ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA ENERGETICA

TESI DI LAUREA

in
Trasporto di Particelle e di Radiazione M

INVESTIGATION OF GUIDING-CENTER DYNAMICS IN TURBULENT PLASMAS

CANDIDATO
Filippo Arlotti

RELATORE:
Prof.Ing. Domiziano Mostacci

CORRELATORE/CORRELATORI
Prof. Guido Ciraolo
Prof. Cristel Chandre

Anno Accademico 2021/2022

Sessione IV

Abstract

The transport of particles inside thermonuclear magnetic fusion devices is of great interest for the exploitation of the latter technology. The greatest limit on confinement time is indeed associated with the transport of particles across the plane perpendicular to magnetic field lines, the experimental data reveal much larger values of diffusion coefficients with respect to the expected ones. A phenomenon considered one of the main ones responsible for this behaviour is turbulence-induced transport, the latter is associated with microinstabilities of the plasma but it can lead to great values of heat and particles fluxes. In this manuscript, a guiding centre model is exploited to study the motion of particles inside a turbulent electrostatic field in simplified slab geometry. It is showed how to obtain the gyrokinetic Hamiltonian expression at the second order in the small parameter (related to the amplitude of the fluctuations in the electrostatic potential) and the related governing equations for the coordinates in the transverse plane. It is then outlined how to obtain the same kind of governing equations for a full kinetic approach and some considerations on re-scaling factors are made. The numerical Python code is presented in its main part and peculiarities, and a brief description of its operation is also available. The numerical analysis of results is made in MATLAB and starts with verifying the proper functioning of a method that allows one to obtain the guiding centre trajectories from the full orbits of particles. A time-step analysis of the influence of the latter parameter over the quality of the numerical result is then presented for both the guiding centre and full orbits approach. The main analysis of results is focused on the investigation of the influence of physical parameters ρ and η, respectively *Larmor radius* and *real parameter*, on the dynamics of the system, together with the investigation of an unveiled anomalous ballistic behaviour of some particles for certain regimes of motion. The computation of rotational number with weighted Birkhoff average leading to really interesting results is included in this last analysis. A possible explanation for this super-diffusive regime through the existence of meandering (twistless) invariant tori is then provided. The last part of numerical analysis is focused on the comparison of results from the guiding centre (reduced) model with the ones from the simple integration of the equations of motion (full kinetic), the presence of the anomalous super-diffusive behaviour in a non-approximated and more reliable model would mean that the phenomenon may be present in turbulent plasmas and that it is taken into account by the considered gyrokinetic model.

Contents

Chapter 1

Introduction

In this chapter, a little overview of the main aspects concerning nuclear fusion, magnetic confinement, transport of particles inside nuclear fusion devices and possible numerical approaches to the plasma dynamics investigation will be presented, the chapter itself is mostly based on books and literature considered as a reference from the scientific community, in particular, it is based on [Fre08], [Wes04], [Hel02], [Sta07], [Lit80], [HSK00] and [Cha18].

1.1 Nuclear Fusion

1.1.1 Thermonuclear fusion

The growth of the world population together with the increase in the average quality of life of people are strictly related to a rise in energy demand and consumption. Exploiting nuclear reactions to produce large amounts of energy is particularly interesting, every reaction is associated with an enormous release of energy. This is due to the mass deficit between reactants and products, it is orders of magnitude larger than the one associated with common chemical reactions. Nuclear reactions can be classified into two main categories, fission and fusion. Both processes lead to the production of nuclei characterised by being better bounded. In order to better understand these phenomena, the concept of nuclear binding energy should be introduced. A single nucleus is characterised by a specific atomic mass $\mathcal{A}$ and it is made of a certain number of neutrons $\mathcal{N}$ and protons $\mathcal{Z}$, these are the so-called nucleons and they are strongly bounded to form the nucleus itself. The energy required to break the bonds between nucleons Δ can be calculated through the well-known Einstein formula $E = mc^2$, one can simply write $\Delta = \Delta mc^2$ where Δm is the above-mentioned mass defect, this time between nucleus and nucleons considered as free particles while c is the light speed in vacuum. It is then possible to calculate the average nucleon binding energy $\Delta/\mathcal{A}$, the larger the latter is, the more stable the corresponding nucleus is. Nuclear reactions that lead to products characterised by higher average nucleon binding energy are the exothermic ones, naturally, these are the ones in which we are interested. By looking at Fig. [1.1] it is easier to distinguish the two types of nuclear reactions we introduced previously, fission reactions move towards products

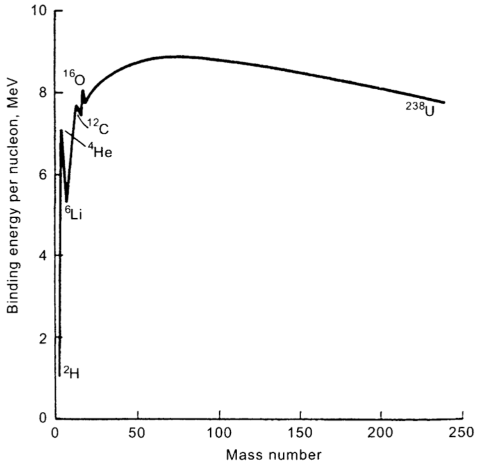


Figure 1.1: Average binding energy per nucleon $\Delta/\mathcal{A}$ as a function of atomic mass number. Extracted from [Sta07].

with a lower $\mathcal{A}$ while the ones towards heavier resulting products are the so-called fusion reactions. Both the types move the system to a state characterised by a higher $\Delta/\mathcal{A}$, the maximum value is obtained for Iron-56 (^{56}Fe), the latter is the most stable nucleus. This is the fundamental concept behind nuclear reactions for energy production purposes. Nuclear fusion is the energy source of stars.

1.1.2 Lawson criterion

Light nuclei interact with each other and fuse together into heavier and more stable ones. The main problem in achieving such reactions is the Coulomb repulsion: the interacting nuclei have to overcome the potential barrier in order to arrive close enough to each other, at that point attractive strong nuclear forces induce the fusion of the nuclei that are interacting. Particles must collide at very high kinetic energy to allow this kind of reaction, the incident energy has to be higher than the Coulomb barrier in principle. The required energies are well above the average electron binding energy, therefore the matter is fully ionized, and it reaches the state of plasma. It is necessary to obtain a sufficiently hot and dense plasma for a considerable time in order to allow fusion reactions to occur, this is the condition that has to be met. Thanks to a simple energy balance it is possible to obtain the so-called Lawson criterion [Wes04], this states that the triple product of temperature T, density n and confinement time τ_E needs to exceed a certain threshold in order to achieve a net energy production:

$$nT\tau_E \geq 10^{21}\ keVs/m^3 \quad \text{with} \quad T \gtrsim 20keV. \tag{1.1}$$

Two main approaches to the problem are currently under investigation. The first strategy is exploiting the so-called inertial confinement, in that case, the main goal is to produce a very high-density plasma, it should be in the order of $10^{30}m^{-3}$ so that the confinement time can be low, $\tau_E \approx 10^{-9}s$. This can be achieved by compressing fuel pellets with shock waves generated by a great number of powerful lasers or ions beam. In this field, the main contributions to the research come from the defence departments of some nations since hydrogen bombs are based on this operating principle. At Lawrence Livermore National Laboratory (LLNL) some major results have been obtained in one of the last experimental campaigns [Bis22], this is very important for this technology in recent times it didn't seem to be competitive for non-military applications. The alternative approach, magnetic confinement fusion, aims to produce a lower density plasma, $n \approx 10^{19} m^{-3}$, but with a greater confinement time, τ_E in the order of $1s$ to satisfy the constraints of Lawson criterion from Eq. [1.1]. This is the technology on which we will focus more for the purposes of my work and it will be presented and further analysed in the next section.

1.1.3 Controlled fusion reactions

Before moving on to the next section it is necessary at least to specify which are the reactions occurring in such applications, we anticipated that we are trying to replicate what happens inside stars, in reality, it is not possible to exploit the same reactions that power the latter. For instance, the proton-proton chain on which energy production in the Sun is based, involving the decay of protons into neutrons, is too slow to be exploited on Earth. The most promising reaction for terrestrial applications is the deuterium-tritium reaction, they are both hydrogen isotopes and their interaction leads to the production of a nucleus of Helium and a neutron other than a massive quantity of energy:

$$ {}_1\mathrm{D}^2 + {}_1\mathrm{T}^3 \quad \Rightarrow \quad {}_2\mathrm{He}^4 + \mathrm{n} + 17.6 MeV. \tag{1.2}$$

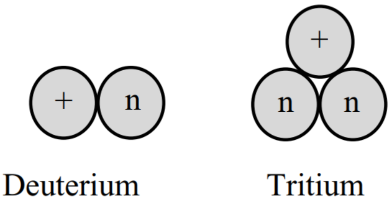


Figure 1.2: Nuclear structure of the basic fusion fuels. Extracted from [Fre08].

The energy of $17MeV$ is released in the form of kinetic energy and redistributed between the products accordingly to their mass ratio. The choice of focusing on this reaction is due to the fact that it is the easiest to initiate, despite the fact that currently there are a lot of problems. The latter is associated with the fact that this reaction produces a large number of neutrons and that it requires a supply of tritium in order to operate continuously, tritium is radioactive with a half-life of 12.26 years, as we speak there is no tritium on the Earth. Having made the choice of using this reaction, one has to deal with the problems related to neutrons and to the lack of fuel. Since nuclear fission is well known and exploited, nuclear engineers, know how to deal with activated materials as well as radiation damage caused by high-energy neutrons, the solutions are complicated but already established, despite the fact that usually in fission one has to deal with neutron with the energy of $2MeV$ while for the concerning applications, neutrons will reach $14MeV$, it is a problem but it seems it can be solved. The most concerning problem is the tritium supply, nowadays the designed solution is the so-called Lithium breeding blanket surrounding regions where D-T reactions take place. The chemical element that is most favourable for breeding tritium is lithium:

$$ {}_3\mathrm{Li}^6 + \mathrm{n}\ (\mathrm{slow}) \quad \Rightarrow \quad {}_2^4\mathrm{He} + \mathrm{T} + 4.8 MeV. \tag{1.3}$$

$$ {}_3\mathrm{Li}^7 + \mathrm{n}\ (\mathrm{slow}) \quad \Rightarrow \quad {}_2^4\mathrm{He} + \mathrm{T} + \mathrm{n} - 2.5 MeV. \tag{1.4}$$

Also in this case the isotopes are present in different proportions, ${}_3\mathrm{Li}^7$ is the 92.6% while ${}_3\mathrm{Li}^6$ is only the 7.4%, also, in this case, nuclear data show that ${}_3\mathrm{Li}^6$ reaction is much easier to be initiated and that it dominates the breeding. This system hasn't been tested and the scientific community still doesn't know if it will work or not, definitive answers will be given with the operation of DEMO (DEMOnstration power plant), it will be ITER's successor.

