathbf{x}$, $\mathbf{p}$,$\mathbf{v} \in \mathbb{R}^d$ are respectively position, momentum and velocity vectors. Sometimes $\mathbf{x} \in \mathbb{T}^d = (\mathbb{R}/\mathbb{Z})^d$. Every vectors are written in **bold type**.
- ∂_y is the partial derivative operator with respect to the real variable y.
- $\nabla_{\mathbf{x}}$ is the gradient vector operator, defined by $\nabla_{\mathbf{x}} = \begin{pmatrix} \partial_{x_1} \\ \partial_{x_2} \\ \vdots \\ \partial_{x_d} \end{pmatrix}$.
- $\mathbf{E} = \mathbf{E}(t, \mathbf{x}) : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^d$ is the eletric field vector.

Plan of the report

Chapter 1 is a more detailed introduction to kinetic theory. It shows the different scales one can study the problem at, and also how to set our problem.

Chapter 2 deals with the Vlasov's equation itself, showing how to establish it and presenting some results on its basis.

Chapter 3 is about the Landau damping. It's an electric field oscillation relaxation phenomenon, suggested in 1946 by the Russian physicist Lev Davidovich Landau.

Chapter 4 shows some simulations I worked on at the beginning of the internship, so I could have a better understanding of the topic.

Conclusion takes a step back from the whole topic, and the internship in general, concluding about this experience in its whole.

Appendix provides some elementary mathematical tools used in the report : elementary theory of L^p spaces, Fourier series and Laplace transform.

Internship progression follow-up provides a wide vision about the time evolution of the internship.

Bibligraphy indexes every references of the report.

Chapter 1

Kinetic theory

1.1 Motivation

Describing motions has been an major topic in physics for centuries. In the 17^{th} century, Newton was the first to formalize it, founding what we call nowadays *classical mechanics*. Though it has been a revolution and is still widely used in many applications, it might not be the best way to study all kinds of systems, especially when it comes to extreme scales. Actually, classical mechanics works within the scope of several assumptions and approximations, which may not be pertinent according to what we study. Morover, the direct application of classical mechanics (essentially Newton's laws) can lead to chaotic problems. This is what we are going to show with the N-body problem.

1.2 The N-body problem

1.2.1 Definition

The N-body problem goes back to the initial will of predicting motions of planets in the solar system. Knowing only initial positions and speeds of three planets, Newton was able to write an equation that describes their motions. However, it happened that as time went on, their behavior was different from what the equation said. Newton then understood what will be at the heart of the N-body problem : all orbits are affected by attractive forces of all planets at the same time, which means every trajectories depend on the current state of the whole system. Thus, it is very difficult to predict the system state, knowing only its initial configuration. We can extend this to an entire galaxy, for which we want to predict the behavior of every stars.

This problem arises as well when it comes to studying particles in a gas. Since it is considered as a particle cloud for which each particles are treated with classical mechanics laws, the same problem occurs. **The only difference is the force in question.**

1.2.2 Computations : a complex problem

To study a galaxy, Newton's law of gravity is at stake. For a gas, we use Coulomb's law. In both cases, we will show that the problem is the same :

First, consider N stars of mass m_i and positions $\mathbf{x}_i$ in $\mathbb{R}^3$, $i \in [\![1, \dots, N]\!]$. The exerted force of a star of mass m_i on another star of mass m_j is given by **Newton's law of gravity** :

$$\mathcal{F}_{ij} = \mathcal{G}\ \frac{m_i m_j (\mathbf{x}_j - \mathbf{x}_i)}{||\mathbf{x}_j - \mathbf{x}_i||^3} \tag{1.1}$$

Each m_i has a matching position vector $\mathbf{x}_i$. $||\cdot||$ is the Euclidean distance and $\mathcal{G}$ is the gravitational constant.

Second, consider N' particles of electrical charges q_i and position $\mathbf{x'}_i$ in $\mathbb{R}^3$, $i \in [\![1, \dots, N']\!]$. The force exerted by a particle of electrical charge q_i on another particle of electrical charge q_j is given by **Coulomb's law** :

$$\mathcal{F'}_{ij} = \mathcal{K}\ \frac{q_i q_j (\mathbf{x'}_j - \mathbf{x'}_i)}{||\mathbf{x'}_j - \mathbf{x'}_i||^3}. \tag{1.2}$$

$\mathcal{K}$ is the Coulomb's constant.

We see that in both cases, the forces are of the same nature. Only constant terms and sign change. Now, we use Newton's second law that says that :

$$\ddot{\mathbf{x}}_i = \begin{cases} \displaystyle\sum_{\substack{j=1 \\ j\neq i}}^{N} \mathcal{F}_{ij} = \mathcal{G} \sum_{\substack{j=1 \\ j\neq i}}^{N} \frac{m_j m_i (\mathbf{x}_j - \mathbf{x}_i)}{||\mathbf{x}_j - \mathbf{x}_i||^3} & \textbf{for a galaxy} \\ \displaystyle\sum_{\substack{j=1 \\ j\neq i}}^{N'} \mathcal{F'}_{ij} = \mathcal{K} \sum_{\substack{j=1 \\ j\neq i}}^{N'} \frac{q_i q_j (\mathbf{x'}_j - \mathbf{x'}_i)}{||\mathbf{x'}_j - \mathbf{x'}_i||^3} & \textbf{for a gas} \end{cases} \tag{1.3}$$

This shows that, whatever N stands for (gas or galaxy), the problem results into $3N$ similar equations (because $\mathbf{x}$ is in $\mathbb{R}^3$). The issue comes from the value of N itself : for plasmas or galaxies, N is respectively of the order of 10^{23} or 10^{11} (for Milky Way for instance). In both cases, so much equations form systems we cannot solve directly. That means we have to change our approach.

1.3 Different points of view

1.3.1 Different scales

Previously we described the problem with a *microscopic* approach : we considered separately each bodies, and applied classical mechanics laws. The construction of the problem is simple, and in some ways, intuitive. Yet for all that it eventually becomes really difficult to solve.

Another way to look at the problem is with the *macroscopic* approach. It is directly opposed to the microscopic viewpoint, where we study each elements individually. Here, we study objects and phenomenons at a bigger scale, so much that it becomes visible for naked eyes. Outside the scope of our subject, this approach leads to the use of hydrodynamics.

A middle ground between *microscopic* and *macroscopic* scales is the *mesoscopic* scale. The idea is to treat particles statistically, so we can focus on their global behavior. Mathematically, this viewpoint - which will be ours, leads us to use the *distribution function of particles* f.

1.3.2 The distribution function

The **distribution function** $f(t, \mathbf{x}, \mathbf{p})$ is defined as the number of particles in an elementary volume $d\mathbf{x}$ around $\mathbf{x}$ that has a momentum $d\mathbf{p}$ around $\mathbf{p}$ at time t. In other words, it is the number of particles in an elementary *phase-space* volume $d\mathbf{x}d\mathbf{p}$ at time t. Consider N particles confined in a volume Ω of $\mathbb{R}^3$ at time t. We have

$$N(t) = \int_{\Omega \times \mathbb{R}^3} f(t, \mathbf{x}, \mathbf{p}) d\mathbf{x} d\mathbf{p}. \tag{1.4}$$

At the same time, we introduce **the definition of the density** : it is the number of particles that have a position $d\mathbf{x}$ around $\mathbf{x}$ at a given time t. Given the previous definition, that means

$$\rho(t, \mathbf{x}) = \int_{\mathbb{R}^3} f(t, \mathbf{x}, \mathbf{p}) d\mathbf{p} \tag{1.5}$$

Next chapter deals with the study of f

Chapter 2

The Vlasov-Poisson system

The Vlasov-Poisson system describes the time evolution of f for a plasma, with no magnetic field. The only force at stake here is the electrical force due to the electic field.

2.1 How to establish the Vlasov's equation

To study Vlasov's equation and try to have a better understanding of it, let's see how one can establish it. Two approachs will be shown. In both cases, it can be shown for any value of d.

2.1.1 Using Liouville's Theorem

We assume the following statement : if collisions are neglected, the distribution function obeys Liouville's Theorem.

Liouville's Theorem *The distribution function is constant along any trajectory in phase space. That is :*

$$\frac{df}{dt} = \partial_t f + \sum_{i=1}^{d} \left(\frac{d\mathbf{x}_i}{dt} \partial_{\mathbf{x}_i} f + \frac{d\mathbf{p}_i}{dt} \partial_{\mathbf{p}_i} f \right) = 0 \tag{2.1}$$

A proof can be found in [4].

Developping the expression in (2.1) gives

$$\partial_t f + \mathbf{p} \cdot \nabla_{\mathbf{x}} f + m\mathbf{a} \cdot \nabla_{\mathbf{p}} f = 0 \tag{2.2}$$

Here $m\mathbf{a} = \frac{d\mathbf{p}}{dt}$ where $\mathbf{a}$ is the acceleration vector. Now we use Newton's second law. That is

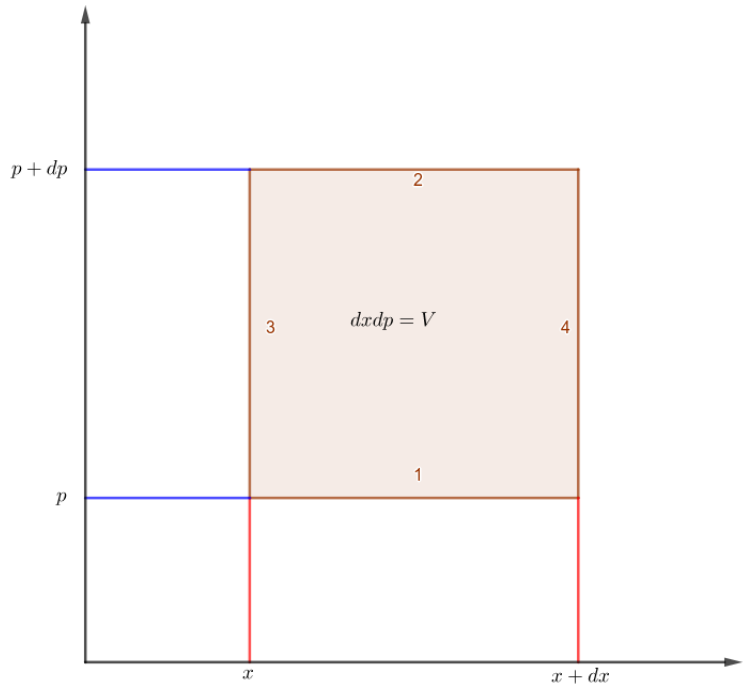
$$m\mathbf{a} = \mathbf{F}. \tag{2.3}$$

We assumed that the only force is $\mathbf{E}$. Plus, since we consider m as constant, that gives

$$\boxed{\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + \mathbf{E} \cdot \nabla_{\mathbf{v}} f = 0}$$
Vlasov's equation (2.4)

2.1.2 Hydrodynamic approach

Another way to obtain the equation is by studying directly a phase space elementary volume. It is impossible to draw a phase space if $d \geq 2$. Therefore, we are going to show it with $d = 1$. Let's consider the following phase space, provided with a volume $V = dxdp$:



Let's compute the rate of change of the number of particles in V. That means we compute the difference between the number of particles that go into V and the number of particles that leave V. That is :

$$df(t,x,p)dxdp = \left(f(t+dt,x,p) - f(t,x,p)\right) dxdp \tag{2.5}$$

For each side i of the area (i in $\{1,2,3,4\}$) we calculate the number of particles that go through it :

- Side 1 : $df(t,x,p)dxdp = df(t,x,p)dx\frac{dp}{dt}dt = df(t,x,p)dx\ a_x dt$ where a_x is the acceleration component over x.
- Side 2 : $df(t,x,p+dp)dxdp = df(t,x,p+dp)dx\ a_x dt$

- Side 3 : $df(t,x,p)dxdp = df(t,x,p)\frac{dx}{dt}dtdp = df(t,x,p)dp\ pdt$
- Side 4 : $df(t,x+dx,p)dxdp = df(t,x+dx,p)\frac{dx}{dt}dtdp = df(t,x+dx,p)dp\ pdt$

Eventually, according to (1.4), we have

$$df\,dxdp = -a_x df\,dxdt - pdf\,dpdt \tag{2.6}$$

This can be written as follows :

$$\partial_t f + p\partial_x f + a_x \partial_p f = 0 \tag{2.7}$$

From here, one can generelize this to higher dimension, which gives :

$$\partial_t f + \mathbf{p}\cdot\nabla_{\mathbf{x}} f + \mathbf{a}\cdot\nabla_{\mathbf{p}} f = 0 \tag{2.8}$$

This is the same left term as the equation (2.2). So, the same way, we indeed obtain Vlasov's equation (remember that the only external force at stake here is $\mathbf{E}$, and m is set to 1).

$$\boxed{\partial_t f + \mathbf{v}\cdot\nabla_{\mathbf{x}} f + \mathbf{E}\cdot\nabla_{\mathbf{v}} f = 0}$$
Vlasov's equation (2.9)

2.2 Remarks related to the equation

In the scope of this study there are few remarks to say. Even if our assumptions make the problem simpler (no magnetic field, no collisions, negligible relativistic effects...), this equation is still complex to study. Let's interest ourselves in the shape of the equation and see if it can be even more simplified.

The equation can be seen as the sum of two terms :

$$\partial_t f + \mathbf{v}\cdot\nabla_{\mathbf{x}} f + \mathbf{E}\cdot\nabla_{\mathbf{v}} f = 0 \tag{2.10}$$

The first term is the **free transport operator**. It describes how moves a free particle, that is to say a particle subjected to no external force. If there is no force at all, the particle moves in constant speed. That means that with no external force, the free transport operator equals zero. This is called the **free transport equation** :

$$\partial_t f + \mathbf{v}\cdot\nabla_{\mathbf{x}} f = 0 \tag{2.11}$$

The second term is the term that describes the action of external force(s) on the particle. This term is variable according to the problem one would study and the assumption made about the gas. For us, since we only consider electrical field, this term is defined using the Poisson's equation :

$$\nabla_{\mathbf{x}} \cdot \mathbf{E} = -\rho \tag{2.12}$$

Next section deals with some results related to the free transport equation.

2.3 Free transport equation

In this section, since we deal with the free transport equation, we assume there are no external force to interact with particles.

2.3.1 Case for x in $\mathbb{R}^d$

Here, we interest ourselves in N particles gathered in an small initial phase space volume $d\mathbf{x}d\mathbf{v}$. We are going to prove the following result :

Theorem 1. *For any* $\mathbf{x}$ *in* $\mathbb{R}^d$*, the denstity* $\rho(t, \mathbf{x})$ *converges towards zero as time goes to infinity. That is to say :*

$$\forall\ \mathbf{x} \in \mathbb{R}^d,\ \ \rho(t, \mathbf{x}) \underset{t\to\infty}{\longrightarrow} 0$$

Proof. This result relies on Bardos and Degond's paper [2] and Cédric Mouhot's work [1]. The free transport equation with the initial configuration of f form the following system

$$\begin{cases} \partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f = 0 \\ f(t = 0, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{x}, \mathbf{v}) \end{cases}. \tag{2.13}$$

Its solution reads

$$f(t, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{x} - t\mathbf{v}, \mathbf{v}). \tag{2.14}$$

To prove the theorem, let's use the definition of density :

$$\rho(t, \mathbf{x}) = \int_{\mathbb{R}^d} f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}. \tag{2.15}$$

From here, we can substitute equality (2.14) to (2.15). That is :

$$\rho(t, \mathbf{x}) = \int_{\mathbb{R}^d} f_0(\mathbf{x} - t\mathbf{v}, \mathbf{v}) d\mathbf{v}. \tag{2.16}$$

Thus, we can also affirm that

$$\rho(t,\mathbf{x}) \leq \int_{\mathbb{R}^d} \sup_{\mathbf{v}'\in\mathbb{R}^d} f_0(\mathbf{x}-t\mathbf{v},\mathbf{v}')d\mathbf{v}. \tag{2.17}$$

In a view to simplify the writing, let g be the function defined as

$$g(\mathbf{y}) = \sup_{\mathbf{v}'\in\mathbb{R}^d} f_0(\mathbf{y},\mathbf{v}'). \tag{2.18}$$

Substituing (2.18) to (2.17), that gives

$$\rho(t,\mathbf{x}) \leq \int_{\mathbb{R}^d} g(\mathbf{x}-t\mathbf{v})d\mathbf{v}. \tag{2.19}$$

Now we apply the change of variable

$$\mathbf{y} = \mathbf{x} - t\mathbf{v} \Rightarrow \mathbf{v} = \frac{\mathbf{x}-\mathbf{y}}{t} \Rightarrow \frac{d\mathbf{v}}{d\mathbf{y}} = \left(\frac{-1}{t}\right)^d. \tag{2.20}$$

Detailing the computation of inequality (2.19), that gives

$$\begin{aligned}\rho(t,\mathbf{x}) &\leq \underbrace{\int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{+\infty}}_{d\ times} g(\mathbf{x}-t\mathbf{v})d\mathbf{v} \\ &\leq \int_{+\infty}^{-\infty}\cdots\int_{+\infty}^{-\infty} g(\mathbf{y})\left(\frac{-1}{t}\right)^d d\mathbf{y}\end{aligned} \quad . \tag{2.21}$$

Note that we swapped bounds because of the change of variable. That implies

$$\rho(t,\mathbf{x}) \leq (-1)^{2d}\frac{1}{t^d}\int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{+\infty} g(\mathbf{y})d\mathbf{y} \tag{2.22}$$

where bounds are swapped back, which explains the $2d$ power. For any d, $(-1)^{2d} = 1$. Therefore this is equivalent to

$$\rho(t,\mathbf{x}) \leq \frac{1}{t^d}\int_{\mathbb{R}^d} g(\mathbf{y})d\mathbf{y}. \tag{2.23}$$

Now we remplace g by its definition. That gives

$$\rho(t,\mathbf{x}) \leq \frac{1}{t^d}\int_{\mathbb{R}^d} \sup_{\mathbf{v}'\in\mathbb{R}^d} f_0(\mathbf{y},\mathbf{v}')d\mathbf{y} \tag{2.24}$$

which implies, since everything is positive,

$$||\rho(t,\cdot)||_{L^\infty} \leq \frac{1}{t^d}||f_0(t,\cdot,\cdot)||_{L^1L^\infty}. \tag{2.25}$$

hence the result.

■

2.3.2 Case for x in $\mathbb{T}^d = \mathbb{R}^d/\mathbb{Z}^d$

Here, the problem is the same as previously, except that $\mathbf{x} \in \mathbb{T}^d = (\mathbb{R}/\mathbb{Z})^d$, which represent the set of periodical function of period 1. We want to proove the following theorem :

Theorem 2. *For any* $\mathbf{x}$ *in* $\mathbb{T}^d$*, the density* $\rho(t,\mathbf{x})$ *converges towards its average as* $t \to \infty$*. That is to say :*

$$\forall\ \mathbf{x} \in \mathbb{T}^d,\ \rho(t,\mathbf{x}) \underset{t\to\infty}{\longrightarrow} \bar{\rho}$$

where $\bar{\rho}$ *is the average of* $\rho_0(\mathbf{x}) = \rho(0,\mathbf{x})$*.*

Proof. For the whole proof, we assume $d = 1$. Therefore the gradient operator $\nabla_{\mathbf{x}}$ becomes is a simple partial derivative ∂_x, and $\mathbf{x}$ and $\mathbf{v}$ become respectively x and v. But it can be generalised to any dimension. Consider the free transport equation

$$\partial_t f + v\partial_x f = 0. \tag{2.26}$$

Since $x \in \mathbb{T}$, we use Fourier series for f which results into

$$f(t,x,v) = \sum_{k=-\infty}^{+\infty} \hat{f}_k(t,v)e^{2\pi ikx}. \tag{2.27}$$

We can then substitute (2.25) in (2.24) :

$$\partial_t\left(\sum_{k=-\infty}^{+\infty} \hat{f}_k(t,v)e^{2\pi ikx}\right) + v\partial_x\left(\sum_{k=-\infty}^{+\infty} \hat{f}_k(t,v)e^{2\pi ikx}\right) = 0. \tag{2.28}$$

This is equalivalent to

$$\sum_{k=-\infty}^{+\infty}\left[\partial_t(\hat{f}_k(t,v)e^{2\pi ikx}) + v\partial_x(\hat{f}_k(t,v)e^{2\pi ikx})\right] = 0. \tag{2.29}$$

This implies that for any $k \in \mathbb{Z}$,

$$\partial_t \left(\hat{f}_k(t,v) e^{2\pi i k x} \right) + v \partial_x \left(\hat{f}_k(t,v) e^{2\pi i k x} \right) = 0. \tag{2.30}$$

The second term can be computed easily. Thus, we have

$$e^{2\pi i k x} \left(\partial_t \hat{f}_k(t,v) + 2\pi i k v \hat{f}_k(t,v) \right) = 0. \tag{2.31}$$

The solution of this equation is :

$$\hat{f}_k(t,v) = \hat{f}_k(0,v) e^{-2\pi i k v t}. \tag{2.32}$$

Therefore, using the equality (2.25), that gives

$$f(t,x,v) = \sum_{k=-\infty}^{+\infty} \hat{f}_k(0,v) e^{2\pi i k (x - vt)}. \tag{2.33}$$

We have thus another expression for f. Now let's work on the density.
On one hand, we use its expression as a Fourier serie. That is

$$\rho(t,x) = \sum_{k=-\infty}^{+\infty} \hat{\rho}_k(t) e^{2\pi i k x}. \tag{2.34}$$

On the other hand, we use the definition :

$$\rho(t,x) = \int_{\mathbb{R}} f(t,x,v) dv. \tag{2.35}$$

Now, we substitute the equality (2.31) to (2.33). That gives :

$$\rho(t,x) = \int_{\mathbb{R}} \left(\sum_{k=-\infty}^{+\infty} \hat{f}_k(0,v) e^{-2\pi i k (x - vt)} \right) dv \tag{2.36}$$

Since we supposed f as C^2 fonction with compact support, we can swap the integral with the sum. That gives :

$$\begin{aligned}\rho(t,x) &= \sum_{k=-\infty}^{+\infty} \int_{\mathbb{R}} \hat{f}_k(0,v) e^{-2\pi i k(x-vt)} dv \\ &= \sum_{k=-\infty}^{+\infty} \left(\int_{\mathbb{R}} \hat{f}_k(0,v) e^{-2\pi i k t v} dv \right) e^{2\pi i k x}\end{aligned} . \tag{2.37}$$

We identify this equality and (2.32). That implies :

$$\hat{\rho}_k(t) = \int_{\mathbb{R}} \hat{f}_k(0,v) e^{-2\pi i k t v} dv. \tag{2.38}$$

Our assumptions about f allow to use Riemann-Lebesgue Theorem. It says that, for any $k^* \neq 0$,

$$\hat{\rho}_{k^*}(t) \underset{t\to\infty}{\longrightarrow} 0. \tag{2.39}$$

However, this does not imply the convergence of the sum (2.32). To show this, let's focus on the expression of $\hat{\rho}_k$ given by (2.36). Let $\mu(v)$ and $\nu(v)$ defined as follows :

$$\begin{cases} \mu(v) = -\frac{1}{2\pi i k t} e^{-2\pi i k t v} \\ \nu(v) = \hat{f}_k(0,v) \end{cases} . \tag{2.40}$$

The expression of $\hat{\rho}_k$ becomes

$$\hat{\rho}_k(t) = \int_{\mathbb{R}} (\mu' \nu)(v) dv. \tag{2.41}$$

Since f is in C^2, $\hat{f}_k$ is in C^2 as well. Therefore, we integrate by part $\hat{\rho}_k$:

$$\hat{\rho}_k(t) = \Big[(\mu\nu)(v) \Big]_{-\infty}^{+\infty} - \int_{\mathbb{R}} (\mu\nu')(v) dv. \tag{2.42}$$

The compact support of f implies that **the first term equals zero**.
Using the definition of μ and ν, the remaining term is :

$$\hat{\rho}_k(t) = \frac{1}{2\pi i} \frac{1}{kt} \int_{\mathbb{R}} e^{-2\pi i k t v} \hat{f}_k'(0,v) dv. \tag{2.43}$$

We see that for any $k \neq 0$ we can integrate $\hat{\rho}_k$ by part as many time as $\hat{f}_k$ is derivable. The result is that supposing f in C^n, $\hat{\rho}_k$ obeys the following equality :

$$\forall k \neq 0,\ \hat{\rho}_k(t) = \frac{1}{(2\pi i)^n} \frac{1}{(kt)^n} \int_{\mathbb{R}} e^{-2\pi i k t v} \hat{f}_k^{(n)}(0, v) dv. \tag{2.44}$$

Given this result, we thus know the speed of convergence of $\hat{\rho}_k(t)$ towards zero as time goes to infinity. This expression can be substitute in (2.35). Since we supposed f in C^2, that gives :

$$\rho(t, x) = \sum_{k=-\infty}^{+\infty} \frac{1}{(2\pi i)^2} \frac{1}{(kt)^2} \int_{\mathbb{R}} e^{-2\pi i k t v} \hat{f}_k^{(2)}(0, v) dv. \tag{2.45}$$

We know that for any $k \neq 0$, this sum converges towards zero. That implies :

$$\rho(t, x) \underset{t \to \infty}{\longrightarrow} \lim_{t \to \infty} \hat{\rho}_0(t). \tag{2.46}$$

By definition of Fourier series (see the appendix), $\hat{\rho}_0$ is the average of ρ.

We thus showed that for x in $\mathbb{T}$, the density converges towards its average which is constant in time. ■

2.3.3 Results comparison

Previously, according the set in which we consider $\mathbf{x}$, the density behaves differently as time goes on. Let's sum up.

If $\mathbf{x}$ is in $\mathbb{R}^d$, the density converges towards zero. This seems intuitive : if there is no force to interact with particle, they all should travel in rectilign uniform motion in every direction. Thus, the distance between each of them would increase indefinitely.

However, if $\mathbf{x}$ is in $\mathbb{T}^d$, the density converges towards an average. According to its definition, that means in each elementary space volume $d\mathbf{x}$, the density tends to oscillate less and less. This oscillations drop is a phenomenon known as **the Landau damping**. This is what next chapter deals with.

Chapter 3

The Landau damping

The Landau damping, named after the Russian physicist Lev Davidovich Landau who first suggested it in 1946, is a phenomenon that descibes an exponential decrease with respect to time (which is the actual meaning of "damping") of density oscillations in a plasma. Previously, we showed this phenomenon mathematically considering only the free transport equation. However, this result can be extend to the case where an external electrical force is applied on particles.