1.2 Magnetic confinement

It is now time to introduce the second approach to the problem of generating a plasma that satisfies the Lawson criterion. It is the most promising one and it is the one on which research has focused most, from the engineering point of view many more solutions to problems have been developed and designed for this configuration. In order to reach the required energies for fusion reactions, it is necessary to heat a mixture of deuterium and tritium gas to a sufficiently high temperature, in the order of 100 - 200 million K, so that the average thermal energy is in the order of 10-20 keV (not sufficient for Lawson criterion) but that a considerable number of particles in the tail of the corresponding Maxwellian distribution (for both the species) have energies exceeding the Coulomb barrier allowing fusion reactions to occur. This gas at very high temperatures must be isolated, it is necessary to avoid any contact with the walls surrounding the plasma. The rapid heat transfer and the associated heat fluxes are likely to result in structural damages as well as in an instantaneous cooling of the plasma, fusion reactions cannot occur in this case. Since the plasma is made of charged particles and charges can interact with any electromagnetic field $\mathbf{E}$ and $\mathbf{B}$, the resulting motion is determined by the Lorentz force:

$$ m\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}). \tag{1.5}$$

Magnetic confinement exploits this principle in order to confine particles inside the desired volume, in reality, the operation of a device exploiting this kind of principle is very difficult and the motion of confined particles is still under analysis.

1.2.1 Main drifts mechanisms

It is possible to introduce a little about how the particles move inside the above-mentioned electromagnetic fields. Starting with the most simple case, we consider the presence only of the magnetic field $\mathbf{B}$, in this case, the particles are free to move along the field lines but they are constrained to gyrate around them in the transverse plane (perpendicular direction). If $\mathbf{B}$ is uniform and constant, the resulting particles trajectories will be helix

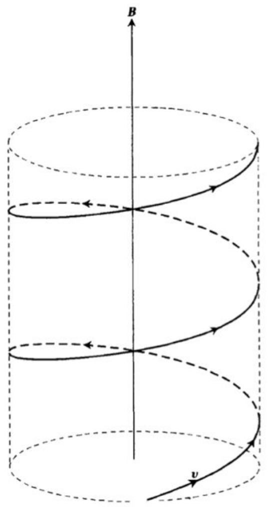


Figure 1.3: Gyromotion of a positively charged particle, extracted from [Den90].

around the magnetic field lines, the resulting motion is known as *gyromotion* and it is characterised by a couple of parameters, the *gyrofrequency* Ω and the so-called *Larmor radius* ρ, their expression is respectively the following:

$$\Omega = \frac{qB}{m} \quad ; \quad \rho = \frac{v_\perp}{\Omega}. \tag{1.6}$$

Since the dynamics of particles in the presence of only **E** is well known, we switch directly to the case of the presence of both **B** and **E**. When they are present at the same time, particles are not constrained to follow the same magnetic field line, they move in a direction that is perpendicular to both fields, and the resulting motion is the so-called **E** × **B** drift, the resulting velocity of particles associated with this kind of drift is the following:

$$\mathbf{v}_{\mathbf{E}\times\mathbf{B}} = \frac{\mathbf{E}\times\mathbf{B}}{|\mathbf{B}|^2}. \tag{1.7}$$

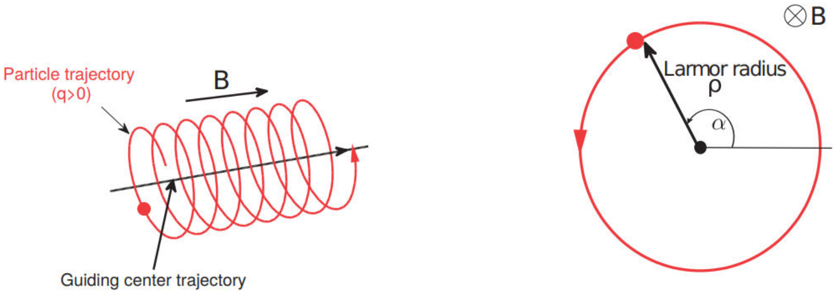


Figure 1.4: Sketch of the gyromotion of a positively charged particle in uniform magnetic field **B** together with its projection on the transverse plane, from [Mer16].

As anticipated the direction of the resulting velocity is determined by the cross-product between the two electromagnetic fields, one has also to notice that this kind of motion is independent of charge and mass, electrons and ions ideally follow the same path without creating any current in the plasma. Things get tough when considering a more realistic case, in a device for magnetic confinement **B** is not homogeneous and has a certain curvature, this leads to the generation of two additional drifts in the dynamics of the particles, we have respectively:

$$\mathbf{v}_{\nabla \mathbf{B}} = \frac{mv_\perp^2}{2q}\frac{\nabla \mathbf{B} \times \mathbf{B}}{B^3} \quad ; \quad \mathbf{v}_\mathrm{c} = \frac{mv_\parallel^2}{qB}(\nabla \times \mathbf{b})_\perp . \tag{1.8}$$

Where $v_\parallel$ indicates the velocity parallel to the magnetic field $\mathbf{B} = \mathrm{B}\mathbf{B}$. Nowadays, the most widely used design for fusion devices is the torus-shaped one, this is the simplest geometry that allows the magnetic field lines to be closed on themselves. Despite this, field lines cannot be some simple rings, this is due to the two additional drifts that we just introduced. Being charged dependent, $\mathbf{v}_{\nabla \mathbf{B}}$ and $\mathbf{v}_\mathrm{c}$ determine drifts in opposite directions for positively charged ions and electrons. This results in a vertical charge separation that produces an additional component to the electrical field and that leads to a radial expulsion of the plasma consequently to the action of $\mathbf{E} \times \mathbf{B}$ drift, the configuration would be intrinsically unstable. Thus, it is necessary to twist magnetic field lines into a helical shape around the torus, this leads to the cancellation of gradient and curvature drifts thanks to the fact that the motion of particles along the magnetic fields line is much faster than the one in the transverse direction allowing effective confinement.

1.2.2 Promising magnetic fusion concepts

The twist in the magnetic field lines is mostly obtained in two ways: by internal currents in tokamaks or by external coils in stellarators, they are the two most promising designs for magnetic fusion devices and a little schematic of their configuration is shown in Fig. [1.5]. The tokamak is characterised by a strong toroidal magnetic field generated by external coils and a poloidal magnetic field generated by the current flowing inside the torus itself. The current in the plasma is generated using a transformer, the plasma is the secondary winding, it is required an external current drive to power the machine. The induced current in the machine is used to heat the plasma thanks to Ohmic dissipation, the problem is that the resistivity of plasma decreases with temperature and additional heating systems are required. The main problem of the tokamak is that their operation is based on the principle of a transformer, it is not possible to operate in a steady state, and the latter is the desired regime for energy production. The stellarator instead doesn't need these strong currents in the plasma for the twisting in the magnetic field lines, these are obtained by a set of complex external coils, they are already positioned and oriented in order to obtain the desired magnetic field shape. This means that it doesn't need to use a transformer and that it can operate at a steady-state regime. The main problem of this configuration is the above-mentioned set of external coils, their positioning and operation are very complicated.

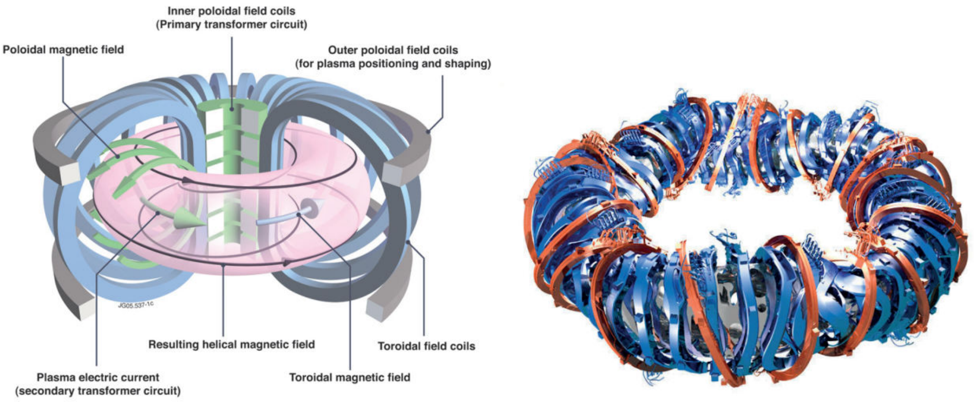

Inner poloidal field coils
(Primary transformer circuit)
Poloidal magnetic field
Outer poloidal field coils
(for plasma positioning and shaping)
JG05.537-1c
Resulting helical magnetic field
Toroidal field coils
Plasma electric current
(secondary transformer circuit)
Toroidal magnetic field

Figure 1.5: Schematic of basic configurations of the two most promising designs for fusion devices, the Tokamak on the left [EF23] and the Stellarator on the right [IPP23].

1.3 Transport in confined plasmas

Even if macroscopic instabilities are stabilized, there are still some problems in reaching and satisfying Lawson's criterion. One of the biggest problems is associated with the radial transport of particles and energy, this phenomenon leads to a dramatic decrease in the confinement time τ_E. Transport theory's main goal is then to investigate and control (if possible) such phenomena. Transport processes are usually considered of diffusive nature, the diffusion coefficient is defined through the usual formula:

$$D = \frac{a^2}{\tau_E} \tag{1.9}$$

We assume the validity of Fick's law so that the flux of particles can be expressed as:

$$\Gamma = -D\nabla n. \tag{1.10}$$

Where a is the distance that particles must travel before going out of the system in the designed confinement time τ_E while n is the density of particles.

1.3.1 *Classical* transport

In order to model these phenomena one should start from the most obvious cause for this kind of process, a collision between particles, despite the fact that, as we will see later, this is the mechanism that contributes less to the experimentally measured fluxes. Collisions between particles lead to the so-called *classical* transport, it is essentially due to particles moving from the magnetic field lines after having collided with each other [Hel02]. In order to describe this process, usually a random walk model in a magnetized plasma is used. The idea behind this is to show how a particle diffuses away from its initial position after a series of random collisions assuming that all motion is $2-\text{D}$ in the plane perpendicular to the magnetic field. Differently from what showed in Fig. [1.6] we

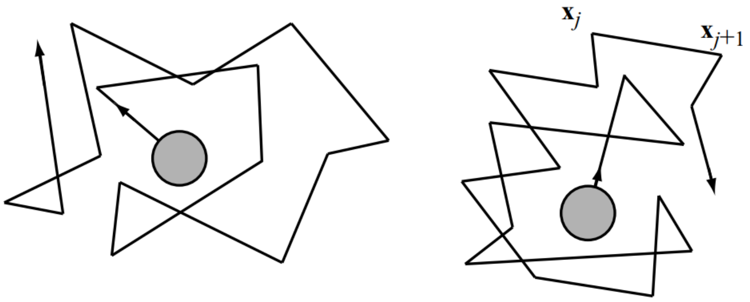


Figure 1.6: Trajectory of two particles undergoing random collisions. Note that on average $\Delta x = 0$ while $(\Delta x)^2 \neq 0$, extracted from [Fre08].

have also to remember that, while considering a magnetized plasma, the orbits between collisions are not straight lines, they are circular gyro orbits instead. This kind of model gives us the following expression of diffusion coefficient for *classical* transport:

$$D_{\text{class}} = \frac{\Delta x^2}{\Delta t} \approx \frac{\rho}{1/\nu_c}. \tag{1.11}$$

With ρ being the Larmor radius and ν_c being the Coulomb collisional frequency. For typical fusion-relevant conditions, one finds that the diffusion coefficient is approximately $D_{\text{class}} = 10^{-3} m^2/s$. The actual measured total diffusion coefficient is in the order of $D = 1m^2/s$, one can easily understand why it was anticipated that the *classical* transport is considered negligible if compared to other mechanisms.