3.1 Intuition

One can interpret the phenomenon as follows : it is directly due to interactions between electric field and particles (this is why the effect is important in plasma. Since particles are ions, they have electric charge and are thus more sensitive to electric field). Imagine that particles have their initial velocities all set slightly larger or smaller than the electric field phase velocity. This way, as time goes on, every particles travelling faster than the wave would see themselves yielding their energy to the electric wave, and thus slowing down. The exact opposite happens as well for particles thar are initially slower than the electric wave, in which case the wave would see itself yielding its energy (and its speed decrease) to the particle, which would gain speed.

At first sight, the result of these reciprocal energy transfers should cancel each other out. But this is true only if we consider that there are as much particles that go faster than the wave as particles that go slower. However, the speed distribution of particles is defined by the Maxwell-Boltzman distribution (typically, a Gaussian law). In this case, there are more particles that go faster than the electric wave, than particles that go slower. This initial distribution results in a decay of the electric field that eventually leads to its stability.

3.2 Landau computations

In this section we will give a detailed glimpse of computations that led Landau to his discovery. References will be provided at the end of the report. Yet, one should note than some computation steps are not detailed in these papers. Since these computations are very complex, my work contributed first of all to provide more details by adding many steps in the progress of Landau's proof. This being said, with a view to consistency with the report, notation will differ from original Landau's work. Here we follow the book [3] section 34.

We consider the Vlasov's equation with a self consistent field for initial conditions. We consider the case with a purely potential electric field $\mathbf{E} = -\nabla_{\mathbf{x}}\phi$, so only the electron distribution is being changed. Moreover, we assume that the initial perturbation is very small. Therefore, our initial distribution looks as follows :

$$f(0, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{v}) + g(\mathbf{x}, \mathbf{v}) \tag{3.1}$$

where $f_0(\mathbf{v})$ corresponds to the initial equilibirum state, and $g(\mathbf{x}, \mathbf{v})$ is the initial perturbation such as $g \ll f_0$. This perturbation remains small subsequently. We are thus looking for a distribution function with the following form :

$$f(t, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{v}) + \delta f(t, \mathbf{x}, \mathbf{v}). \tag{3.2}$$

We can then write the system, with our previous assumptions and considering the Poisson's equation. That is :

$$\begin{cases} \partial_t(\delta f) + \mathbf{v} \cdot \nabla_{\mathbf{x}}(\delta f) + \nabla\phi \cdot \nabla_{\mathbf{v}}(\delta f_0) = 0 \\ \Delta\phi = \int (\delta f) d\mathbf{v} \end{cases} . \tag{3.3}$$

As we did in chapter 2, we can use Fourier series expression for δf and ϕ. That is :

$$\begin{cases} \delta f(t, \mathbf{x}, \mathbf{v}) = \sum_{k=-\infty}^{+\infty} \hat{f}_k(t, \mathbf{v}) e^{2\pi i \mathbf{k}\cdot\mathbf{x}} \\ \phi(t, \mathbf{x}) = \sum_{k=-\infty}^{+\infty} \hat{\phi}_k(t) e^{2\pi i \mathbf{k}\cdot\mathbf{x}} \end{cases} . \tag{3.4}$$

That implies, $\forall k \in \mathbb{Z}^d$

$$\begin{cases} \partial_t \hat{f}_k e^{2\pi i \mathbf{k}\cdot\mathbf{x}} + i\mathbf{k} \cdot \mathbf{v} \hat{f}_k e^{2\pi i \mathbf{k}\cdot\mathbf{x}} + i\hat{\phi}_k e^{2\pi i \mathbf{k}\cdot\mathbf{x}} \mathbf{k} \cdot \nabla_{\mathbf{v}}(\delta f_0) = 0 \\ -k^2 \hat{\phi}_k = \int \hat{f}_k d\mathbf{v} \end{cases}$$

which is equivalent to :

$$\begin{cases} \partial_t \hat{f}_k + i\mathbf{k}\cdot\mathbf{v} f_k + i\hat{\phi}_k \mathbf{k}\cdot\nabla_{\mathbf{v}}(\delta f_0) = 0 \\ k^2\hat{\phi}_k = -\int \hat{f}_k d\mathbf{v} \end{cases} \tag{3.5}$$

To solve these equations, we can use Laplace transform of $\hat{f}_k$ noted f_{wk} as follows :

$$f_{wk}(\mathbf{v}) = \int_0^{+\infty} \hat{f}_k(t,\mathbf{v}) e^{iwt} dt \tag{3.6}$$

where the integration is over $\mathbb{R}^+$ because we integrate with respect to $t \geq 0$. The inverse transformation is given by

$$\hat{f}_k(t,\mathbf{v}) = \frac{1}{2\pi}\int_{-\infty+i\sigma}^{+\infty+i\sigma} f_{wk}(\mathbf{v}) e^{-iwt} dw. \tag{3.7}$$

The intergral is taken along straight line in the complexe plan, parallel to and above the real axis ($\sigma > 0$), and also passing above all f_{wk} singularities. **Laplace transform definition is provided in the appendix.**

From here, noting that

$$\begin{aligned} \int_0^{+\infty} \partial_t \hat{f}_k e^{-iwt} dt &= \left[\hat{f}_k e^{-iwt}\right]_0^{+\infty} - iw\int_0^{+\infty} \hat{f}_k e^{-iwt} dt \\ &= -g_k - iw f_{wk} \end{aligned}, \tag{3.8}$$

we multiply (3.5) by e^{-iwt}, which gives :

$$\begin{cases} \partial_t \hat{f}_k e^{-iwt} + i\mathbf{k}\cdot\mathbf{v}\hat{f}_k e^{-iwt} + i\hat{\phi}_k\mathbf{k}\cdot\nabla_{\mathbf{x}}(\delta f_0)e^{-iwt} = 0 \\ k^2\hat{\phi}_k e^{-iwt} = -e^{-iwt}\int \hat{f}_k d\mathbf{v} \end{cases} . \tag{3.9}$$

Now we integrate with respect to t the system above. Let's detail computations.

For the first line of (3.9) : we first split the integral into three simpler ones :

$$\underbrace{\int \partial_t \hat{f}_k e^{-iwt} dt}_{I_1} + \underbrace{\int i\mathbf{k}\cdot\mathbf{v}\hat{f}_k e^{-iwt} dt}_{I_2} + \underbrace{\int i\hat{\phi}_k\mathbf{k}\cdot\nabla_{\mathbf{x}}(\delta f_0)e^{-iwt} dt}_{I_3} = 0$$

where the integration constant is set to 0. Now we compute each $I_{n\in(1,2,3)}$.

- I_1 corresponds to (3.8) so

$$I_1 = -g_k - iwf_{wk}. \tag{3.10}$$

- For I_2 :

$$I_2 = i\mathbf{k}\cdot\mathbf{v}\int \hat{f}_k e^{-iwt}dt = i\mathbf{k}\cdot\mathbf{v}f_{wk}. \tag{3.11}$$

- $I_3 = i\int \hat{\phi}_k \mathbf{k}\cdot\nabla_{\mathbf{x}}(\delta f_0)e^{-iwt}dt$ in which we can identify ϕ_{wk} as shown in (3.6). Thus,

$$I_3 = i\phi_{wk}\mathbf{k}\cdot\nabla_{\mathbf{x}}(\delta f_0). \tag{3.12}$$

We gather results (3.10), (3.11) and (3.12). That is

$$-g_k - iwf_{wk} + i\mathbf{k}\cdot\mathbf{v}f_{wk} + i\phi_{wk}\mathbf{k}\cdot\nabla_{\mathbf{x}}(\delta f_0) = 0$$

which results into

$$f_{wk} = \frac{1}{i(\mathbf{k}\cdot\mathbf{v}-w)}\Big(g_k - i\phi_{wk}\mathbf{k}\cdot\nabla_{\mathbf{x}}(\delta f_0)\Big). \tag{3.13}$$

For the second line of (3.9), computations are simpler :

$$\begin{aligned}\int k^2\hat{\phi}_k e^{-iwt}dt &= k^2\int \hat{\phi}_k e^{-iwt}dt \\ &= \int e^{-iwt}\int \hat{f}_k d\mathbf{v}dt \\ &= \iint \hat{f}_k e^{-iwt}dtd\mathbf{v} \\ &= \int f_{wk}\ d\mathbf{v}\end{aligned}. \tag{3.14}$$

Eventually (3.13) and (3.14) give

$$\begin{cases} f_{wk} = \frac{1}{i(\mathbf{k}\cdot\mathbf{v}-w)}\left(g_k - i\phi_{wk}\mathbf{k}\cdot\nabla_{\mathbf{x}}(\delta f_0)\right) \\ \phi_{wk} = -\frac{1}{k^2}\int f_{wk}\ d\mathbf{v} \end{cases}. \tag{3.15}$$

Before going further, we assume the following result (see [3] section 29) :

$$\epsilon_l(w,k) = 1 - \frac{1}{k^2}\int \frac{1}{(\mathbf{k}\cdot\mathbf{v}-w)}\mathbf{k}\cdot\nabla_{\mathbf{v}}(\delta f_0)d\mathbf{v} \tag{3.16}$$

where $\epsilon_l(w,k)$ is the longitudinal permitivitty of plasma with a stationary distribution f_0. From (3.12) we can substitute f_{wk} in ϕ_{wk} expression, such as

$$\begin{aligned}\phi_{wk} &= -\frac{1}{k^2}\int \frac{1}{i(\mathbf{k}\cdot\mathbf{v}-w)}\Big(g_k - phi_{wk}\mathbf{k}\cdot\nabla_{\mathbf{v}}(\delta f_0)\Big)d\mathbf{v} \\ &= -\frac{1}{k^2}\left(\int \frac{g_k}{i(\mathbf{k}\cdot\mathbf{v}-w)}d\mathbf{v} - \int \frac{1}{(\mathbf{k}\cdot\mathbf{v}-w)}\phi_{wk}\mathbf{k}\cdot\nabla_{\mathbf{v}}(\delta f_0)d\mathbf{v}\right). \\ &= -\frac{1}{k^2}\left(\int \frac{g_k}{i(\mathbf{k}\cdot\mathbf{v}-w)}d\mathbf{v} - \phi_{wk}\int \frac{1}{(\mathbf{k}\cdot\mathbf{v}-w)}\mathbf{k}\cdot\nabla_{\mathbf{v}}(\delta f_0)d\mathbf{v}\right)\end{aligned} \tag{3.17}$$

Now, using (3.16), we see that

$$k^2(1-\epsilon_l) = \int \frac{1}{(\mathbf{k}\cdot\mathbf{v}-w)}\mathbf{k}\cdot\nabla_{\mathbf{v}}(\delta f_0)d\mathbf{v}. \tag{3.18}$$

We can identify (3.18) with (3.17). We see that

$$\phi_{wk} = -\frac{1}{k^2}\left(\int \frac{g_k}{i(\mathbf{k}\cdot\mathbf{v}-w)}d\mathbf{v} - \phi_{wk}k^2(1-\epsilon_l)\right). \tag{3.19}$$

By gathering each ϕ_{wk} that gives

$$\phi_{wk}\left(1-(1-\epsilon_l)\right) = -\frac{1}{k^2}\int \frac{g_k}{i(\mathbf{k}\cdot\mathbf{v}-w)}d\mathbf{v} \tag{3.20}$$

which provides the expression of ϕ_{wk} :

$$\boxed{\phi_{wk} = -\frac{1}{k^2\epsilon_l}\int \frac{g_k}{i(\mathbf{k}\cdot\mathbf{v}-w)}d\mathbf{v}} \tag{3.21}$$

Plus, substituing (3.13) in (3.7), that gives an expression of f_k :

$$\boxed{\hat{f}_k(t,\mathbf{v}) = \frac{1}{2\pi}\int_{-\infty+i\sigma}^{+\infty+i\sigma}\left[\frac{1}{i(\mathbf{k}\cdot\mathbf{v}-w)}\Big(g_k - i\phi_{wk}\mathbf{k}\cdot\nabla_{\mathbf{x}}(\delta f_0)\Big)\right]e^{-iwt}dw} \tag{3.22}$$

We want to study the time evolution of $\phi_k(t)$. To do so, we use the formula given by (3.7) :

$$\hat{\phi}_k(t) = \frac{1}{2\pi}\int_{-\infty+\sigma i}^{+\infty+\sigma i}\phi_{wk}e^{-iwt}dw. \tag{3.23}$$

Formally, we should study properties of ϕ_{wk} before using this formula. These properties are discussed in [3] at the end of section 34. They allow to affirm that the expression (3.21) is well defined, and finds its only poles at zeros of $\epsilon_l(w,k)$. It is shown that ϕ_{wk} can be anatycally continued to the whole complex plan. Doing so, the contour integration in (3.23) can be moved sufficiently so it does not cross any pole of the integrand. Let $w_k = w'_k + iw''_k$ be the root of $\epsilon_l(w,k)$ that has the smallest imaginary part. It results into the fact that the field perturbation is exponentially damped, following the law

$$\hat{\phi}_k(t) \propto e^{iw'_k t} e^{-|w''_k|t} \tag{3.24}$$

which implies that the damping rate of the electric field is given by $|w''_k|$.

Now, consider the distribution function. The equality (3.22) shows a pole on its integrand at the point $w = \mathbf{k} \cdot \mathbf{v}$. This pole determines the behavior of the integral as $t \to \infty$. That is

$$\hat{f}_k(t, \mathbf{v}) \propto e^{-i\mathbf{k}\cdot\mathbf{v}t}. \tag{3.25}$$

That means that « the distribution function becomes a more and more rapidly ocillating function of $\mathbf{v}$. » ([3] p. 150) Therefore, as the density is the integral over $\mathbf{v}$ of the distribution function, it is damped.