1.3.2 *Neo-classical* transport

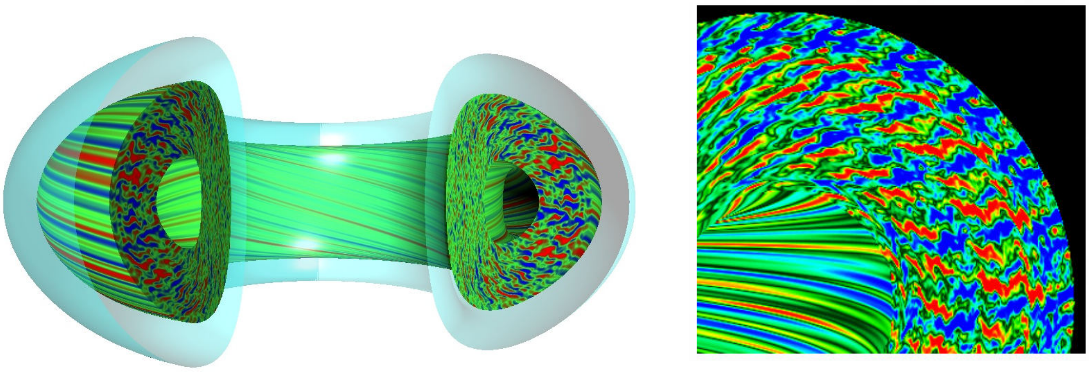
One further phenomenon associated with the additional radial motion of particles is the so-called *neo-classical* transport. It is due to the toroidal geometry of fusion devices, we know that there are drifts associated with the curvature of the magnetic field lines as well as with the presence of ∇B. The exploited geometry causes a modification of collisionless trajectories with a consequent extra movement of particles in the transverse plane, actually, the excursion with respect to the magnetic field lines exceeds ρ. The expression of the coefficients can still be calculated through the use of a modified random walk model accounting for the mentioned mechanism, it gives us the following expression of the particle diffusion coefficient:

$$D_{\text{neo}} = 5.2q^2 \Big(\frac{R_0}{r}\Big)^{\frac{3}{2}} D_{\text{class}}. \tag{1.12}$$

Where R_0 is the major radius of the machine and q is... . Note that for particular values of the parameters D_{neo} can be up to two orders of magnitude larger than the corresponding classical values. Neoclassical theory is considered a useful reference point for the lower limit on particle transport but it is still considered optimistic with respect to the operation of actual fusion devices, the experimentally measured diffusivities are still usually larger by one or two orders of magnitude despite the fact *neo-calssical* transport increase massively the concerning coefficients. One usually refers to the additional transport as to the *anomalous* one.

1.3.3 Turbulence-induced transport

At the moment, it is accepted to associate this *anomalous* transport to plasma turbulence, it consists of instabilities on microscopic scale lengths, and one usually refers to them as to microinstabilities. These phenomena are induced by the non-thermodynamic equilibrium steady state at which the plasma is maintained in order to meet conditions necessary to allow fusion reactions. It is present a strong pressure gradient between the core and the edge of the plasma, we have respectively $\approx 1 \cdot 10^5$ Pa in the first region while $1 \cdot 10^-6$ Pa in the second. This radial gradient across the plasma provides a source of free energy which especially destabilizes small-scale plasma waves, i.e. characteristic wavelengths perpendicular to the magnetic field lines of the order of ions and electron Larmor radius. These micro-instabilities involve density and temperature fluctuations δn and δT respectively but also electrostatic or electromagnetic fluctuations, $\delta \phi$ and δA. As anticipated, when the plasma is in this turbulent state, the fluctuations result in a radial transport of particles and energy. While considering only electrostatic fluctuations $\delta \mathbf{E}$ it is possible to calculate the related arising transport. It is also possible to attempt a theoretical description of the mechanism through the use of random walk estimates for turbulent transport arising from the fluctuating $\mathbf{E} \times \mathbf{B}$ drift velocity $\delta v_\perp$. Since the potential can modelled as a superimposition of various modes, one can describe fluctuations of the latter in terms of their Fourier components, the resulting electrostatic fluctuations can be expressed as:



(a) Turbulence inside a tokamak from numerical simulation.

(b) A detailed view of turbulence in a torus section.

Figure 1.7: Turbulence highlighted at different levels of detail, extracted from [LPP22].

$$\delta \phi = \sum_k \delta \phi_k e^{jk \cdot x}. \tag{1.13}$$

Each component of the fluctuations will arise a fluctuation in the $\mathbf{E} \times \mathbf{B}$ drift, the global effect on $\mathbf{v}_{\mathbf{E}\times\mathbf{B}}$ will be considered as a superimposition of various components as well:

$$\delta \mathbf{v}_k = -j \frac{k \times \mathbf{B}}{B^2} \delta\phi_k. \tag{1.14}$$

The fluctuation $\delta \mathbf{v}_k$ will persist for a certain time τ_k, the so-called *correlation time*, it will lead to a certain displacement in space: $\delta r_k \approx \delta \mathbf{v}_k \tau_k$. As anticipated, by summing over the wave-number spectrum, it is possible to obtain an estimate of the random walk for the turbulent diffusion:

$$D = \sum_k \frac{(\delta r_k)^2}{\tau_k} = \sum_k \Big(\frac{k_\perp \delta\phi_k}{B}\Big)^2 \tau_k. \tag{1.15}$$

The *correlation time* is determined by the process which most rapidly limits the unidirectional $\mathbf{E} \times \mathbf{B}$ drift. We consider that the processes involved can be:

- Time variation of the fluctuation:

$$\tau_k \approx 1/\omega_k.$$

- Time for a particle to move along a parallel wavelength of the fluctuations:

$$\tau_k \approx 1/k_\perp \nu_T.$$

- Time for magnetic drifts to carry particles over a perpendicular wavelength:

$$\tau_k \approx 1/\omega_k.$$

- Time for collisions to change the particle orbit:

$$\tau_k \approx 1/\nu_{eff}.$$

- The turbulent velocity $\delta \mathbf{v}_k$ carrying a particle a perpendicular wavelength:

$$\tau_k = \Omega_k^{-1}.$$

One should consider this last option when the levels of fluctuations are very high. The expression of Ω_k can be written as the following:

$$\Omega_k = k_\perp \delta \mathbf{v}_k = \frac{k_\perp^2 \delta\phi_k}{B}. \tag{1.16}$$

It means that for high levels of fluctuations the diffusion coefficient becomes linear in $\delta\phi$:

$$D = \sum_k \frac{\delta\phi_k}{B}. \tag{1.17}$$

In case these drifts are in phase with fluctuations of other quantities, density or temperature for example, then a net flux of particles can be generated in the radial direction. Despite the fact that microturbulent fluctuations are characterised by small amplitudes, typically 10^{-3} in the centre of large fusion devices, the associated transport fluxes can be sufficiently high to explain the experimental measurements. It would be very interesting to continue to analyse these phenomena since they can be responsible for the anomalous transport of particles that can lead to the massive decrease in confinement time.

1.4 Kinetic plasma modelling and gyrokinetic models

Plasma physics and dynamics can be investigated through the use of different approaches, each of which is characterised by peculiarities and different approximations. Depending on the phenomena one wants to investigate other than the temporal scale of dynamics or the dimension of the volume under analysis one should use a different approach. The plasma dynamics can be described at different levels, from simulating the response of single particles inserted in an electromagnetic field through the use of simple equations of motion to the use of distribution function together with kinetic equations and also to consider the plasma as a quasi-neutral fluid that is capable of interacting with the electromagnetic field. Every approach has its pro and cons, the simulation of a large system may seem impossible for the single particle approach given the high number of associated equations while it is a standard task for the fluid approach. On the other hand, the study of small-scale phenomena such as the study of particles dynamics in turbulent electromagnetic fields results to be really difficult for the latter, depending on the situation one should choose the most efficient and suitable approach. For the purpose of this work, only kinetic approaches will be introduced and briefly analysed before introducing the gyrokinetic approach.

1.4.1 Equations of motion

This is the easiest approach from the theoretical point of view, we simply compute the values of the independent variables in the 6-D phase space of the solution obtaining the trajectories and the velocity of each particle over time. One has only to solve the initial value problem identified by the set of Eqs. [1.18], the problem is that the number of particles that can be simulated is limited since a great number of equations is associated with each particle.

$$\begin{cases} m\dfrac{d^2\boldsymbol{r}(t)}{dt^2} = e\Big(\boldsymbol{E}(\boldsymbol{r},t) + \dfrac{d\boldsymbol{r}(t)}{dt} \times \boldsymbol{B}(\boldsymbol{r},t)\Big), \\ \\ \left.\dfrac{d\boldsymbol{r}(t)}{dt}\right|_{t=0} = \boldsymbol{v_0}, \\ \\ \boldsymbol{r}(0) = \boldsymbol{r_0}. \end{cases} \tag{1.18}$$

Also without considering the coupling of the electromagnetic fields to Maxwell equations, and without considering the feedback effects the particles have on the EM fields since particles density inside a magnetic fusion device is in the order of 10^19m^{-3} it is easy to understand that this approach cannot be applied to simulate large systems, at least not with nowadays technology.

1.4.2 Vlasov-Maxwell system

Kinetic theory evaluates the behaviour of plasmas in terms of particle motion as well. Since the number of particles to be investigated is very large, a statistical approach is exploited, this description is carried out using the distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$. The latter measures the probability density for a single particle in the 6-D phase space $(\boldsymbol{x}, \boldsymbol{v})$. Each plasma species j is characterised by a specific distribution function that depends on space, velocity and time. The evaluation of $f_j(\boldsymbol{x}, \boldsymbol{v}, t)d^3\boldsymbol{x}d^3\boldsymbol{v}$ provides the number of particles which are located in the infinitesimal six-dimensional phase space $d^3\boldsymbol{x}d^3\boldsymbol{v}$ around the position $(\boldsymbol{x}, \boldsymbol{v})$ at which the distribution function has been evaluated time t. It is then necessary to use a so-called kinetic equation to study the evolution of the distribution function for each species J, one of the most famous ones is the Vlasov equation. In addition to that, we know that electromagnetic fields must respect Maxwell's equations, so it is necessary to couple the Vlasov equation together with this one, the obtained set of equations is the so-called Vlasov-Maxwell system and it is shown in Eq. [1.19], Eq. [1.20] and Eq. [1.21]:

$$\frac{\mathrm{d}}{\mathrm{d}t} f_j(\mathbf{x}, \mathbf{v}, t) = \Big[\mathbf{v}\frac{\partial}{\partial \mathbf{x}} + \frac{q_j}{m_j}(\mathbf{E}(\mathbf{x},\ \mathbf{t} + \mathbf{v} \times \mathbf{B}(\mathbf{x}, t))\frac{\partial}{\partial \mathbf{v}}\Big] f_j(\mathbf{x}, \mathbf{v}, t) = \mathcal{C}[f_j(\mathbf{x}, \mathbf{v}, t)]. \tag{1.19}$$

$$\nabla \cdot \mathbf{E} = \frac{1}{\varepsilon_0}\sum_j \rho_j \ ; \qquad \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}. \tag{1.20}$$

$$\nabla \cdot \mathbf{B} = 0 \ ; \qquad \nabla \times \mathbf{B} = \mu_0\Big(\sum_j \mathbf{j}_j + \varepsilon_0\frac{\partial \mathbf{E}}{\partial t}\Big). \tag{1.21}$$

Where q_j and m_j are respectively the charge and the mass of the species j, $\mathcal{C}$ is an operator that accounts for binary collisions, ρ_j is the charge density and $\mathbf{J}_j$ is the current charge density, the expressions of these last two quantities are evaluated through the integration of the distribution function:

$$\rho_j = q_j \int f_j d^3\mathbf{v} \ ; \qquad \mathbf{j}_j = q_j \int f_j \mathbf{v} d^3\mathbf{v}. \tag{1.22}$$

It is possible to obtain a self-description of the plasma through the coupled solution of Eqs. [1.19], [1.20] and [1.21]. The thing is that for a typical tokamak, the solution would account for plasma physics involving different temporal and spatial scales varying by several orders of magnitude. It is not possible to solve this kind of problem without introducing a set of approximations.