3.3 Application : plasma stability in a tokamak

In this section we show an example of how Landau damping (and plasma in general) are used in the industry. This field of research is topical, and its actual application is still in an experimental and theoretical framework. Its eventual goal is to produce energy thanks to nuclear fusion.

Nuclear fusion is a reaction in which several atomic nuclei collide, which results into a new heavier nucleus and some subatomic particles (neutrons and protons). But the mass of the reaction products is smaller than the sum of all reactants' mass. This difference translates into a release of energy, quantified by Albert Einstein's equation $E = mc^2$.

A tokamak[1] is a torus-shaped device designed to confine a plasma at very high temperature using the power of electromagnectic field, in order to make particles nuclei collide and produce energy thanks to their fusion. It was invented in 1950 by Soviet physicists Igor Tamm and Andrei Sakharov. The torus shape was not the first idea, but the result of the resolution of several problems.

[1]The name comes from the russian «**то**роидальная **ка**мера с **ма**гнитными **к**атушками» which stands for *Toroidal chamber with magnetic coils.*

The main difficulty lies in the fact that to obtain a nuclear fusion, the gas must be heated to extreme temperatures : around 10^7 Kelvin. So far, we don't know any material that can handle such temperature. Because of this, gas must be confined within an isolated volume which shall not be in contact with any material.

The basic idea to confine particles is to use the magnetic field produced by a solenoid :

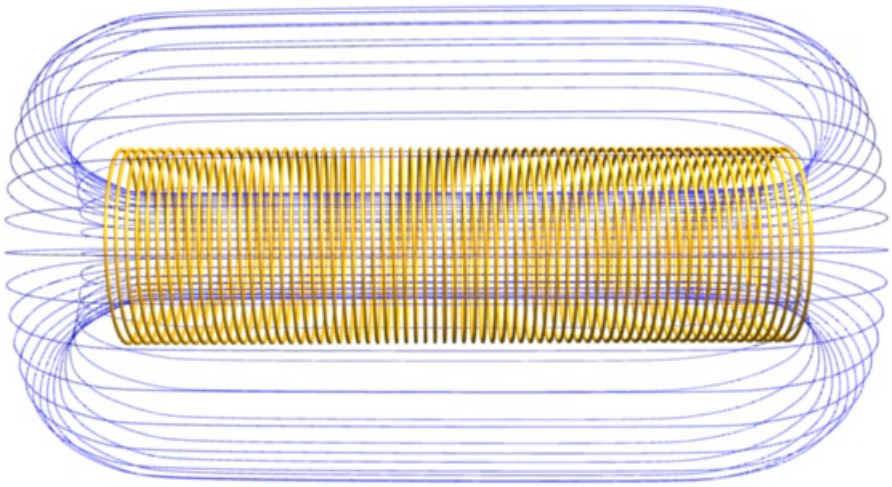


Figure 3.1: Blue lines represent the magnetic field due to the current through the solenoid.

but particles remain free at each borders. Therefore, borders are linked to each other so it eventually shapes a torus in which the magnetic field is confined :

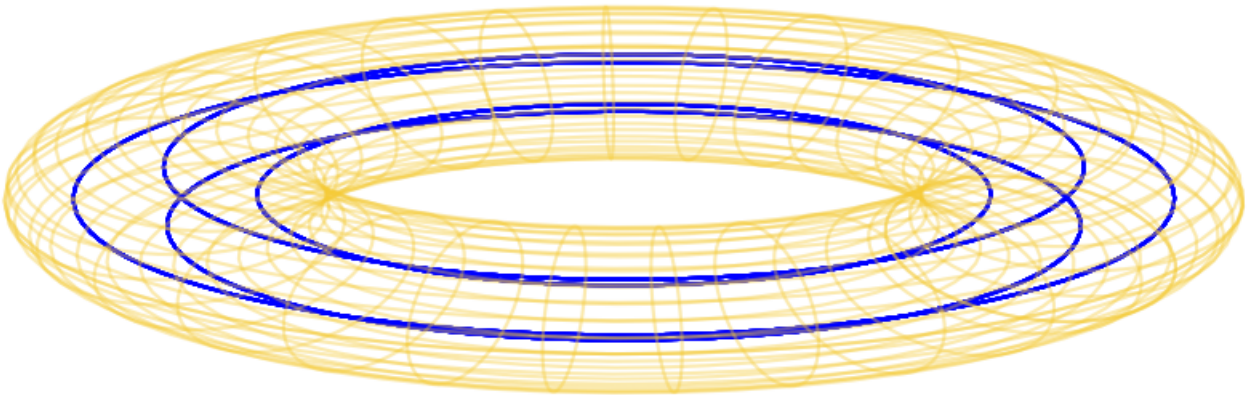


Figure 3.2: Magnetic field (blue lines) confined within the torus.

but from here, several problems show up.

First, by bending the solenoid, we stretch its outer part. The magnetic field is not uniform anymore, but gets stronger as it gets closer to the inner radius (coils are tighter to each other level with inner radius). The plasma will thus see itself flatten as it is close to the inner radius. Therefore, chamber shape must be adapted so *it looks like* a torus, but with a flatter inner edge.

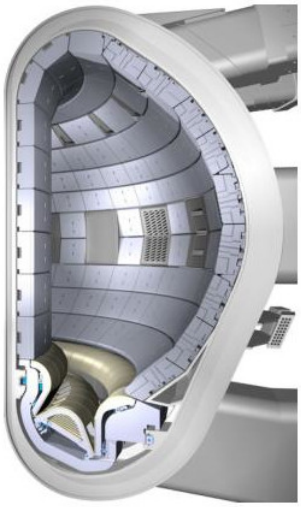


Figure 3.3: Cross-section of ITER tokamak preview.
ITER (International Thermonuclear Experimental Reactor) is an international nuclear fusion research project, made to be the world's largest experimental fuion reactor. See [5] for more details.

Then, as particles spin throughout the torus because of the magnetic field (say $\mathbf{B_1}$), centrifugal force pushs them away towards the exterior. The confinement is broken and torus wall might be damaged. To offset this, an electrical current is sent through a veritcal axis at the center of torus. This current creates a magnetic field ($\mathbf{B_2}$) in the torus which, by induction, creates an electrical current that goes through the plasma itself. This new current triggers a third magnetic field ($\mathbf{B_3}$) shaped in circle around the plasma. These magnetic field ($\mathbf{B_1}$ and $\mathbf{B_3}$) result in a spinning magnetic field around the plasma. It looks as follows :

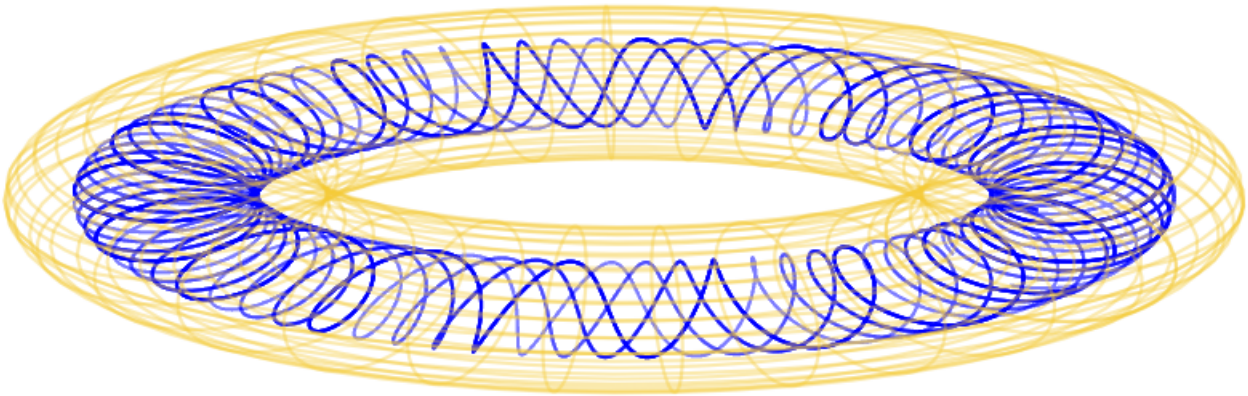


Figure 3.4: In blue, some resulting magnetic field lines

Thus, any particle which sees itself going away would be taken back towards the inner radius. One can assimilate this with Moon's orbit around Earth : it is actually falling towards Earth but at the same time, it's spinning around it so it is like a perpetual fall. In the torus, it's the same idea : particles «fall» towards the wall but spin at the same time so are taken back to the inner radius before repeating the same scenario. This offset is not perfect, but sufficient to allow the confinement to last long enough.

In order to keep an operating tokamak, temperature must be hold at a very high temperature. To do so, several methods are used :

- The electrical current that goes through the plasma heats it due to the Joule effect. But as its temperature increases, its resistance decreases so it becomes harder to heat.
- A very fast compression of the plasma (by moving the confined area towards to the inner radius) brings ions closer to each other, heating the plasma.
- Heating by high frequency waves sent through the torus at specific frequences so their energy can be transferred to the plasma.
- If the tokamak comes to produce more energy that it needs, a part of the excess might be used to hold the temperature at its operating level.

The point is, regarding the intensity of electric currents and magnetic fields in play, tokamaks are subject to potential instabilities. Landau damping (which actually occurs for ions, not only electrons) is one of the most important stabilizing effect in plasma. That explain why it is of such interst for researchers (as the whole theory of plasma in general).

Tokamaks show many advantages in comparison to current nuclear power stations. The high precision required to perform the reaction implies that there is no serious accident that could affect human lives, since any perturbation, whatever it might be, would result in an immediate stop of the reaction. The same remark stands for potential radioactive contamination, since the amount of radioactive matter in the tokamak is of the order of the gram.

At this time, *deuterium* ${}^2H$ and *tritium* ${}^3H$ has been took on as candidate of fusible material. Tritium is the only radioactive element of both, but is not a problem since it's produced only inside the tokamak, and has «only» a 12-years half-life, which is far better than current nuclear wastes. This would allow, if needed, to be able to recycle it within a century.

Chapter 4

Simulations

This chapter deals with every simulations I computed during the internship. They were my first task so I could be able to visualize the problem and thus, have a better understanding of the whole topic. This being said, **these simulations are only a way to approach the topic**. Deeper numerical computations would have required more time and this is not the aim of this internship.

I used Python for every simulations. This was a personnal choice. Of all languages I knew it seemed to be the best one to achieve my ends.

I worked on simulation on both periodical phase space and periodical classical space. Phase space is a space for which coordinates are position $\mathbf{x}$ and velocity $\mathbf{v}$. Since $\mathbf{x}$ and $\mathbf{v}$ are respectively in $\mathbb{R}^d$ and $\mathbb{T}^d$, phase space is a $2d$-dimensions space. Therefore, we can't draw it if d is greater that 1.

4.1 Phase space evolution

4.1.1 Results and illustrations

As we work in dimension 1, $\mathbf{x}$ and $\mathbf{v}$ have only one component. We consider N particles gathered in a small area, with random velocities. On phase space, that looks like this :

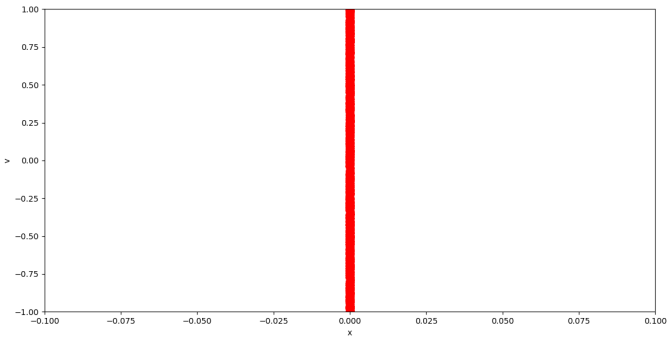


Figure 4.1: Initial phase space configuration with N = 10^4.
Absciss is the position, ordinate is the velocity.
Particles are represented by red dots.

Here, initial position bounds are $\pm 10^{-3}$, and velocity bounds are ± 1. These values are arbitrary, and can be changed in the code. **Note that velocity is not supposed to be as rigourously bounded. This is to simplify the program.** As time goes on, phase space evolves as follows :

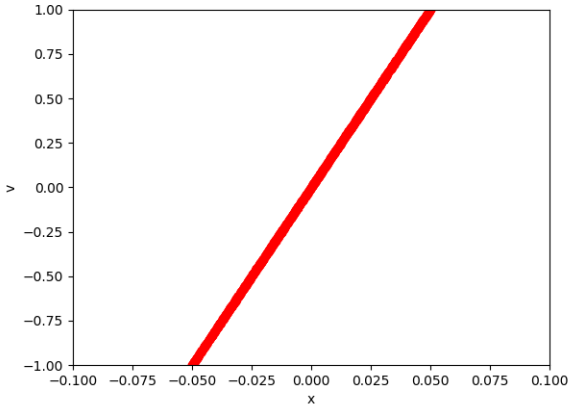


Figure 4.2: $t = 0.05$

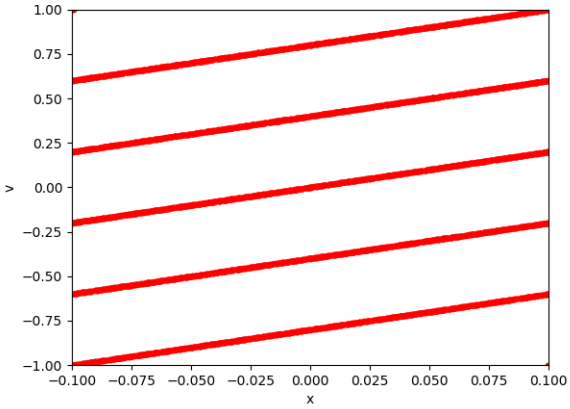


Figure 4.3: $t = 0.5$

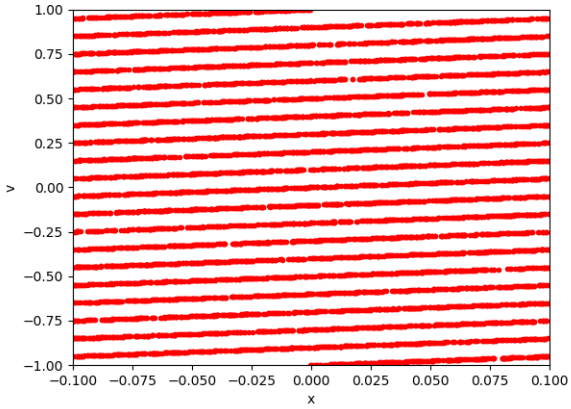


Figure 4.4: $t = 2$

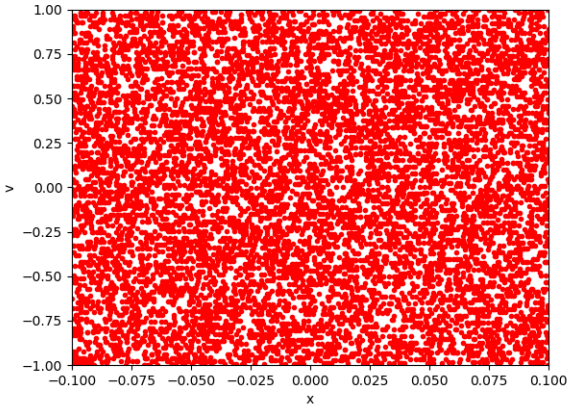


Figure 4.5: $t \gg 10$

It seems particles tends to occupy phase space homogeneously. Obiously, this is due to the fact that we are working in a **periodical** space. In $\mathbb{R}$, particles should move away from their initial position at a given speed, and distance themselves from each other. This is what is shown in Chapter 2, subsection 2.3.1.

4.1.2 Code details

Here is the code for the phase space evolution simulation :

```
1  import random
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import matplotlib.animation as animation
5
6  N = 10000 # Number of particles
7
8  xmax = 0.1  # Bounds
9  xmin = -xmax
10 ymax = 1
11 ymin = -ymax
12
13 t = 0.1 # Time step
14
15 x = (np.random.rand(N) -0.5)*2*xmax*0.01 # Initializes x and v
16 v = (np.random.rand(N) -0.5)*2*ymax
17
18 axes = plt.gca()  # Plot settings
19 axes.set_xlim([xmin,xmax])
20 axes.set_ylim([ymin,ymax])
21 plt.xlabel('x')
22 plt.ylabel('v')
23
24 def animate(x,v):   # Describes how to move from a position to another
25   for k in range(N):
26     x[k]+=t*v[k]
27     if x[k]>xmax:
28       x[k]-=2*xmax
29     if x[k]<xmin:
30       x[k]+=2*xmax
31   return x
32
33 def algo(x,v):      # Calls 'animates' for each itertion
34   n=0
35   while True:
36     if n == 0:
37       line, = plt.plot(x,v,'.', color='r')
38       n+=1
39       plt.pause(1)
40     else:
41       line.set_xdata((animate(x,v)))
42       plt.pause(0.5)
43
44 algo(x,v)
45 plt.show()
```

Line 6 The number of particles (N) is set.

Lines 8 to 11 Position and velocity bounds are set (respectively ($xmin$,$xmax$) and ($ymin$,$ymax$).

Line 13 Time step is chosen (determines the speed of the animation)

Lines 15 and 16 Initializes particles using two arrays (one for positions, one for velocities).

Lines 18 to 22 Plot settings

Line 24 Defines $animate(x, v)$: compute the new x according to the velocity.

Line 33 Defines $algo(x, v)$: calls $animate(x, v)$ for each iteration.

Line 44 Calls $algo(x, v)$. The simulation stops when the user closes the plot.

4.2 2D-space : the torus

4.2.1 Results and illustrations

Here, we work with $\mathbf{x}$ in $\mathbb{T}^2$. This simulation provides a visual representation of the torus. This way, we can show the periodical loop by drawing it. Since we work in dimension 2, we can't draw phase space. However, we can still represent classical space. Imagine the initial figure :

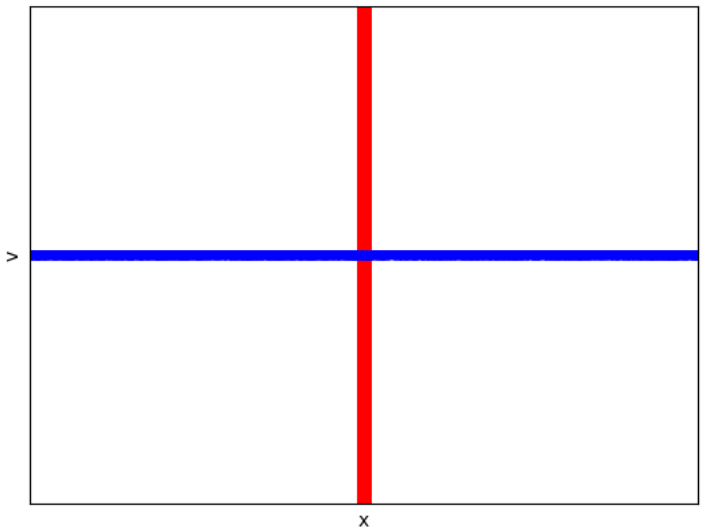


Figure 4.6

Then, we roll up the figure by linking its top and bottom edges. That shapes a tube :

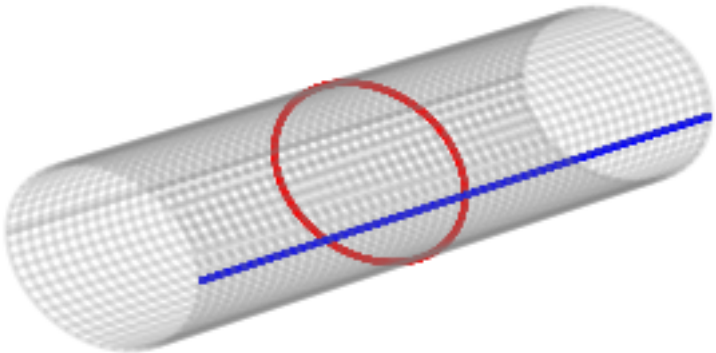


Figure 4.7

Now we link borders with each other and stretch the tube to shape a torus :

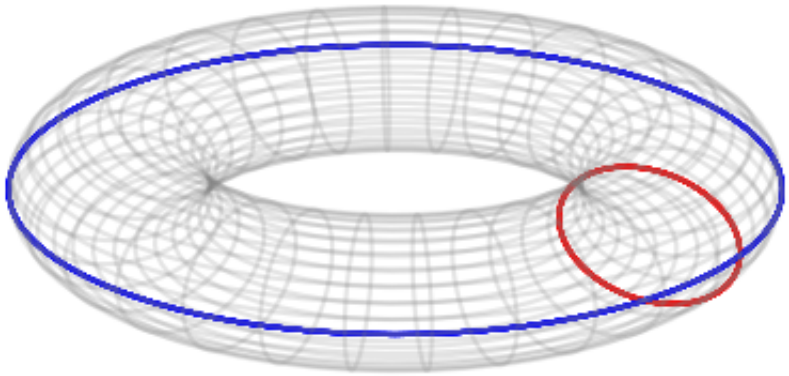


Figure 4.8

This way, periodicity translates into the fact that particles are moving on the torus surface. The period is represented by a complete circuit around the torus (along the red circle for y-axis, and along the blue circle for x-axis).

If we run the same simulation as before, initial configuration looks as follows :

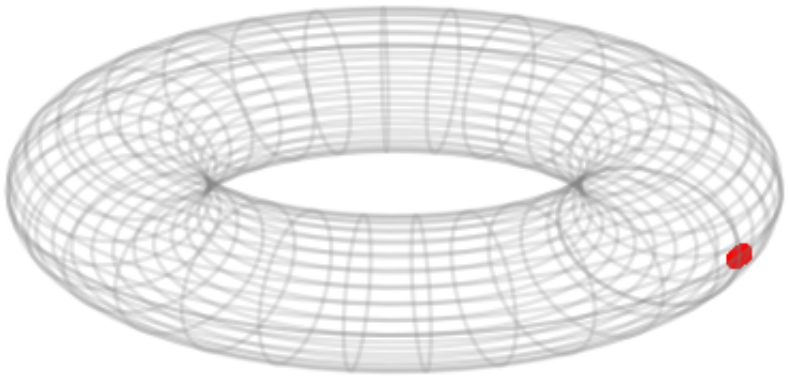


Figure 4.9: Initial configuration.
Particles are represented by red dots. $N = 10^4$.

In a view to simplify figures, parameters are not the same as before. Minor and major radius are respectively set to 7 and 20. For the same reason, maximum velocity is set to 20. Initial position bounds are larger as well. **These values are only for a better visualisation**. They are arbitrary so the simulation is easier to run.

As time goes on, it evoles as follows :

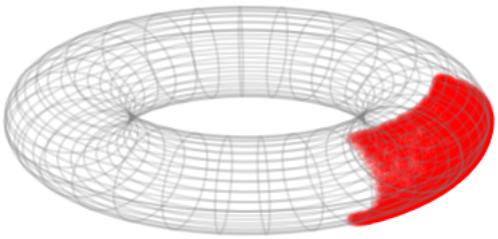


Figure 4.10: $t = 0.5$

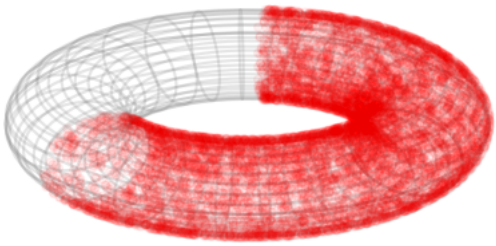


Figure 4.11: $t = 2$

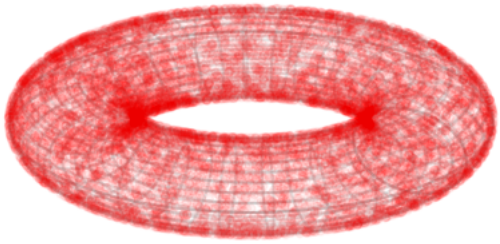


Figure 4.12: $t = 10$

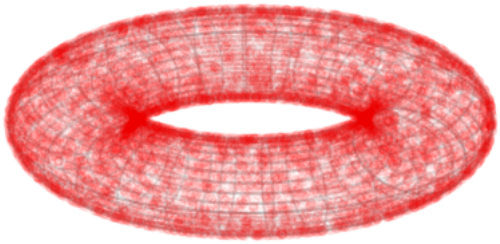


Figure 4.13: $t = 100$

Particles seem to occupy the torus homogeneously. One must keep in mind that **particles are travelling on the surface of the torus, not inside**. This simulation is the equivalent as the previous one, but in two dimensions. We could have run the same program as before, using a classical plan to display particle positions and applying periodicity on both axis. This simply provides another way to see it.

4.2.2 Code details

Here is provided the code for the torus simulation. Every pictures in the previous subsection were generated with this code. As it appears bellow, this codes generates figures 4.9 to 4.13, according to the value of t. It takes only few changes to generate figures 4.7 and 4.8.

```
1  import numpy as np
2  from mpl_toolkits.mplot3d import Axes3D
3  import matplotlib.pyplot as plt
4  import random as rd
5
6  #Parameters : N (nb of particles), r & R (minor & major radius), V velocity max.
7  N = 10000
8  r, R = 7, 20
9  V =20
10
11 # Generate torus mesh
12 angle = np.linspace(0, 2*np.pi, 60)
13 theta, phi = np.meshgrid(angle, angle)
14 X = (R + r * np.cos(phi)) * np.cos(theta)
15 Y = (R + r * np.cos(phi)) * np.sin(theta)
16 Z = r * np.sin(phi)
17
18 #Initializing speeds and positions
19 V1 = ((np.random.rand(N) -0.5)*2*V)
20 V2 = ((np.random.rand(N) -0.5)*2*V)
21
22 thetaInit = ((np.random.rand(N) -0.5)*2*np.pi/128)
23 phiInit = ((np.random.rand(N) -0.5)*2*np.pi/128)
24
25 # Choses time
26 t=10
27
28 #Moves particles according to t
29 theta2 = thetaInit + t*((V1/R))
30 phi2 = phiInit + t*((V2/r))
31 X2 = (R + r * np.cos(phi2)) * np.cos(theta2)
32 Y2 = (R + r * np.cos(phi2)) * np.sin(theta2)
33 Z2 = r *  np.sin(phi2)
34
35 # Display the mesh
36 fig = plt.figure()
37 ax = fig.gca(projection = '3d')
38 ax.set_xlim3d(-R, R)
39 ax.set_ylim3d(-R, R)
40 ax.set_zlim3d(-R, R)
41 ax.set_xlabel('x')
42 ax.set_ylabel('y')
43 ax.set_zlabel('z')
44 plt.axis("off") #set to "on" if you want the axis and grid to appear
45 ax.scatter(X2,Y2,Z2,'.', color="red", alpha=0.05)
46 ax.plot_wireframe(X, Y, Z, color="grey", alpha=0.2)
47 plt.show()
```

Line 7 to 9 Choice of N, torus dimensions and maximum velocity.