1.4.3 Gyrokinetic framework

It is well known that many plasma phenomena involve processes which are slow compared to the Larmor frequency and which vary slowly in space compared to the Larmor radius of a generic particle. It is possible to study this kind of phenomenon without moving from the kinetic approach, this can be done through the use of simpler and reduced kinetic equations called gyrokinetic equations. An accurate review of this approach can be found at [BH07]. In general, gyrokinetic theory and the related distribution functions, deal with the dynamics of gyro-centres, they are slightly displaced with respect to the guiding centres, and they account for the feedback effects of the electromagnetic fields with respect to the charge of the single particle. The main advantage of such kinetics equations lies in the reduced dimensionality of the phase-space, an average over the Fast Larmor motion is performed, this is called also gyro-average and it reduces the phase space dimension by 1. The resulting independent variables are the three coordinates of the gyrocentre $\mathbf{r}_{gy}$, the velocity component along the magnetic field lines $v_{\|gy}$ and the magnetic moment of the latter μ_{gy}:

$$f(\mathbf{x}, \mathbf{v}, t) \Rightarrow f(\mathbf{r}_{gy}, v_{\|gy}, \mu_{gy}, t). \tag{1.23}$$

In addition to the reduction of the order of the system, one has also to remember that the dynamics associated with the gyrocentres is smoother with respect to the one of particles, this means that the time discretization with which the solution is obtained can be much larger than the used for a full kinetic approach. One doesn't have to account for the fast dynamics around the magnetic field lines, this simplifies a lot the investigation of the dynamics from a numerical point of view.

1.4.4 Guiding-Center approach

The fact that one should account for the feedback effects from the electromagnetic fields in order to study the dynamics of the gyrocentres complicates the equations and the solution for sure. It is possible to not account for these effects and study directly the dynamics of Guiding-centres through the use of guiding centre theory from [Lit79], [Lit80], [Lit81], [Lit83] and [CB09]. The particles dynamics that one investigates in this way is the dynamics of the so-called test particles, these ones are not producing any effect in the electromagnetic fields despite the fact they have a charge. This kind of model can still be very interesting to investigate particular regimes or phenomena happening in a plasma. It is a simplified approach with respect to a full gyrokinetic one but it can furnish

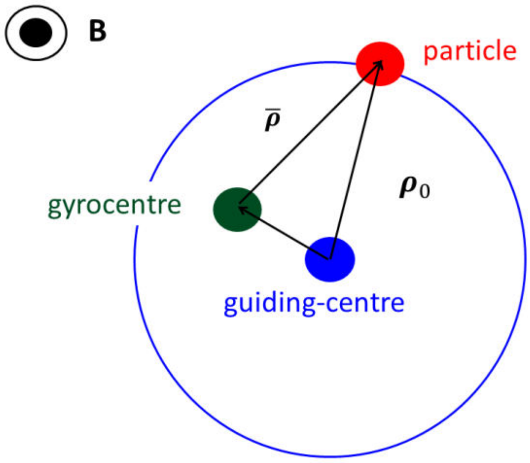


Figure 1.8: Schematic projection of particle, guiding-centre and gyrocentre positions on the transverse plane, extracted from [GL21].

a lot of useful information without being characterised by having a high computational cost. For the purpose of the work, it has been decided to not use distribution functions, the approach is more similar to the one of the equations of motion, briefly described in Subsec. 1.4.1, but the equations will be the ones related to guiding centre dynamics in the gyrokinetic framework. This can be done only for a limited number of particles naturally, since mechanical statistics is not involved, and only because we are considering only test particles.

1.5 Scope and outline of the thesis

The purpose of my work is to investigate the transport of particles due to turbulence in an electrostatic field, the topic has been already introduced in Sec. 1.3, and this is exploited through the use of guiding centre approach on test particles dynamics. Since we are interested only in the motion on the transverse plane (associated with transport), a further simplification has been introduced, the motion of particles is investigated only in two dimensions, and the investigation of dynamics along the third dimension is not of our interest and it is not influenced by the considered magnetic field. **B** is chosen as uniform and constant along this third dimension, the system is investigated in a slab geometry as a matter of fact. As anticipated, the motion of particles is analysed through the use of governing equations derived in a gyrokinetic framework, the derivation of the latter is shown in the next chapter. The considered framework can be considered as a non-usual gyrokinetic one since the observable expressed in series, e.g. the Hamiltonian, are expressed considering three terms of these series, up to the second order, nowadays the standard is considering only the zeroth and first-order ones. Since the second order is usually considered negligible and its implementation is quite complex, it has not been inserted in the majority of gyrokinetics code. Since one of the main problems in fusion devices is associated to transport due to turbulence, the idea is to investigate the dynamics of our simplified system accounting for this additional term in order to analyse any resulting regime and to evaluate the accuracy of this kind of analysis with respect to full kinetic approach, the one exploited through the integration of standard equations of motion. This work is mostly based on [Lit79], [Cha21] and [TC18], the model has been checked

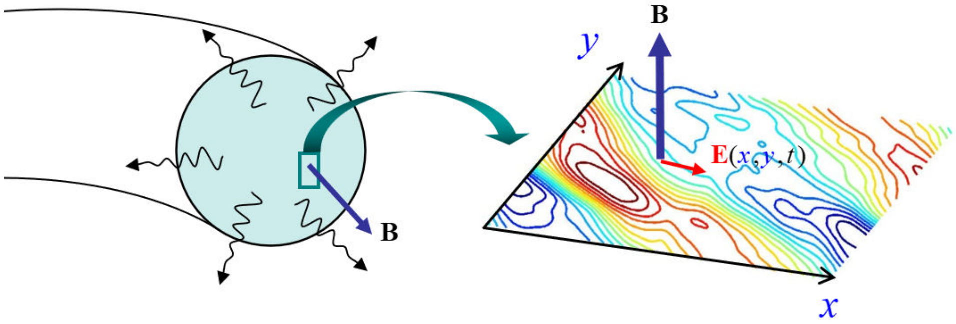


Figure 1.9: Geometry of the problem under analysis, focus on the turbulent potential.

again and further calculations to obtain the expression of the third order of the Hamiltonian have been performed. The numerical results are obtained from a code developed by Cristel Chandre, Research Professor at Centre National de la Recherche Scienitfique (CNRS) at Institut Mathématiques de Marseille. The code is available (on request) at `github.com/cchandre/Guiding-Center/` it computes the guiding centres' trajectories (as well as particles themselves now) in the above-described framework while considering a mock turbulent electrostatic potential similar to the one described in [PVM+88]. The work has been carried out at the Institut de Recherche sur la Fusion par Confinement Magnétique (IRFM) in the Groupe Thèorie et Simulation Numérique (GTSN) group at CEA (Commissariat à l'Énergie Atomique et aux Énergies Alternatives) in Cadarache (FR) under the supervision of Guido Ciraolo, who leads the research group working on numerical simulations of plasma physics. The manuscript is structured as described in the following bullet list:

- **Chapter 2** Contains the introduction to the complete starting model, the change of coordinates to guiding centre ones, the procedure to obtain the expression of the Hamiltonian in the gyrokinetic framework, the procedure to obtain the governing equations for guiding centres and particles and finally the procedure to obtain the equations for computing the total energy of particles over time.

- **Chapter 3** Includes the description of the code, starting from the explanation of how the code works, the three `.py` files are deeply described so that one can easily understand how to run the code in a proper way while fixing specific parameters other than having understood how the equations derived in Chapter 2 are implemented inside the code. It also explains how to generate certain types of plots directly at the end of the execution other than how to store save and analyse data from the code.

- **Chapter 4** Contains the presentation of the main numerical results, the chapter starts from the check on the proper functioning of `fo2gc` function, and it passes then to the time-step analysis exploited to find the maximum value of `TimeStep` parameter, this is crucial for the quality and the accuracy of the results. The main part of this chapter is focused on the influence of physical parameters over particle dynamics other than investigating an anomalous super-diffusive regime leading to a ballistic behaviour of some particles. The final part of the chapter compares the dynamics obtained from the guiding centre approach to the one that it is possible to obtain thanks to the standard integration of the equation of motion.

- **Chapter 5** Summarises the results obtained in the present work while trying to highlight the innovative consequences of these ones. A little focus on future perspectives of the work is also presented.

Chapter 2

Model description

2.1 Introduction to the model and context

As anticipated in chapter 1, the study of transport and diffusion associated with turbulence in an electromagnetic field is of great interest. As we know, charged particles interact with the fields and induce a sort of feedback effect in them, this is due to the charge of the particles themselves. If we consider a high-density plasma, the effect of a single particle can not be neglected, the global charge can strongly influence the electromagnetic fields and the dynamics of the system. Through the coupling with Maxwell equations of the desired equation for studying particles dynamics, it is possible to find a solution considering this effect. In this report, the system analysed does not account for this interaction, not completely, the considered potential map already accounts for the presence of the particles given that its analytical expression has been derived from a measured potential inside a real device, the particles have already influenced the measurements. It is possible to say that we are considering the motion of test particles, this means that there is no feedback effect on electromagnetic fields from the particles themselves. Those particles are non-interacting with each other, that's the classical assumption of non-interaction that characterizes the Vlasov equation. Despite the fact that this system is simplified, its dynamic is of great interest for multiple applications. A simple mathematical approach to finding a solution for this problem is solving the equations of motion:

$$m\ddot{\mathbf{x}} = q(\mathbf{E} + \dot{\mathbf{x}} \times \mathbf{B}). \tag{2.1}$$

$$\dot{\mathbf{x}} = \mathbf{v}. \tag{2.2}$$

Through the integration of $\ddot{x}$ and $\dot{x}$ it is possible to obtain the solution in the 6-dimensional phase space considering time dependence. Those calculations have a high computational cost since the system has 6 independent variables. In order to decrease the computational time we would like to reduce the order of the system, meaning decrease the number of independent variables. If it is possible that the system would be easier to study resulting in various advantages.

2.2 Complete model

It is now time to introduce and analyse a little the complete starting model, as anticipated, the system is characterised by a 6D phase space, and it is possible to define vector position $\mathbf{x}$ and velocity $\mathbf{v}$:

$$\mathbf{x} = (x, y, z)$$
$$\mathbf{v} = (v_x, v_y, v_z)$$

Some parameters of interest are q, the charge of the particles, m, the mass of the particles and B, the amplitude of the magnetic field. It is well known the expression of the Hamiltonian for this kind of system:

$$H = \frac{1}{2} m \boldsymbol{v}^{\mathbf{2}} + q\phi. \tag{2.3}$$

knowing that ϕ is the electrostatic potential it is valid:

$$\boldsymbol{E} = -\boldsymbol{\nabla}\phi$$

It is known also the expression of the Poisson bracket for this system:

$$\Big\{F, G\Big\} = \frac{1}{m}\Big(\frac{\partial F}{\partial \mathbf{x}}\frac{\partial G}{\partial \mathbf{v}} - \frac{\partial F}{\partial \mathbf{v}}\frac{\partial G}{\partial \mathbf{x}}\Big) + \frac{qB}{m^2}\Big(\frac{\partial F}{\partial v} \times \frac{\partial G}{\partial v}\Big). \tag{2.4}$$

Using an identity this last expression can be written in a new form:

$$\Big\{F, G\Big\} = \frac{1}{m}\Big(\frac{\partial F}{\partial \mathbf{x}}\frac{\partial G}{\partial \mathbf{v}} - \frac{\partial F}{\partial \mathbf{v}}\frac{\partial G}{\partial \mathbf{x}}\Big) + \frac{\partial F}{\partial \mathbf{v}} \cdot \Big(\frac{\partial G}{\partial \mathbf{v}} \times \frac{qB}{m^2}\Big). \tag{2.5}$$

It is possible to study the evolution in time of a generic observable F obtaining the expression for its derivatives in time, to do that it is just necessary to calculate the Poisson bracket of the generic observable and the Hamiltonian itself, therefore it is generally valid the following expression:

$$\dot{F} = \Big\{F, H\Big\}$$

For instance, we can test it on a generic independent variable:

$$\Big\{v_x, H\Big\} = -\frac{1}{m}\Big(\frac{\partial H}{\partial x}\Big) + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \cdot \Big(\frac{\partial H}{\partial \boldsymbol{v}} \times \frac{qB}{m^2}\Big)$$

Since the Poisson brackets applied to an independent variable are giving us two contributions, it is possible to conclude that the actual brackets are not canonical. It is also possible to affirm that the system under analysis is physical, this means that I am able to measure the velocity and the position of a particle for example. Later on, we will see that, while referring to guiding centres, the system won't be physical anymore. The guiding centre doesn't exist in reality and I can't measure its velocity and position. Once the reference system takes advantage of the position of the guiding centre, the system is no more physical but we will see that its Poisson Bracket will be canonical.

2.3 Change of coordinates

It is needed to refer to a new reference system, the one taking advantage of the position of the guiding centre. The latter is a non-physical point around which the single particle rotates with a radius equal to the well-known *Larmor* radius, it is known also as *gyroradius* and its expression is:

$$\rho = \frac{m}{\|q\|B}\sqrt{v_x^2 + v_y^2}$$

It is also well known which is the frequency with which the particle rotates around the guiding centre, it is called Cyclotron frequency and its expression is:

$$\Omega = \frac{qB}{2\pi m}. \tag{2.6}$$

Studying the solution of the equation of motion in a simplified case of electromagnetic field it is possible to calculate easily the position of this point, resulting in:

$$\boldsymbol{\chi} = \mathbf{x} - \boldsymbol{\rho} = \mathbf{x} - \frac{m}{qB}\hat{\mathbf{z}} \times \ \mathbf{v}. \tag{2.7}$$

The first simplification introduced in the system is the geometry, a slab one is considered. The magnetic field is considered uniform and constant along $\hat{\ \mathbf{z}}$:

$$\boldsymbol{B} = B\hat{\mathbf{z}}$$

The potential is considered constant along $\hat{\mathbf{z}}$ as well, it varies only in time and in the transverse plan:

$$\phi = \phi(x, y, t)$$

The particle and the guiding centre will have the same velocity along $\hat{\mathbf{z}}$ but it will remain constant $\Rightarrow v_z = Cst$. It is possible to calculate the expression of $\dot{z}$ and $\dot{v_z}$ like we did before:

$$\dot{z} = \Big\{z, H\Big\} = \frac{1}{m}\frac{\partial H}{\partial v_z} = v_z. \tag{2.8}$$

$$\dot{v_z} = \Big\{v_z, H\Big\} = -\frac{1}{m}\frac{\partial H}{\partial z} = 0. \tag{2.9}$$

In Eq. [2.8] and [2.9] it has been considered the expression of Hamiltonian of Eq. [2.3]. Knowing that z varies linearly and that v_z is a constant could allow us to not study the motion in these two independent variables, reducing the order of the system, but for now, we maintain the terms associated with their study so that the procedure is more general. In theory, the order of the system has already been reduced of 2 letting us have a system of order 4. It is now necessary to perform the first change of coordinates, naturally, this needs to be invertible in order to let us the possibility of going back once we have obtained the results in the new reference system. Invertible transformations mean that the function that binds two variables has to be bijective, from a value of the starting variable I obtain only one value of the arriving variable. In theory, the transformation should also be considered sufficiently smooth, this means that it doesn't introduce any singularity in the functions and in their derivatives. As it was anticipated, performing the first change of coordinates means moving into the system of coordinates of the guiding center:

$$(x, y, z, v_x, v_y, v_z) \;\Rightarrow\; (\chi_1, \chi_2, \chi_3, \rho, \theta, v_\parallel)$$

It is possible to represent the global change of variables through the following relations:

$$\mathbf{x} = (x_1, x_2, x_3) \;\Rightarrow\; \boldsymbol{\chi} = \boldsymbol{x} - \frac{m}{qB}\hat{\mathbf{z}} \times \boldsymbol{v} = (\chi_1, \chi_2, \chi_3). \tag{2.10}$$

$$\boldsymbol{v}_\perp = (v_x, v_y) \;\Rightarrow\; \rho = \frac{m}{|q|B}\sqrt{v_x^2 + v_y^2}. \tag{2.11}$$

$$\boldsymbol{v}_\perp = (v_x, v_y) \;\Rightarrow\; \theta = \tan^{-1}\left(\frac{v_x}{v_y}\right). \tag{2.12}$$

$$v_z = v_\parallel. \tag{2.13}$$

Where χ_1, χ_2, and χ_3 are the coordinates of the guiding centre in a Cartesian reference system. ρ and θ are respectively the Larmor radius and the gyro-angle. Firstly it is necessary to check that the transformation that is taking place is bijective, this is a very important requirement, in order to do that it should be obtained the expression of all the old variables as a function of the new ones. The procedure is quite simple, from Eq. [2.10] it is possible to obtain:

$$x = \chi_1 + \rho\cos\theta. \tag{2.14}$$

$$y = \chi_2 - \rho\sin\theta. \tag{2.15}$$

From Eq. [2.11] it is possible to obtain:

$$\sqrt{v_x^2 + v_y^2} = \rho\frac{|q|B}{m} = v_\perp. \tag{2.16}$$

Given the considered reference system the velocity just obtained comes with a direction, it is identifiable with:

$$\hat{\perp} = -\sin\theta\;\hat{\mathbf{x}} - \cos\theta\;\hat{\mathbf{y}}. \tag{2.17}$$

Using Eq. [2.16] and [2.17] we can obtain x and y components of $v_\perp$:

$$v_{\perp x} = v_x = -\rho\frac{|q|B}{m}\sin\theta. \tag{2.18}$$

$$v_{\perp y} = v_y = -\rho\frac{|q|B}{m}\cos\theta. \tag{2.19}$$

As anticipated from Eq. [2.13] the expression of v_z as a function of $v_\parallel$ is already explicit. This with Eq. [2.14], [2.15], [2.18] and [2.19] is the confirmation that the transformation is invertible, so it is possible to proceed with the procedure. The transformation that is taking place is not canonical, the Poisson bracket is being modified. To calculate the new expression of the P.B. it is necessary the use of square matrix J that accounts for the change of coordinates, meaning that it contains the information linking the old and the new coordinates. In particular, each element of the generalized version of J is a Poisson bracket that acts on the new coordinates and has the old ones as its variables. In general, it is valid:

$$\Big\{F,G\Big\}=\sum_{k,l=1}^{M}\frac{\partial F}{\partial z'_k}J_{kl}(\boldsymbol{z'})\frac{\partial G}{\partial z'_l}=\frac{\partial F}{\partial \boldsymbol{z'}}J(\boldsymbol{z'})\Big(\frac{\partial G}{\partial \boldsymbol{z'}}\Big)^T. \tag{2.20}$$

With

$$k,l=1,...,M\ \ :J_{kl}(\boldsymbol{z'})=\Big\{z'_k,z'_l\Big\}.$$

It is required the expression of the elements of the matrix $J(\boldsymbol{z'})$ in order to obtain the final expression of the PB. In general, it is valid:

$$k,l=1,...,2N\ \ :\ \ J_{kl}(\boldsymbol{z'})=\Big\{z'_k,z'_l\Big\}=\sum_{k,l=1}^{2N}\frac{\partial z'_k}{\partial z_k}J_{kl}(\boldsymbol{z})\frac{\partial z'_l}{\partial z_l}=\frac{\partial z'_k}{\partial \boldsymbol{z}}J(\boldsymbol{z})\Big(\frac{\partial z'_l}{\partial \boldsymbol{z}}\Big)^T. \tag{2.21}$$

As we can see from Eq. [2.21] it is required to know the matrix $J(\boldsymbol{z})$. The general expression of the latter is the following:

$$J(\boldsymbol{z})=\begin{pmatrix}\{x_1,x_1\} & \{x_1,x_2\} & \{x_1,x_3\} & \{x_1,v_1\} & \{x_1,v_2\} & \{x_1,v_3\}\\ \{x_2,x_1\} & \{x_2,x_2\} & \{x_2,x_3\} & \{x_2,v_1\} & \{x_2,v_2\} & \{x_2,v_3\}\\ \{x_3,x_1\} & \{x_3,x_2\} & \{x_3,x_3\} & \{x_3,v_1\} & \{x_3,v_2\} & \{x_3,v_3\}\\ \{v_1,x_1\} & \{v_1,x_2\} & \{v_1,x_3\} & \{v_1,v_1\} & \{v_1,v_2\} & \{v_1,v_3\}\\ \{v_2,x_1\} & \{v_2,x_2\} & \{v_2,x_3\} & \{v_2,v_1\} & \{v_2,v_2\} & \{v_2,v_3\}\\ \{v_3,x_1\} & \{v_3,x_2\} & \{v_3,x_3\} & \{v_3,v_1\} & \{v_3,v_2\} & \{v_3,v_3\}\end{pmatrix}. \tag{2.22}$$

As anticipated each of these elements is a Poisson bracket of which we know the expression from Eq. [2.5]. In fact, the expression of the elements of $J(\boldsymbol{z})$ is the same as the ones belonging to $J(\boldsymbol{z'})$. It is possible to write:

$$k,l=1,...,M\ \ :J_{kl}(\boldsymbol{z})=\Big\{z_k,z_l\Big\}. \tag{2.23}$$

The PB is the one coming from Eq. [2.5] as anticipated. Through the computing of each element it is possible to obtain the following matrix $J(\boldsymbol{z})$:

$$J(\boldsymbol{z})=\begin{pmatrix}0 & 0 & 0 & \frac{1}{m} & 0 & 0\\ 0 & 0 & 0 & 0 & \frac{1}{m} & 0\\ 0 & 0 & 0 & 0 & 0 & \frac{1}{m}\\ -\frac{1}{m} & 0 & 0 & 0 & \frac{B\,e}{m^2} & 0\\ 0 & -\frac{1}{m} & 0 & -\frac{B\,e}{m^2} & 0 & 0\\ 0 & 0 & -\frac{1}{m} & 0 & 0 & 0\end{pmatrix}. \tag{2.24}$$

Now it is time to compute the elements of the next matrix. Firstly it is necessary to introduce the general expression of $J(\boldsymbol{z'})$:

$$J(\boldsymbol{z'})=\begin{pmatrix}\{\chi_1,\chi_1\} & \{\chi_1,\chi_2\} & \{\chi_1,\chi_3\} & \{\chi_1,\rho\} & \{\chi_1,\theta\} & \{\chi_1,v_\parallel\}\\ \{\chi_2,\chi_1\} & \{\chi_2,\chi_2\} & \{\chi_2,\chi_3\} & \{\chi_2,\rho\} & \{\chi_2,\theta\} & \{\chi_2,v_\parallel\}\\ \{\chi_3,\chi_1\} & \{\chi_3,\chi_2\} & \{\chi_3,\chi_3\} & \{\chi_3,\rho\} & \{\chi_3,\theta\} & \{\chi_3,v_\parallel\}\\ \{\rho,\chi_1\} & \{\rho,\chi_2\} & \{\rho,\chi_3\} & \{\rho,\rho\} & \{\rho,\theta\} & \{\rho,v_\parallel\}\\ \{\theta,\chi_1\} & \{\theta,\chi_2\} & \{\theta,\chi_3\} & \{\theta,\rho\} & \{\theta,\theta\} & \{\theta,v_\parallel\}\\ \{v_\parallel,\chi_1\} & \{v_\parallel,\chi_2\} & \{v_\parallel,\chi_3\} & \{v_\parallel,\rho\} & \{v_\parallel,\theta\} & \{v_\parallel,v_\parallel\}\end{pmatrix}. \tag{2.25}$$