Lines 12 to 16 Generates the mesh for the torus, using its parametric equation.

Lines 19 to 24 Creates velocity lists, one for each component of $\mathbf{v}$.

Line 26 t is set.

Lines 29 to 33 Refresh particle positions according to t.

Lines 36 to the end are plot settings.

4.3 Study the convergence of the density

This section is linked with computations made in section 2 of chapter 2. Mathematically, we showed that ρ converges towards zero if $\mathbf{x}$ is in $\mathbb{R}^d$, but converges towards a constant (its average) if $\mathbf{x}$ is in $\mathbb{T}^d$ (Landau damping). Following simulations try to show this convergence.

Previous simulations treated cases for which $\mathbf{x}$ was in $\mathbb{T}^1$ and $\mathbb{T}^2$. Beyond $d = 3$, it is impossible to draw any space. However, we can compute the density in as much area as we want, regardless of the dimension.

As before, this program simulates the motion of particles in a torus. However, we focus on the **computation of density**. We recall that density is the number of particles that have a position $d\mathbf{x}$ around $\mathbf{x}$ at a given time t. To estimate its value, the program splits the space into a mesh of n^d cells (d is the dimension and n is an arbitrary integer chosen in the code). Then, it counts the number of particles in each cell. Let $\rho_s(t)$ be the number of particles in the cell s at time t (s is in $[\![1, \ldots, n^d]\!]$). If the density is supposed to converge towards its average $\bar{\rho}$ (constant in time), that means that for any s, $\rho_s(t)$ should converge towards N/n^d as time goes on.

Here, the convergence speed of $\rho_s(t)$ depends on velocity range and space bounds. Though, since particles are moving in constant speed, these parameters are not important : increasing velocity range is amount to the same thing than reducing space bounds. The trajectory of each particle is determined from the beginning. **This is why, as before, these parameters are arbitrary. For the whole section, position and velocity bounds are set to** ± 1.

However, we must chose a condition which determines either $\rho_s(t)$ has converged or not. That means we must chose a tolerance between the theoretical convergence value of ρ (that is N/n^d) and each actual value of $\rho_s(t)$ for any s in $[\![1, \ldots, n^d]\!]$. We will call this tolerance η. **Note that the algorithm stops when for any** s, $\rho_s(t)$ **equals** $N/n^d \pm \eta$**. It is our convergence criterion.** η **represents a number of particles**. Its value will be specified in each example.

The computation time to move particles and compute $\rho_s(t)$ at each iteration increases especially as d, N and n do. This being said, the code allows to run the simulation for any value of these. We will provide examples for $d = 2$ and $d = 3$, and fix $n = 10$. Higher values require a lot of time to compute.

4.3.1 Results in dimension 2

Following graphs show the density of each cell with respect to time. Each point on the graph matchs with a single cell, at a given time t (so for each value of t, there are n^d points). **We fix** $N = 10^5$.

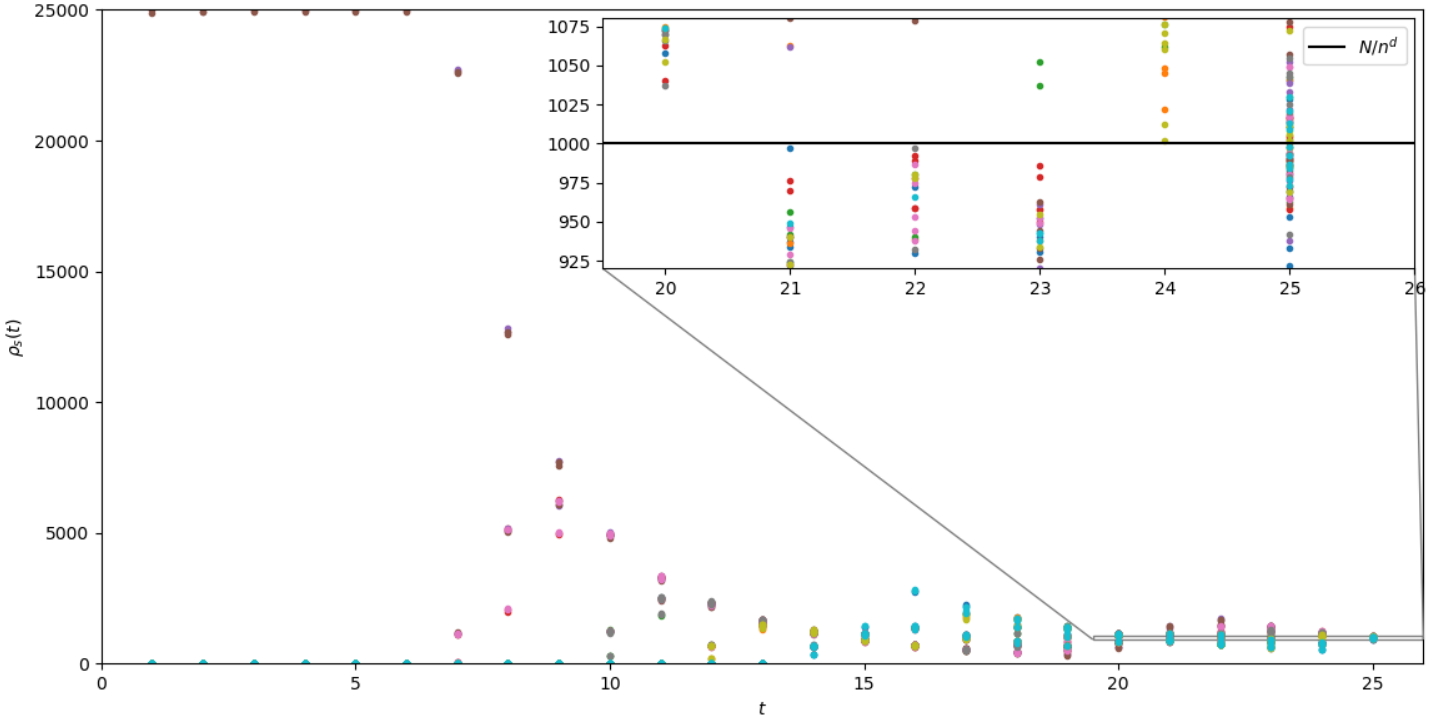


Figure 4.14: $N = 10^5$, $n = 10$, $d = 2$ (100 subspaces), $N/n^d = 1000$, $\eta = 80$.
Convergence criterion : $\rho_s(t) = 1000 \pm 0.08\%$.
Colors are only for differentiation purpose.

At the beginning, all cells are empty except those which contain particles at initial instants. This is why some points are at zero at first values of t (in fact most of them are at zero, but they overlap each others so they appear as a single point). In this configuration, it takes about 11 seconds for particles to be approximately equally distributed. Afterwards, we see that the number of particle in each cell seems to oscillate less and less. The zoomed-up part of the graph shows each value of $\rho_s(t)$ with more precision when it comes to converge. This being said, that does not provide any information of what happens after.

Let's see what happens with a smaller value of η.

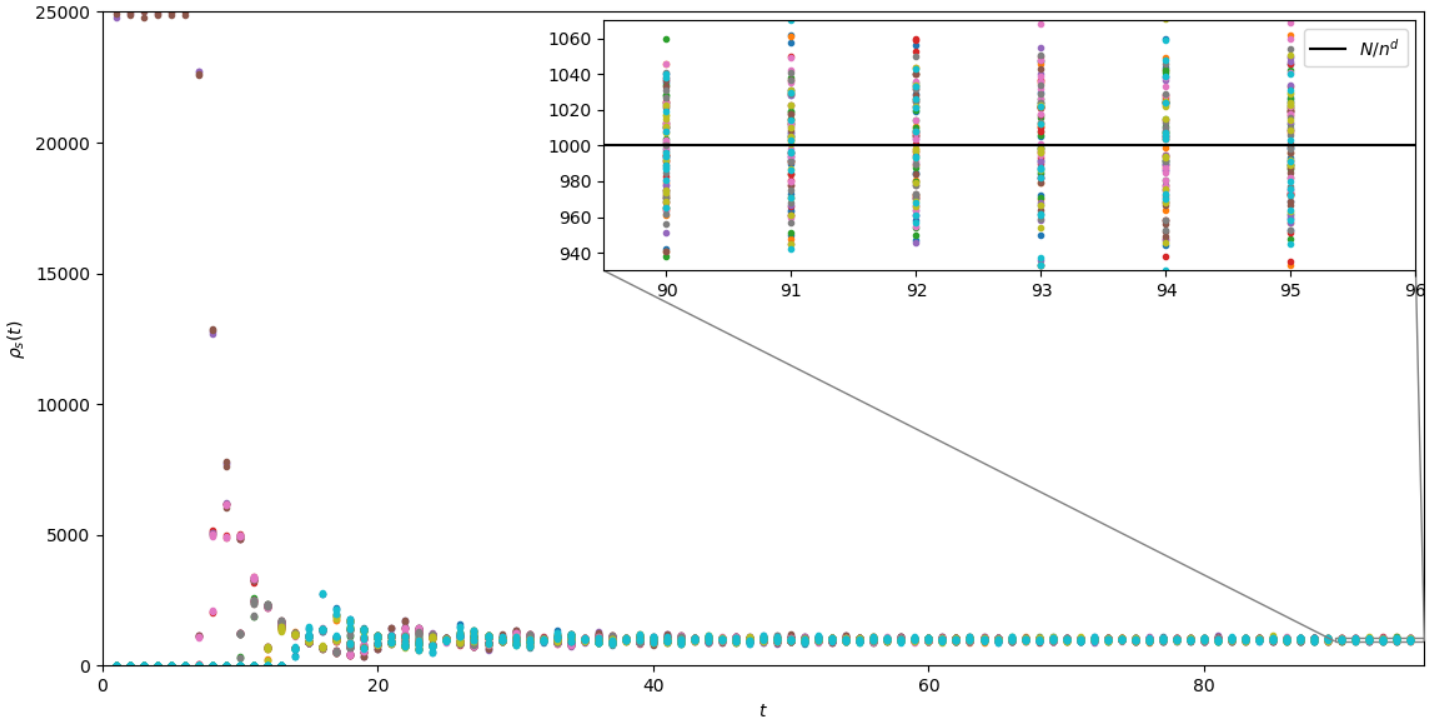


Figure 4.15: $N = 10^5$, $n = 10$, $d = 2$ (100 subspaces), $N/n^d = 1000$, $\eta = 70$.
Convergence criterion : $\rho_s(t) = 1000 \pm 0.07\%$.

Setting η to 70 instead of 80, we see that for the same initial configuration, it required 20 seconds for particles to be approximately equally distributed, and 96 seconds to reach convergence criterion. Results show that the more η decreases, the more time it takes to converge. When η is inferior to 50, it took too much time for me to wait for the convergence (even incrementing the time step). Though, the graph seems compatible with the convergence :

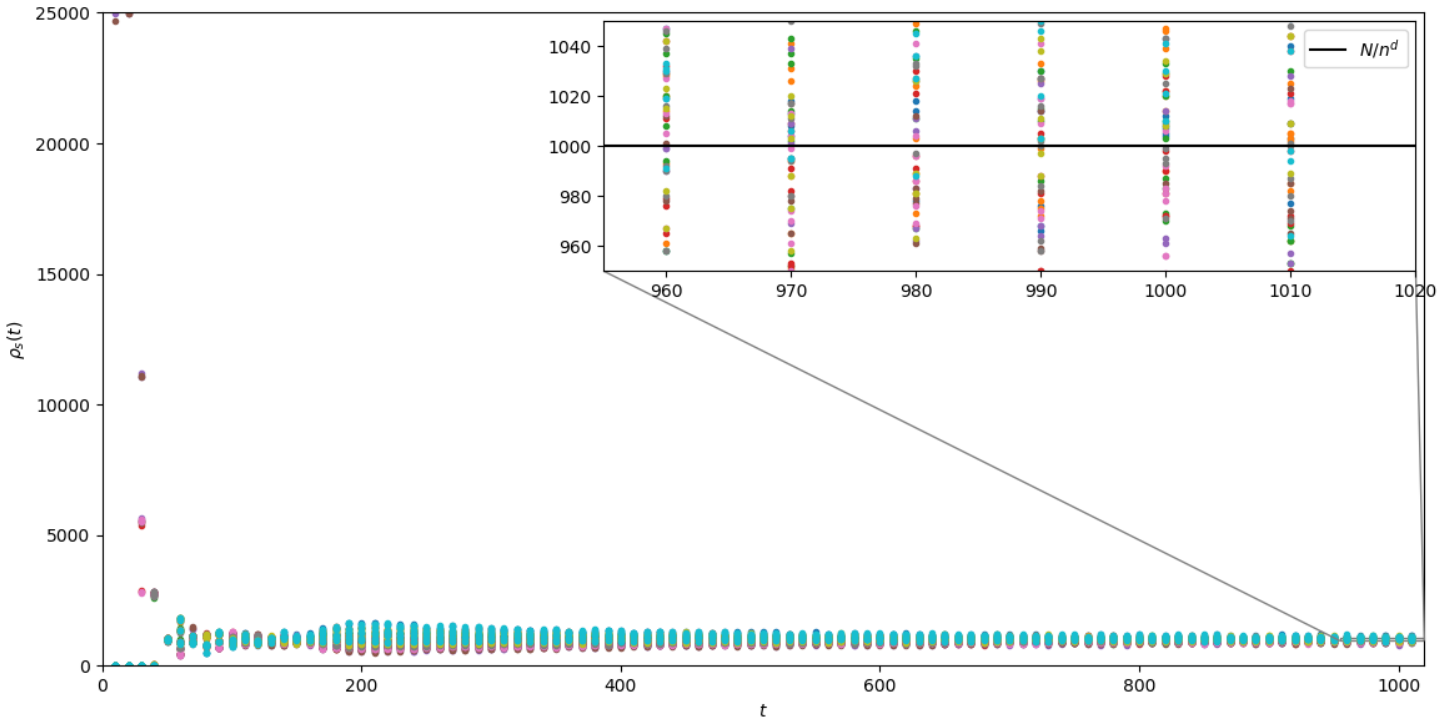


Figure 4.16: $n = 10$, $d = 2$ (100 subspaces), $\eta = 50$.
Convergence criterion : $\rho_s(t) = 1000 \pm 0.05\%$.
In this example, convergence criterion is not reached.