Taking advantage of Eq. [2.21] and knowing the expression of all the necessary terms from $J(\boldsymbol{z})$ we are able to do all the necessary calculations obtaining the final expression of $J(\boldsymbol{z'})$ as done before:

$$J(\boldsymbol{z'}) = \begin{pmatrix} 0 & -\frac{1}{B\,e} & 0 & 0 & 0 & 0 \\ \frac{1}{B\,e} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{m} \\ 0 & 0 & 0 & 0 & -\frac{e}{m\,|e|\,\sqrt{{v_x}^2+{v_y}^2}} & 0 \\ 0 & 0 & 0 & \frac{e}{m\,|e|\,\sqrt{{v_x}^2+{v_y}^2}} & 0 & 0 \\ 0 & 0 & -\frac{1}{m} & 0 & 0 & 0 \end{pmatrix}. \tag{2.26}$$

Now that all these elements are known, it is possible to obtain the expression of the modified Poisson Bracket through Eq. [2.20]:

$$\Big\{F,G\Big\} = \frac{1}{q\rho B}\Big(\frac{\partial F}{\partial \theta}\frac{\partial G}{\partial \rho} - \frac{\partial F}{\partial \rho}\frac{\partial G}{\partial \theta}\Big) - \frac{1}{qB}\Big(\frac{\partial F}{\partial \chi_1}\frac{\partial G}{\partial \chi_2} - \frac{\partial F}{\partial \chi_2}\frac{\partial G}{\partial \chi_1}\Big) + \frac{1}{m}\Big(\frac{\partial F}{\partial \chi_3}\frac{\partial G}{\partial u_\parallel} - \frac{\partial F}{\partial u_\parallel}\frac{\partial G}{\partial \chi_3}\Big). \tag{2.27}$$

Before proceeding it would be advisable to make a change in the variables introducing the magnetic moment μ in place of gyro-radius ρ. Giving the definition of the new variable:

$$\mu = \frac{mv_\perp^2}{2B}.$$

It is possible to find the link between the two variables that are being exchanged giving that:

$$v_\perp^2 = \frac{\rho^2|q|^2B^2}{m^2} \quad \Rightarrow \quad \mu = \frac{m}{2B}\frac{\rho^2|q|^2B^2}{m^2} = \frac{\rho^2 q^2 B}{2m}. \tag{2.28}$$

The expression of the Hamiltonian needs to be modified:

$$H = \mu B + \frac{1}{2}mv_\parallel + q\phi(\chi_1 + \rho\cos\theta, \chi_2 - \rho\cos\theta, \tau). \tag{2.29}$$

The expression of the Poisson brackets needs to be modified as well:

$$\Big\{F,G\Big\} = \frac{q}{m}\Big(\frac{\partial F}{\partial \theta}\frac{\partial G}{\partial \mu} - \frac{\partial F}{\partial \mu}\frac{\partial G}{\partial \theta}\Big) - \frac{1}{qB}\Big(\frac{\partial F}{\partial \chi_1}\frac{\partial G}{\partial \chi_2} - \frac{\partial F}{\partial \chi_2}\frac{\partial G}{\partial \chi_1}\Big) + \frac{1}{m}\Big(\frac{\partial F}{\partial \chi_3}\frac{\partial G}{\partial u_\parallel} - \frac{\partial F}{\partial u_\parallel}\frac{\partial G}{\partial \chi_3}\Big). \tag{2.30}$$

It is possible to check if this P.B. is canonical, as it has been done before:

$$\dot{\theta} = \Big\{\theta, H\Big\} = \frac{q}{m}\frac{\partial H}{\partial \mu} = \frac{q}{m}B = \frac{qB}{m} = \omega_{cyclotron}.$$

with $\omega_{cyclotron}$ known also as cyclotron angular velocity. It is clear that whenever we put an independent variable inside these P.B. only one term gives its contribute meaning that those brackets are canonical. From now on we do not want to change the brackets anymore, in order to do that it is necessary to perform only canonical transformations. The goal is continuing the reduction of the order of the system, in particular, it is necessary to eliminate the dependency on θ, that's because the fast rotating dynamic isn't interesting in order to obtain the trajectory of the guiding centre, representing the mean trajectory of the particle. To calculate the correct trajectory of the ions (including the fast dynamics) smaller time steps for the integration are required compared to what could be used use if it has to be calculated only the guiding centre trajectory. The computational time required for obtaining the same solution using a much smaller time step is significantly higher. It is important to understand that we are not going to simply eliminate the dependency but at the end of the process, θ won't be a variable anymore. It is necessary to account for the fact that the particle is rotating around the guiding centre and that the Electromagnetic field acts precisely on the latter, the field can be different at two different θ angles influencing the trajectory of the particle itself. What is possible to do is to consider an average effect of the potential for every revolution of the particle around the G.C. Consequently, the Hamiltonian becomes:

$$H(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau, v_\parallel) = \mu B + \frac{1}{2} m \boldsymbol{v}_\parallel^2 + q\phi(\chi_1, \chi_2, \chi_3, \mu, \theta, v_\parallel). \tag{2.31}$$

In order to account for the contribution of θ on every revolution we simply integrate over the latter so that we account for its influence without having its contribution at the end of the procedure. We define the gyro-average operator as:

$$\langle f \rangle_\theta = \frac{1}{2\pi} \int_0^{2\pi} f(x, \theta, ...)\, d\theta = J_0(f).$$

As it was mentioned before the averaging over θ will be the result of a canonical transformation, we mentioned that from now on the Poisson bracket shouldn't be modified. On the other hand, the expression of the Hamiltonian will be changed due to the transformation itself.

2.4 Autonomization of the system

The first thing that is necessary to do before modifying anything in the model is the autonomization of the system. Once we have done this procedure, the time will become variable and the solution will be easier to find. The system will not depend explicitly on the independent variable time, it will be called a time-invariant system. The price for doing that will be the introduction of another variable k, the energy of the system in our case. Overall, therefore, there is an increase of two independent variables. As anticipated, this procedure is very useful whenever the system is characterised by a Hamiltonian with the following expression:

$$H = H(z(t), t).$$

The main concept behind this procedure is taking advantage of:

$$z = f(z, t) = f(z, \tau) \quad with \; \tau = t \quad \Rightarrow \quad \dot{\tau} = 1.$$

τ is the new variable time and it depends on time as well, we cancel fictitiously the time dependence from our function making time a variable. It is possible to reduce the latter to:

$$H(z, t, k) = H(z, \tau) + k. \tag{2.32}$$

The system now has two more independent variables, the system at this moment has 8 independent variables indeed. It may seem not an advantage to have increased the order of the system but during the resolution of equations, the system will be easier to be handled. The new system is characterised by observables that depend on the new variable τ and k so it is necessary to modify the Poisson bracket in order to obtain a bracket that truly describes the exact evolution of the state of the system. The extended P.B. is slightly different from the original one:

$$\Big\{F, G\Big\}_\tau = \Big\{F, G\Big\} + \Big(\frac{\partial F}{\partial t}\frac{\partial G}{\partial k} - \frac{\partial F}{\partial k}\frac{\partial G}{\partial t}\Big). \tag{2.33}$$

Following Eq. [2.32] and Eq. [2.33] it is possible to write the new expression of the Hamiltonian and of the Poisson Bracket:

$$H(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau, v_\parallel, k) = \mu B + \frac{1}{2}m\boldsymbol{v}_\parallel^2 + q\phi(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau) + k. \tag{2.34}$$

$$\begin{aligned} \Big\{F, G\Big\}_\tau = & \frac{q}{m}\Big(\frac{\partial F}{\partial \theta}\frac{\partial G}{\partial \mu} - \frac{\partial F}{\partial \mu}\frac{\partial G}{\partial \theta}\Big) - \frac{1}{qB}\Big(\frac{\partial F}{\partial \chi_1}\frac{\partial G}{\partial \chi_2} - \frac{\partial F}{\partial \chi_2}\frac{\partial G}{\partial \chi_1}\Big) + \\ & + \frac{1}{m}\Big(\frac{\partial F}{\partial \chi_3}\frac{\partial G}{\partial u_\parallel} - \frac{\partial F}{\partial u_\parallel}\frac{\partial G}{\partial \chi_3}\Big) + \Big(\frac{\partial F}{\partial t}\frac{\partial G}{\partial k} - \frac{\partial F}{\partial k}\frac{\partial G}{\partial t}\Big). \end{aligned} \tag{2.35}$$

Now that the system is autonomized, it is possible to say that the energy is conserved if the system is isolated and it is possible to establish how the latter enters or leaves the system. As usual, it is possible to simply write:

$$\dot{k} = \Big\{k, H\Big\} = -\frac{\partial H}{\partial t} = q\frac{\partial \phi}{\partial t}. \tag{2.36}$$

From this equation, it can be clearly seen that the energy can be transferred inside and outside the system only through the potential. Now the system is autonomized and ready to be reduced with gyro-kinetic theory.

2.5 Assumptions on the electrostatic field

Some assumptions have to be made to ensure that the procedure does not contain too large errors, if the potential is varying too rapidly in space and time we would produce inaccurate results as well as the potential has an excessive amplitude, the resulting assumptions are the following:

- Low amplitude of the potential: $\rightarrow \varepsilon_\delta \phi(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau)$.
- Potential slowly varying in time: $\rightarrow \phi(\chi_1, \chi_2, \chi_3, \mu, \theta, \varepsilon_\omega \tau)$ with $\varepsilon_\omega = \frac{\tau}{T}$.
- Potential slowly varying along $\hat{\mathbf{z}}$: $\rightarrow \phi(\chi_1, \chi_2, \varepsilon_\parallel \chi_3, \mu, \theta, \tau)$ with $\varepsilon_\parallel = \frac{L_\perp}{L_\parallel}$.

Choice of ordering:

$$\varepsilon_\delta = \varepsilon_\omega = \varepsilon_\parallel.$$

The resulting expression of the potential is the following:

$$\varepsilon_\delta \phi(\chi_1, \chi_2, \varepsilon_\delta \chi_3, \mu, \theta, \varepsilon_\delta \tau).$$

Taking into account these last assumptions we obtain our final Hamiltonian formulation:

$$H(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau, v_\|, k) = \mu B + \frac{1}{2} m \boldsymbol{v}_\|^2 + q \varepsilon_\delta \phi(\chi_1, \chi_2, \varepsilon_\delta \chi_3, \mu, \theta, \varepsilon_\delta \tau) + k. \tag{2.37}$$

2.6 Application of Lie transform

Now it is time to perform the change of coordinates in order to eliminate the θ dependence. A summary of what Lie transform is and how to exploit this kind of transformation can be found in Sec. A.2. In order to do that the main goal is choosing the right generating function. It is necessary to change coordinates and the consequent definition of a specific generating function for each order of H in which one wants to annul the dependency. It is recalled the expression of the Hamiltonian for this system:

$$H(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau, k, v_\|) = \mu B + \frac{1}{2} m \boldsymbol{v}_\|^2 + q \varepsilon_\delta \phi(\chi_1, \chi_2, \varepsilon_\delta \chi_3, \mu, \theta, \varepsilon_\delta \tau) + k.$$

The dependence on the θ variable is contained in the potential contribution. It is needed to recall the assumptions made on the potential:

- Low amplitude of the potential.
- Potential slowly varying in time.
- Potential slowly varying along $\hat{\mathbf{z}}$.