Though convergence criterion is not always reached (or at least, I could not wait long enough to reach it), density seems to oscillte less and less around N/n^d.

In following section, we will provide the same kind of graph in dimension 3.

4.3.2 Results in dimension 3

We use the same parameters as before, except for the number of particles. In dimension 3, the computation time increases a lot according to N. We can't reduce n since the convergence of ρ is meaningful only if we consider a lot of elementary volumes. To avoid too much computation time, **we will reduce N to 10^4. That means that our convergence criterion is that for any s, $\rho_s(t)$ must equal $10 \pm \eta$.**

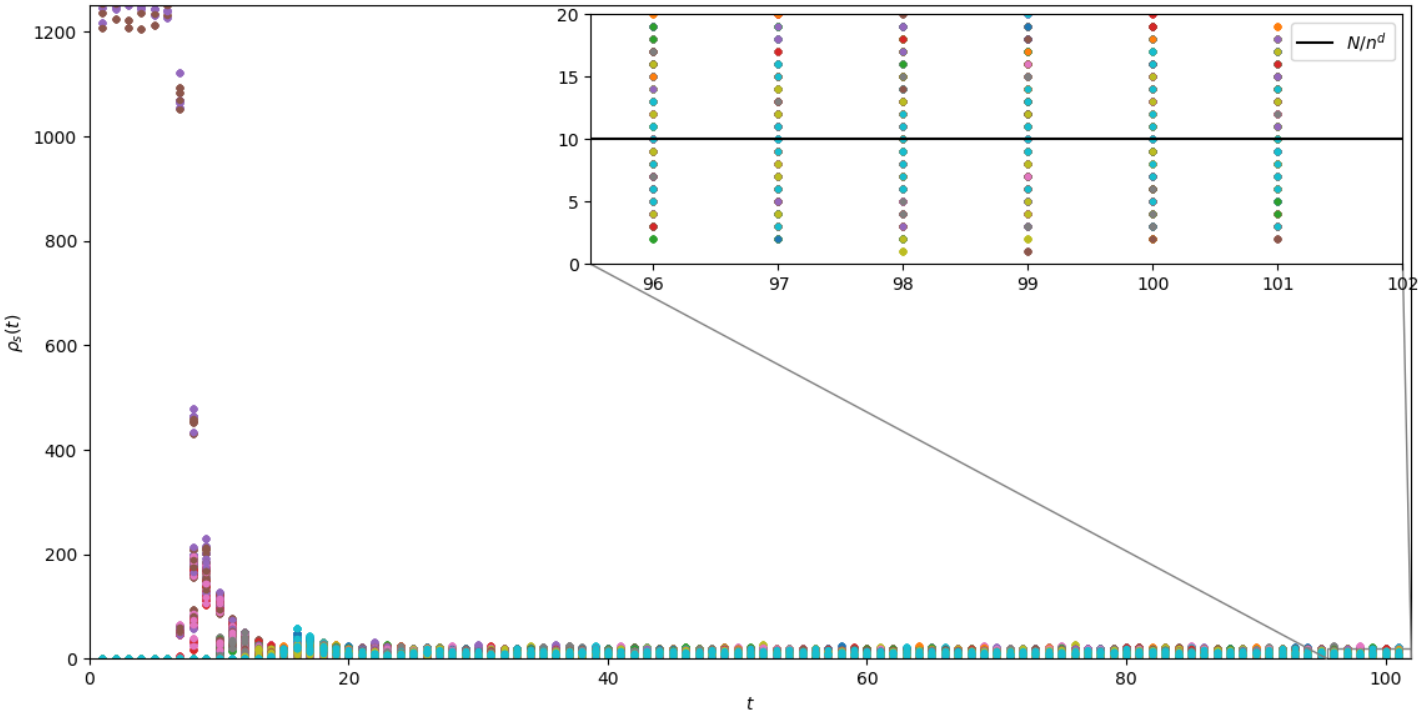


Figure 4.17: $N = 10^4,\ n = 10,\ d = 3$ (1000 subspaces), $\eta = 10$
Convergence criterion : $\rho_s(t) = 10 \pm 100\%$

.

The problem is that as η decreases, it becomes really unlikely to reach the convergence criterion **if N is not large enough**. The difference between η and N/n^d must be significant (of the order of n^d) for the convergence criterion to be relevant. But here, computation time prevented me from running the algorithm with higher values of N. **Again, this simulation is only a way to give a glimpse of the topic**. Nonetheless, if we run the algorithm with a greater time step, every value of ρ_s seem to oscillates around N/n^d :

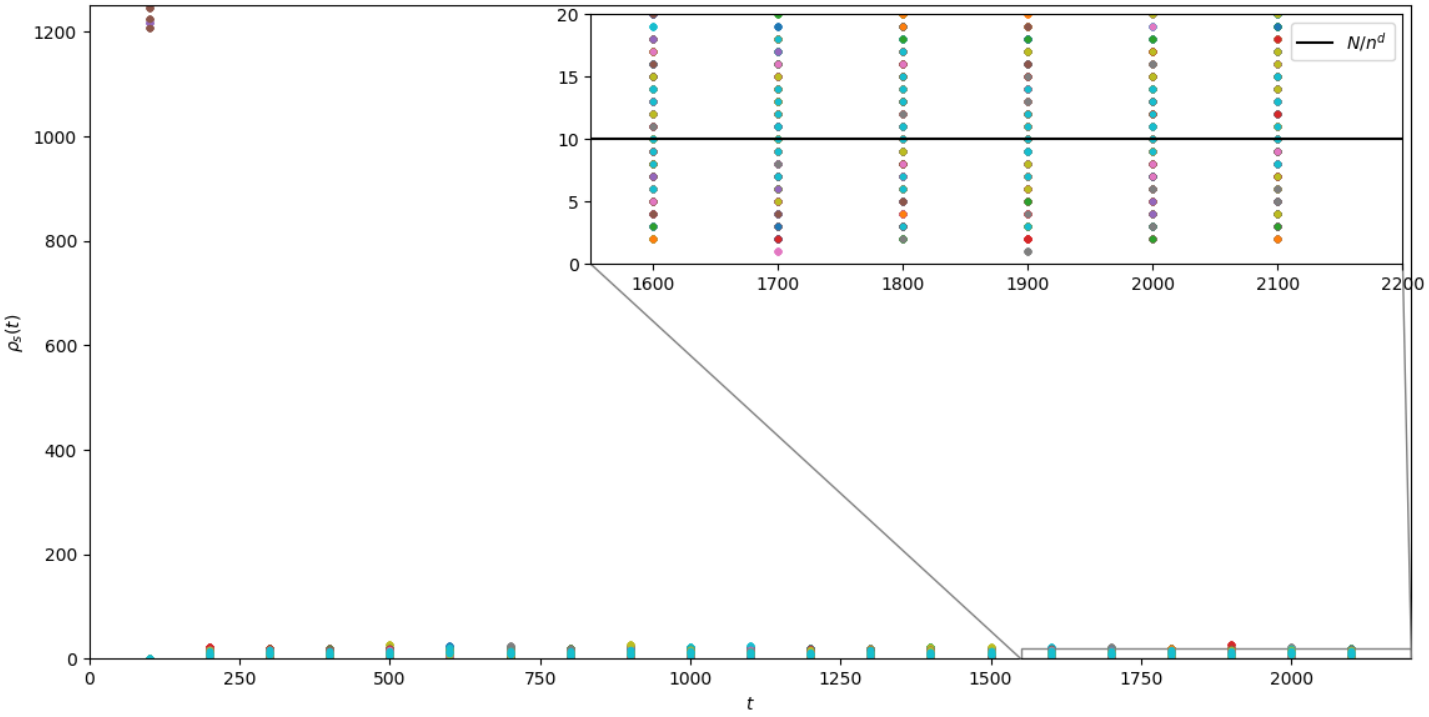


Figure 4.18: $N = 10^4,\ n = 10,\ d = 3$ (1000 subspaces), $\eta = 8$.
Convergence criterion : $\rho_s(t) = 10 \pm 80\%$
In this example, the convergence criterion is not reached.

.

Even if the convergence criterion is not always reached according to parameters we set, graphs show that the density indeed seems to oscillate around a constant value.

4.3.3 Code details

The code is separated in two files : $rhoMain.py$ and $rhoAlgo.py$.

- $rhoMain.py$ contains the main program, with the choice of parameters, the main loop which runs the simulation, and plot settings.
- $rhoAlgo.py$ contains the definition of two functions : $move()$ and $fillrho()$ which respectively defines how to refresh particles positions and how to compute every value of $\rho_s(t)$.

Here are provided codes and some explanations about them.

rhoMain.py

```
import numpy as np
import matplotlib.pyplot as plt
import random as rd
from mpl_toolkits.axes_grid1.inset_locator import zoomed_inset_axes
from mpl_toolkits.axes_grid1.inset_locator import inset_axes
from mpl_toolkits.axes_grid1.inset_locator import mark_inset
import rhoAlgo

# Parameters : N particles, n^d subspaces, eta and N/n^d = rhoth
N = 10**4
d = 3
n = 10
rhoth = N/(n**d)
eta = 0.1*rhoth

# Defines space and velocity bounds
xmax = 1
xmin = -xmax
vmax = 1
vmin = -vmax
step = (2*xmax)/n

# Initializes positions (X), velocities (V), density array (rho), and error array (err)
X = np.array([0.01*xmax*2*(np.random.rand(d)-0.5) for i in range(N)])
V = np.array([0.01*vmax*2*(np.random.rand(d)-0.5) for i in range(N)])
rho = np.zeros((n,)*d)      # creates a n-sized d-dimension array
err = np.array([abs(X-rhoth) for X in rho])

# Initializes time, and time arrays (for final plot)
t = 0
tstep = 1
Time = np.zeros((n,)*d)
tlim = 2000

# Zoomed-up plot and basic plot settings
fig, ax = plt.subplots()
axins = inset_axes(ax, width=6.5, height=2, loc=1)

# Moves particles until rho is close enough to its theoretical value
print("Running...")
if d<=2:
  while ((np.any(err >= eta)) and (t <= tlim)):
    print(t)        # allows to see the program progression
    X = rhoAlgo.move(X,V,N,t,d,xmax,xmin)
    rho = rhoAlgo.fillrho(X,N,d,n,xmin,step)
    err = np.array([abs(i-rhoth) for i in rho])
    Time+=tstep
    ax.plot(Time,rho,'.')
    axins.plot(Time,rho,'.')
    t += tstep
else:
  while ((np.any(err >= eta)) and (t <= tlim)):
    print(t)        # allows to see the program progression
    X = rhoAlgo.move(X,V,N,t,d,xmax,xmin)
    rho = rhoAlgo.fillrho(X,N,d,n,xmin,step)
    err = np.array([abs(i-rhoth) for i in rho])
    Time += tstep
    for i in Time:
      for j in rho:
        ax.plot(i,j,'.')
        axins.plot(i,j,'.')
    t += tstep

# General plot settings
mark_inset(ax, axins, loc1=4, loc2=3, fc="none", ec="0.5")
x1, x2, y1, y2 = t-(11*tstep/2), t+tstep, rhoth-eta, rhoth+eta
rhoth2 = plt.axhline(y=rhoth, color='black', linestyle='-', label=r'$N/n^d$')
axins.legend(handles=[rhoth2])
axins.set_xlim(x1, x2)
axins.set_ylim(y1, y2)
ax.set_xlabel(r'$t$')
ax.set_ylabel(r'$\rho_s(t)$')
ax.set_ylim(0, N/(2**d))
ax.set_xlim(0, t+tstep)
plt.show()
```

Lines 10 to 14 set every parameters : N, n, d, η (noted as eta) and N/n^d (noted as $rhoth$).

Lines 17 to 21 set periodicity parameters : position and velocity bounds.

Lines 24 to 27 initialize position and velocity arrays, density array (rho, for which each elements is a single subspace density), error array (err, for which each element is the difference between every densities and $rhoth$. This array allows to define our convergence criterion).

Lines 30 to 34 initialize time t, time step $tstep$ (determines each iteration of the simulation), time array $Time$ (to display rho with respect to time), and $tlim$ (simulation shutoff parameter).

Lines 40 to 62 run the simulation as long as the convergence criterion (or the shutoff parameter) is not reached. Positions and densities are refreshed at each iteration thanks to $move()$ and $fillrho()$ functions (detailed in next code).

All other lines treat plot settings and packages import.