It is possible to write the Hamiltonian as in the theory of perturbation:

$$H(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau) = H_0(\mu, v_\|, k) + \varepsilon_\delta V(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau).$$

through the definition of:

- $H_0 = H_0(\mu, v_\|, k) = \mu B + \frac{1}{2} m \boldsymbol{v}_\|^2 + k.$
- $\varepsilon_\delta V = \varepsilon_\delta V(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau) = q \varepsilon_\delta \phi(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau).$

Before starting the procedure to obtain the modified expression of Hamiltonian, at which Lie transforms have been applied at the desired order, it is necessary to make some clarifications on Poisson Brackets. They are expressed as the extended form of Eq. [2.30]. It is possible to define some sub-brackets:

- $\Big\{F, G\Big\}_{\mathrm{g}} = \Big\{F, G\Big\}_{(\theta,\mu)} = \frac{q}{m}\Big(\frac{\partial F}{\partial \theta}\frac{\partial G}{\partial \mu} - \frac{\partial F}{\partial \mu}\frac{\partial G}{\partial \theta}\Big).$
- $\Big\{F, G\Big\}_{\perp} = \Big\{F, G\Big\}_{(\chi_1,\chi_2)} = -\frac{1}{qB}\Big(\frac{\partial F}{\partial \chi_1}\frac{\partial G}{\partial \chi_2} - \frac{\partial F}{\partial \chi_2}\frac{\partial G}{\partial \chi_1}\Big).$

- $\left\{F, G\right\}_{\|} = \left\{F, G\right\}_{(v_{\|}, \chi_3)} = +\frac{1}{m}\left(\frac{\partial F}{\partial \chi_3}\frac{\partial G}{\partial u_{\|}} - \frac{\partial F}{\partial u_{\|}}\frac{\partial G}{\partial \chi_3}\right).$
- $\left\{F, G\right\}_{\mathrm{t}} = \left\{F, G\right\}_{(\tau, k)} = +\left(\frac{\partial F}{\partial t}\frac{\partial G}{\partial k} - \frac{\partial F}{\partial k}\frac{\partial G}{\partial t}\right).$

When it is necessary to develop a Poisson Bracket it is possible to take advantage of the following expression:

$$\left\{F, G\right\} = \frac{1}{\varepsilon}\left\{F, G\right\}_{\mathrm{g}} + \left\{F, G\right\}_{\perp} + \left\{F, G\right\}_{\|} + \left\{F, G\right\}_{\mathrm{t}}. \tag{2.38}$$

Given that the observables to which those brackets are applied are slowly varying in time and slowly varying along $\hat{\mathbf{z}}$ direction we expect it is possible to write the following modified expression:

$$\left\{F, G\right\} = \frac{1}{\varepsilon}\left\{F, G\right\}_{\mathrm{g}} + \left\{F, G\right\}_{\perp} + \varepsilon_{\delta}\left\{F, G\right\}_{\|} + \varepsilon_{\delta}\left\{F, G\right\}_{\mathrm{t}}. \tag{2.39}$$

Taking into account this, some terms will move from the first to the third order, for example, it is very important to consider this. Another important point is to analyse the application of the Poisson Bracket to a generic observable F and H_0. Before doing that it is necessary to calculate the derivatives of H_0 with respect to all the independent variables:

$$\frac{\partial H_0}{\partial \mu} = B;$$

$$\frac{\partial H_0}{\partial v_{\|}} = mv_{\|};$$

$$\frac{\partial H_0}{\partial k} = 1;$$

$$\frac{\partial H_0}{\partial t} \neq 0;$$

$$\frac{\partial H_0}{\partial \chi_1} = 0;$$

$$\frac{\partial H_0}{\partial \chi_2} = 0;$$

$$\frac{\partial H_0}{\partial \chi_3} = 0.$$

Given the validity of these last equations, it is possible to write the expression of the following useful relation:

$$\left\{F, H_0\right\} = \frac{1}{\varepsilon}\left\{F, H_0\right\}_{\mathrm{g}} + \varepsilon_{\delta}\left\{F, H_0\right\}_{\mathrm{t}}. \tag{2.40}$$

2.7 Gyrokinetic Hamiltonian model

In the following passages are shown the main terms obtained during the calculations of the Lie Transform at the third order, the procedure is almost the same as the second order but the number of terms has increased. The operator for the Lie transform has a different expression:

$$e^{-\varepsilon L_s} = \sum_{n=0}^{n=3} \frac{-\varepsilon^n L_s^n}{n!} = 1 - \varepsilon L_s + \frac{\varepsilon^2}{2} L_s^2 - \frac{\varepsilon^3}{6} L_s^3 + O(\varepsilon^4). \tag{2.41}$$

It is possible to express the generating function as a series:

$$S(z) = \sum_{n=0}^{+\infty} \varepsilon_\delta^n S_n(z) = \sum_{n=0}^{n=3} \varepsilon_\delta^n S_n(z) + O(\varepsilon_\delta^3) = S_0 + \varepsilon_\delta S_1 + \varepsilon_\delta^2 S_2 + \varepsilon_\delta^3 S_3 + O(\varepsilon_\delta^4).$$

The most complex term is coming out from L_s^3 applied to a generic observable F, it becomes:

$$L_s^3(F) = \Big\{S, \Big\{S, \Big\{S, F\Big\}\Big\}\Big\}.$$

In order to obtain the expression of the modified Hamiltonian H' it is simply needed to apply the operator for the Lie transform to the old version of H:

$$e^{-\varepsilon L_s} H = H' = H - \varepsilon\Big\{S, H\Big\} + \frac{\varepsilon^2}{2}\Big\{S, \Big\{S, H\Big\}\Big\} - \frac{\varepsilon^3}{6}\Big\{S, \Big\{S, \Big\{S, H\Big\}\Big\}\Big\}. \tag{2.42}$$

The modified expression of the Hamiltonian will account for first, second and third-order terms in ε_δ other than contributions that don't contain the latter. It is possible to take advantage of a useful notation:

$$H' = H_0 + H^{(\mathrm{I})} + H^{(\mathrm{II})} + H^{(\mathrm{III})}. \tag{2.43}$$

H has always the same expression, $H = H_0 + \varepsilon_\delta V$, it is possible to develop the terms in the various order in ε through the bi-linearity property of the Poisson Brackets:

$$-\varepsilon\Big\{S, H\Big\} = -\varepsilon\Big\{\varepsilon_\delta S_1 + \varepsilon_\delta^2 S_2 + \varepsilon_\delta^3 S_3, H_0 + \varepsilon_\delta V\Big\} =$$

$$= -\varepsilon\Big[\varepsilon_\delta\Big\{S_1, H_0\Big\} + \varepsilon_\delta^2\Big\{S_2, H_0\Big\} + \varepsilon_\delta^3\Big\{S_3, H_0\Big\} + \varepsilon_\delta^2\Big\{S_1, V\Big\} + \varepsilon_\delta^3\Big\{S_2, V\Big\} + O(\varepsilon_\delta^4)\Big].$$

It is possible to follow the same procedure for the terms in ε^2 and ε^3:

$$\begin{aligned}\frac{\varepsilon^2}{2}\Big\{S, \Big\{S, H\Big\}\Big\} = \frac{\varepsilon^2}{2}\Big\{&\varepsilon_\delta S_1 + \varepsilon_\delta^2 S_2 + \varepsilon_\delta^3 S_3, \varepsilon_\delta\Big\{S_1, H_0\Big\} + \varepsilon_\delta^2\Big\{S_2, H_0\Big\} + \varepsilon_\delta^3\Big\{S_3, H_0\Big\} + \\ &+ \varepsilon_\delta^2\Big\{S_1, V\Big\} + \varepsilon_\delta^3\Big\{S_2, V\Big\} + O(\varepsilon_\delta^4)\Big\}.\end{aligned} \tag{2.44}$$

While we are developing we don't write the terms over the third order in ε_δ:

$$\frac{\varepsilon^2}{2}\Big[\varepsilon_\delta^2\Big\{S_1, \Big\{S_1, H_0\Big\}\Big\} + \varepsilon_\delta^3\Big\{S_2, \Big\{S_1, H_0\Big\}\Big\} + \varepsilon_\delta^3\Big\{S_1, \Big\{S_2, H_0\Big\}\Big\} + \varepsilon_\delta^3\Big\{S_1, \Big\{S_1, V\Big\}\Big\}\Big].$$

For the third order in ε it remains only one term, all the others are of the fourth order in ε_δ:

$$-\frac{\varepsilon^3}{6}\Big\{S,\Big\{S,\Big\{S,H\Big\}\Big\}\Big\}= -\frac{\varepsilon^3}{6}\varepsilon_\delta^3\Big\{S_1,\Big\{S_1,\Big\{S_1,H_0\Big\}\Big\}\Big\}.$$

The following equation is the complete expression of the modified Hamiltonian divided in the various order of ε:

$$\begin{aligned}H' = H_0 + \varepsilon_\delta V - \varepsilon\Big[\varepsilon_\delta\Big\{S_1,H_0\Big\}+\varepsilon_\delta^2\Big\{S_2,H_0\Big\}+\varepsilon_\delta^3\Big\{S_3,H_0\Big\}+\varepsilon_\delta^2\Big\{S_1,V\Big\}+\varepsilon_\delta^3\Big\{S_2,V\Big\}\Big]+\\ \frac{\varepsilon^2}{2}\Big[\varepsilon_\delta^2\Big\{S_1,\Big\{S_1,H_0\Big\}\Big\}+\varepsilon_\delta^3\Big\{S_2,\Big\{S_1,H_0\Big\}\Big\}+\varepsilon_\delta^3\Big\{S_1,\Big\{S_2,H_0\Big\}\Big\}+\varepsilon_\delta^3\Big\{S_1,\Big\{S_1,V\Big\}\Big\}\Big]+\\ -\frac{\varepsilon^3}{6}\varepsilon_\delta^3\Big\{S_1,\Big\{S_1,\Big\{S_1,H_0\Big\}\Big\}\Big\}.\end{aligned} \tag{2.45}$$

It is possible now to subdivide in the order of ε_δ and calculate the final expression for each order.