Now, let's detail the code for *rhoAlgo.py*.

rhoMain.py

```
1  import numpy as np
2  from math import *
3
4  def move(X,V,N,t,d,xmax,xmin):
5    X+=t*V
6    for k in range(N):
7      for l in range(d):
8        if X[k][l] >= xmax:
9          X[k][l] = xmin+fmod(X[k][l],xmax)
10       elif X[k][l] <= xmin:
11         X[k][l] = xmax+fmod(X[k][l],xmax)
12   return X
13
14 def fillrho(X,N,d,n,xmin,step):
15   rho = np.zeros((n,)*d)
16   for k in range(N):
17     coord = [0 for j in range(d)]
18     for l in range(d):
19       j=0
20       while ((X[k][l]>xmin+(j+1)*step) and (j<n)):
21         j+=1
22       coord[l]+=j
23     rho[tuple(coord)]+=1
24   return rho
```

Function $move(X, V, N, t, d, xmax, xmin)$ uses velocity array V and a given time t to refresh position array X. The dimension of X and V depends on d. Plus, it must takes in count periodical bounds, which is why $xmax$ and $xmin$ are used as arguments as well.

The loop is simple : for every particles, each position component is changed according to t and its matching component of V. The periodicity is respected using a *modulo* on positions that drive particles back to the lower bound once they go beyond the upper bound, and vice versa. The output of the function is a new position array X.

Function $fillrho(rho, X, N, d, n, xmin, step)$ has been a lot harder for me to establish. I had a lot of difficulty working out how to fill rho correctly for any dimensions. The code works as follows :

1. The array rho is created. It has a size n and a dimension d (it thus contains n^d elements).
2. The fact that we have n^d cells comes from the fact that each axis of $\mathbb{R}^d$ are regularly splitted n times (n interval per axis thus result into n^d cells). Therefore, the idea is to check in which interval each component of the position of the particle is. This is what the loop (line 18) does. It results in a d-tuple (because position has d components), which correspond to the index of the matching element in rho. We thus add 1 to this element (line 23), because that means there is one particle in this cell.
3. We repeat the previous loop for every particles.

In the end, rho is filled up so each of its coefficient is the number of particles in one single cell. The output of the function is the new density array rho.

Conclusion

I have taken many things from this experience. From a global point of view, the topic was very interesting. I have been missing physics since I stopped studying it, so throwing myself into it again has been really pleasant for me.

At first sight the topic seemed very complicated to me. And according how deeply one would study it, it is indeed complicated. Many works and papers about it have been published, and some of them go back to decades. I was afraid to get lost quickly. Fortunately, my supervisors helped me a lot providing some guidelines about what to study specifically.

I was nicely surprised to work on some programmation. This is something I like, and it allowed me to be more at ease with Python, to learn more about graphs, and array computations.

However, since the topic was as wide as interesting, it was also a bit frustrating. Because each of its facet I presented in this report could have required a single dedicated internship. This implied I had to focus on very specific problems, which implied specific assumptions about their nature.

Moreover, this internship made me discover the world of research, and I feel a bit perplex by this. I worked a lot on searchers' papers, and had also some discussions about it with my tutor. What troubles me about it is that from one paper to another, according to the topic, different assumptions are made. But at the same time, loads of papers' proof, theorems, lemmas are based on what is written on other papers. But from a paper to another, some assumptions are not the same, and it felt like they these differences are not always taken into account.

The other point about it is that in most papers dealing with physics, only few try to explain things « in english »so that it makes sense in reality and not only through mathematics.

In the end, I enjoyed a lot this experience. First, the topic itself was really interesting, but not only. It brought me experience about reasearch in general, about how it works. And, whatever the opinion I have about it, that remains enriching.

Appendix

L^p spaces

Let $p \geq 1$. A $L^p(X)$ space is defined as the set of measurable function g on X, that is to say the set of function defined on X that satisfies

$$||g||_p = ||g||_{L^p} = \left(\int_X |g(x)|^p dx \right)^{1/p} < +\infty \tag{4.1}$$

where usually, $X = I \subseteq \mathbb{R}$. We say that two functions g and h are equal if they are equal *almost everywhere* (a.e.). Formally, that means

$$g = h \Leftrightarrow g(x) = f(x) \ \forall x \in X \mid \mu(X) \neq 0 \tag{4.2}$$

where μ is a measure on a measurable space.

$|| \cdot ||_p$ obeys the following rules : if X is a measurable set of non zero measure,

1. $\forall\ g \in L^p(X), ||g||_p = 0 \Leftrightarrow g = 0$ a.e. (Separation)
2. $\forall\ (\lambda, g) \in K \times L^p(X), ||\lambda g||_p = |\lambda|\ ||g||_p$. (Absolute homogeneity)

These two properties are obvious. But in order to show that $||\cdot||$ is a norm, we must prove that it satisfies triangle inequality : $\forall\ g, h \in L^p(X), ||g+h||_p \leq ||g||_p + ||h||_p$. For L^p spaces, it is called Minkowski's inequality. To prove it, we first prove Hölder's inequality.

Hölder's inequality. *Let X a measurable space. Let $p, q \leq 1$, which satisfy*

$$\frac{1}{p} + \frac{1}{q} = 1$$

and g, h two functions such as $g \in L^p(X)$ and $h \in L^q(X)$. Then we have the following statement :

$$||gh||_1 \leq ||g||_p \; ||h||_q \tag{4.3}$$

Proof. We assume that both $||g||_p$ and $||h||_q$ are non zero (in which case the proof is trivial).

We first show Young's inequality using the concavity of logarithm function : let $t = \frac{1}{p}$ and $1 - t = \frac{1}{q}$, then $\forall a, b > 0$,

$$\begin{aligned} ln(ta^p + (1-t)b^q) &\geq tln(a^p) + (1-t)ln(b^q) \\ &\geq ln(a) + ln(b) \\ &\geq ln(ab) \end{aligned}$$

which results into Young's inequality for product

$$\forall a, b > 0, \; p, q \mid \frac{1}{p} + \frac{1}{q} = 1, ab \leq \frac{a^p}{p} + \frac{b^q}{q} \tag{4.4}$$

Now one can apply this with $a = \frac{|g|}{||g||_p}$ and $b = \frac{|h|}{||h||_q}$. That gives :

$$\begin{aligned} \frac{|g|}{||g||_p}\frac{|h|}{||h||_q} &\leq \frac{1}{p}\frac{|g|^p}{||g||_p^p} + \frac{1}{q}\frac{|h|^q}{||h||_q^q} \\ \Leftrightarrow \frac{\int |gh| dx}{||g||_p ||h||_q} &\leq \frac{1}{p} + \frac{1}{q} = 1 \end{aligned}$$

hence the result

$$||gh||_1 \leq ||g||_p \; ||h||_q \tag{4.5}$$

■

Now we can show Minkowski' inequality using this result.

Minkowski's inequality. $\forall \; g, h \in L^p(X)$,

$$||g + h||_p \leq ||g||_p + ||h||_p$$

Proof. First, we note that

$$|g+h|^p \leq (|g|+|h|)\,|g+h|^{p-1} \tag{4.6}$$

Then, we compute

$$\begin{aligned}
||g+h||_p^p &= \int |g+h|^p dx \\
&\leq \int (|g|+|h|)\,|g+h|^{p-1} dx \\
&\leq \int |g||g+h|^{p-1} dx + \int |h||g+h|^{p-1} dx \\
&\leq \left[\left(\int |g|^p dx\right)^{1/p} + \left(\int |h|^p dx\right)^{1/p}\right] \left(\int |g+h|^{(p-1)\frac{p}{p-1}} dx\right)^{1-1/p} \\
&\leq (||g||_p + ||h||_p)\frac{||g+h||_p^p}{||g+h||_p}
\end{aligned}$$

hence the result

$$||g+h||_p \leq ||g||_p + ||h||_p \tag{4.7}$$

■

Therefore, $||\cdot||$ is a norm on L^p.

Riesz-Fischer's Theorem. *L^p is complete. That means every Cauchy sequences converge. We recall that a Cauchy sequence (x_n) is a sequence that satisfy*

$$\forall\ \epsilon > 0,\ \exists N \in \mathbb{N},\ \forall m,n \geq N,\ |x_m - x_n| < \epsilon \tag{4.8}$$

Proof. Let (g_n) a Cauchy sequence in L^p. It has a subsequence (h_n) that satisfies

$$\forall\ n \in \mathbb{N}, ||h_{n+1} - h_n||_p < \frac{1}{2^n} \tag{4.9}$$

To prove that (g_n) converges, we only need to prove that (h_n) does.
Let $h = \sum_n |h_{n+1} - h_n|$. As h is measurable, Minkowski's inequality allows to say that

$$||h||_p \leq \sum_n ||h_{n+1} - h_n||_p < \infty \tag{4.10}$$

which means that the serie $\sum_n (h_{n+1} - h_n)(x)$ converges absolutely (which implies the convergence). Hence h_n converges pointwise to a certain measuable function g, which satisfies

$$||g - h_n||_p \leq \sum_{k \leq n} ||h_{k+1} - h_k||_p \leq \frac{1}{2^{n-1}} \tag{4.11}$$

and therefore, is in L^p. ■

When $p = \infty$: For a given function g on X, we define its norm as follows :

$$||g||_\infty = ||g||_{L^\infty} = \inf\{C \geq 0 : |g(x)| \leq C \text{ a.e.}\} \tag{4.12}$$

where C is called the essential supremum of g. This can be written as

$$||g||_\infty = \sup \ \text{ess}|g| \tag{4.13}$$

Fourier series

For functions in $\mathbb{R}^d$: Any suitably regular complexe-valued function g defined on $\mathbb{R}$ can be defined using an integral representation as follows :

$$g(\mathbf{x}) = \int_{s=-\infty}^{+\infty} F(s) e^{2\pi i s x} ds \tag{4.14}$$

where s is a complex number. F is also a complex-valued function defined on $\mathbb{R}$. The function can be determined using the formula

$$F(s) = \int_{x=-\infty}^{+\infty} f(x) e^{2\pi i s x} dx \tag{4.15}$$

For functions in $\mathbb{T}^d$: We first recall the definition of a periodical function. Let $g : \mathbb{R} \to \mathbb{R}$. g is τ-periodical if and only if

$$\forall\ x \in \mathbb{R},\ g(x + \tau) = g(x) \tag{4.16}$$

The idea that lies behind Fourier series is to define a τ-periodical function as a sum of sinusoidal functions. That is, for a given τ-periodical function g,

$$g(x) = \sum_{k=-\infty}^{+\infty} \hat{g}_k e^{2\pi i \frac{k}{\tau} x} \tag{4.17}$$

where $\forall\ k \in \mathbb{Z}$, $\hat{g}_k$ are called Fourier coefficients of g, and are defined by

$$\hat{g}_k = \frac{1}{\tau} \int_{-\tau/2}^{+\tau/2} g(t) e^{-2\pi i \frac{k}{\tau} t} dt \tag{4.18}$$

Laplace transform

For a given function of $g(t)$ defined for all $t \geq 0$, its Laplace transform $\mathcal{F}(s)$ is

$$\mathcal{F}(s) = \int_0^\infty f(t)e^{-st}dt \tag{4.19}$$

where $s = a + ib$, $(a, b) \in \mathbb{R}^2$ is a complex number.The inverse transformation is defined by

$$f(t) = \frac{1}{2\pi i} \lim_{T\to\infty} \int_{\sigma-iT}^{\sigma+iT} e^{st}F(s)ds \tag{4.20}$$

For (3.6) and (3.7) in chapter 3, we only use different notation. We replaced s by $-iw$, and adapted the integration contour in the inverse transformation in consequence. Moreover, we replaced T by ∞ and removed the lim.

Internship progression follow-up

Week 1	Meeting my supervisor and his collegue, giving me a general introduction to the topic.
Week 2	Working on my own to learn some basic concepts about the topic and getting more familiar with it. My supervisor and his collegue were busy with a workshop they organised during the two first weeks (I attended some conferences)..
Week 3	Starting to work on progammation for simulations about particles behavior (see Chapter 5).
Week 4	Keeping working on simulations and starting studying Vlasov-Poisson system
Week 5	Last works on simulations, with specifier objectives (about density, see Chapter 5 section 5.3). It took me a while to find a way to build up RHO array for higher dimensions. Going into Vlasov's equation in depth. Starting report writing.
Week 6	Working on computations about particles density when $\mathbf{x} \in \mathbb{R}^d$.
Week 7	Working on Landau Damping : computations. Details about Landau computations.
Week 8	Landau damping : interpretation of the phenomenon, researchs about tokamaks and their functioning.
Week 9	Working on Fourier series and Vlasov's equation for $\mathbf{x} \in \mathbb{T}^d$. Completing the report.
Week 10	Reedition of the report, brought of some corrections and changes
Week 11	Writing the appendix.
Week 12	End of the internship. Finishing the final version of the report. Packing stuffs, preparing to go home.

Bibliography

[1] C. Mouhot.
Mathematical Topics in Kinetic Theory

[2] C. Bardos, P. Degond.
Global existence for the Vlasov-Poisson equation in 3 spaces variables with small initial data, 1985.

[3] L. D. Landau, E. M. Lifshitz.
Courses of theoretical physics, Vol. 10, Physical Kinetics, Robert Maxwell, M.C, 1981.

[4] *http://www.nyu.edu/classes/tuckerman/stat.mech/lectures/lecture_2/node2.html*

[5] ITER Wikipedia article.
https://en.wikipedia.org/wiki/ITER

[6] Annual Report and Financial Statements 2016-2017.
http://www.cardiff.ac.uk/__data/assets/pdf_file/0011/1035011/CU_AnnualReviewAccounts2017web.pdf