2.7.1 first-order

The expression of first-order terms is:

$$\varepsilon_\delta\Big[V - \varepsilon\Big\{S1, H_0\Big\}\Big]. \tag{2.46}$$

It is possible to write V through the theory of linear perturbation considering the gyro-average we have already introduced:

$$V = \langle V\rangle + \tilde{V} \quad\Rightarrow\quad \varepsilon_\delta\Big[\langle V\rangle + \tilde{V} - \varepsilon\Big\{S_1,H_0\Big\}\Big].$$

From the previous subsection, it is also known that for a generic observable F it is valid:

$$\Big\{F,H_0\Big\}= \frac{1}{\varepsilon}\Big\{F,H_0\Big\}_{\mathrm{g}}+\varepsilon_\delta\Big\{F,H_0\Big\}_{\mathrm{t}}.$$

Given that we know the dependencies of H_0 that one can easily obtains:

$$\begin{aligned}\varepsilon_\delta\Big[\langle V\rangle_\theta + \tilde{V} - \Big\{S_1,H_0\Big\}\Big]= \varepsilon_\delta\Big[\langle V\rangle_\theta + \tilde{V} - \frac{q}{m}\Big(\frac{\partial S_1}{\partial\theta}\frac{\partial H_0}{\partial\mu} - \frac{\partial S_1}{\partial\mu}\frac{\partial H_0}{\partial\theta}\Big)-\varepsilon\varepsilon_\delta\Big(\frac{\partial S_1}{\partial t}\frac{\partial H_0}{\partial k}+\\ -\frac{\partial S_1}{\partial k}\frac{\partial H_0}{\partial t}\Big)\Big]= \varepsilon_\delta\Big[\langle V\rangle_\theta + \tilde{V} - \frac{qB}{m}\Big(\frac{\partial S_1}{\partial\theta}\Big)-\varepsilon\varepsilon_\delta\Big(\frac{\partial S_1}{\partial t}\Big)\Big].\end{aligned} \tag{2.47}$$

The last contribution is of third order and we will consider it later. The expression becomes:

$$\varepsilon_\delta\Big[\langle V\rangle_\theta + \tilde{V} - \frac{qB}{m}\Big(\frac{\partial S_1}{\partial\theta}\Big)\Big].$$

We are not interested in the averaged term but only in the fluctuating ones. We would like to annul the fluctuations, one possibility is defining the generating function so that the fluctuating terms annul each other:

$$\tilde{V} - \frac{qB}{m}\Big(\frac{\partial S_1}{\partial\theta}\Big)= 0. \tag{2.48}$$

It is now possible to find the expression of S_1:

$$S_1 = \frac{m}{qB}\int \tilde{V} d\theta. \tag{2.49}$$

Through the definition of the generating function of Eq. [2.49], in the first-order term, remains only one contribution, the averaged one of V, it is valid

$$\langle V\rangle_\theta = \frac{q}{2\pi}\int_0^{2\pi} \phi(x+\rho\cos\theta, y-\rho\sin\theta, t)\, d\theta. \tag{2.50}$$

The final expression of the term that will be implemented is:

$$H^{(\mathrm{I})} = \varepsilon_\delta q\langle\phi\rangle_\theta. \tag{2.51}$$

2.7.2 Second order

Now the cancellation of fluctuations in the second order of Eq. [2.45] is performed, the procedure is very similar:

$$\varepsilon_\delta^2\Big[-\varepsilon\Big\{S_1, V\Big\}-\varepsilon\Big\{S_2, H_0\Big\}+\frac{\varepsilon^2}{2}\Big\{S_1, \Big\{S_1, H_0\Big\}\Big\}\Big]. \tag{2.52}$$

One has to develop the terms using the relations previously introduced in order to not make any mistakes. It is possible to develop the first term of Eq. [2.52] taking advantage of Eq. [2.39]:

$$\Big\{S_1, V\Big\}= \frac{1}{\varepsilon}\Big\{S_1, V\Big\}_{\mathrm{g}}+\Big\{S_1, V\Big\}_{\perp}+\varepsilon_\delta\Big\{S_1, V\Big\}_{\parallel}+\varepsilon_\delta\Big\{S_1, V\Big\}_{\mathrm{t}}. \tag{2.53}$$

The last two terms of the previous Eq. are global of order $\varepsilon\varepsilon_\delta^3$ and they are neglected from further calculations while the second term is global of order $\varepsilon\varepsilon_\delta^2$ and it will be considered in third order calculations. Now it is possible to develop the second term of Eq. [2.52]:

$$\Big\{S_2, H_0\Big\}= \frac{1}{\varepsilon}\Big\{S_2, H_0\Big\}_{\mathrm{g}}+\varepsilon_\delta\Big\{S_2, H_0\Big\}_{\mathrm{t}}. \tag{2.54}$$

The last term of Eq. [2.54] is global of order $\varepsilon\varepsilon_\delta^3$ and will be neglected from further calculations. Finally, the last term of Eq. [2.52], requires a long procedure but it is very similar to the previous one:

$$\Big\{S_1, \Big\{S_1, H_0\Big\}\Big\}= \Big\{S_1, \frac{1}{\varepsilon}\Big\{S_1, H_0\Big\}_{\mathrm{g}}+\varepsilon_\delta\Big\{S_1, H_0\Big\}_{\mathrm{t}}\Big\}= \frac{1}{\varepsilon}\Big\{S_1, \tilde{V}\Big\}+\varepsilon_\delta\Big\{S_1, \Big\{S_1, H_0\Big\}_{\mathrm{t}}\Big\}. \tag{2.55}$$

The last term of Eq. [2.55] is globally of order $\varepsilon^2\varepsilon_\delta^3$ and it is neglected from further calculations. Using the equation with which we cancelled the fluctuating terms in Eq. [2.48] we continue to develop the same equation obtaining:

$$\frac{1}{\varepsilon}\Big\{S_1, \tilde{V}\Big\}= \frac{1}{\varepsilon^2}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}+\frac{1}{\varepsilon}\Big\{S_1, \tilde{V}\Big\}_{\perp}+\frac{\varepsilon_\delta}{\varepsilon}\Big\{S_1, \tilde{V}\Big\}_{\parallel}+\frac{\varepsilon_\delta}{\varepsilon}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{t}}. \tag{2.56}$$

The last two terms of Eq. [2.56] are globally of order $\varepsilon\varepsilon_\delta^3$ and are neglected from further calculations while the second term is globally of order $\varepsilon\varepsilon_\delta^2$ and it will be considered in the third order calculations. Using the last relations we just have obtained the resulting second order is:

$$\varepsilon_\delta^2\Big[-\Big\{S_1,V\Big\}_{\rm g}-\Big\{S_2,H_0\Big\}_{\rm g}+\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}\Big].$$

To annul the fluctuations one has to satisfy the consequent equation:

$$\Big\{S_1,V\Big\}_{\rm g}+\Big\{S_2,H_0\Big\}_{\rm g}-\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}=0. \tag{2.57}$$

As before it is possible to write $V=\langle V\rangle_\theta+\tilde{V}$ to obtain:

$$\begin{aligned}\Big\{S_1,\langle V\rangle_\theta\Big\}_{\rm g}+\Big\{S_1,\tilde{V}\Big\}_{\rm g}+\Big\{S_2,H_0\Big\}_{\rm g}-\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}=\\ =\Big\{S_1,\langle V\rangle_\theta\Big\}_{\rm g}+\Big\{S_2,H_0\Big\}_{\rm g}+\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}.\end{aligned} \tag{2.58}$$

It is now possible to write $\Big\{S_1,\tilde{V}\Big\}_{\rm g}$ in average and fluctuating part:

$$\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}=\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}^{(0)}+\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}^{(\sim)}. \tag{2.59}$$

One can now indirectly define S_2 in order to cancel the fluctuations from Eq. [2.58, it is necessary to satisfy the following equation:

$$\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}^{(\sim)}+\Big\{S_1,\langle V\rangle_\theta\Big\}_{\rm g}+\Big\{S_2,H_0\Big\}_{\rm g}=0. \tag{2.60}$$

The only remaining term from the second order is: $\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}^{(0)}$. The expression of the new Hamiltonian can be written as:

$$H'=H_0+\varepsilon_\delta\langle V\rangle_\theta-\varepsilon_\delta^2\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}^{(0)}. \tag{2.61}$$

The expression of the first average is known:

$$\langle V\rangle_\theta=\frac{q}{2\pi}\int_0^{2\pi}\phi(x+\rho\cos\theta,y-\rho\sin\theta,t)\,d\theta.$$

It is necessary to derive an expression for the second-order term:

$$\begin{aligned}H^{(\rm II)}=-\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\rm g}^{(0)}=+\frac{1}{2}\Big\{\tilde{V},S_1\Big\}_{\rm g}^{(0)}=+\frac{q}{4\pi m}\int_0^{2\pi}\Big(\frac{\partial\tilde{V}}{\partial\theta}\frac{\partial S_1}{\partial\mu}-\frac{\partial S_1}{\partial\theta}\frac{\partial\tilde{V}}{\partial\mu}\Big)\,d\theta=\\ =-\frac{q}{4\pi m}\int_0^{2\pi}\Big[\frac{\partial}{\partial\mu}\Big(\tilde{V}\frac{\partial S_1}{\partial\theta}\Big)-\frac{\partial}{\partial\theta}\Big(\tilde{V}\frac{\partial S_1}{\partial\mu}\Big)\Big]\,d\theta.\end{aligned}$$

Knowing the validity of $\int_0^{2\pi} \frac{\partial}{\partial\theta}\left(\frac{\partial S_1}{\partial\mu}\right) d\theta = 0$ and that $\frac{\partial S_1}{\partial\theta} = \frac{m}{qB}\tilde{V}$ it is possible to write:

$$H^{(\mathrm{II})} = -\frac{1}{2}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(0)} = -\frac{q}{4\pi m}\frac{m}{qB}\int_0^{2\pi}\frac{\partial}{\partial\mu}\tilde{V}^2\, d\theta = -\frac{1}{2B}\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle_\theta. \tag{2.62}$$

Using the definition $\tilde{V} = q\phi$ it is now possible to write the final form of the new Hamiltonian approximated to the second order:

$$H' = H_0 + \varepsilon_\delta q\langle\phi\rangle_\theta - \varepsilon_\delta^2 \frac{1}{2}\frac{q}{B}\frac{\partial}{\partial\mu}\langle\tilde{\phi}^2\rangle_\theta. \tag{2.63}$$

It is possible to define the so-called real parameter:

$$\eta = \frac{m}{2qB}. \tag{2.64}$$

so that the equation becomes:

$$H' = H_0 + \varepsilon_\delta q\langle\phi\rangle_\theta - \varepsilon_\delta^2 \frac{\eta\, q^2}{m}\frac{\partial}{\partial\mu}\langle\tilde{\phi}^2\rangle_\theta.$$

This last equation has been also been derived using the Poisson Bracket containing $\frac{\partial}{\partial\rho}$ derivatives instead of $\frac{\partial}{\partial\mu}$, it has been checked that the equations are analogues. In conclusion, θ is no more variable of the Hamiltonian although the fact it contains a term averaged over θ itself. The number of variables is therefore reduced by one. It is necessary to check if the evolution of other variables has been influenced by this procedure. For instance, it is possible to study the evolution in time of μ by simply applying:

$$\dot{\mu} = \Big\{\mu, H\Big\} = \frac{q}{m}\Big(\frac{\partial\mu}{\partial\theta}\frac{\partial H}{\partial\mu} - 1\frac{\partial H}{\partial\theta}\Big) = 0. \tag{2.65}$$

The following expression is valid:

$$\frac{\partial\mu}{\partial\theta} = 0\ ; \quad \frac{\partial H}{\partial\theta} = 0.$$

In fact, the dependence on θ has just been eliminated from the Hamiltonian. If $\dot{\mu} = 0$ then it is constant and it can be fixed as a parameter. The number of independent variables has just been further reduced by one. The resulting order of the system is 4, we recall that considering the autonomization of the system the initial order of the system was 8, this is the real strength of the gyro-kinetic approach. We have also to recall that for the considered model, it is not necessary to study the motion of particles along $\hat{\mathbf{z}}$ because of the considered geometry, this is already associated with a reduction of order 2 of the system, the gyrokinetic framework reduces the order of system by further 2 units, the whole total reduction of the order is not associated only to the use of the guiding centre approach. The contribution $H^{(\mathrm{II})}$ has been derived with ρ as an independent variable, it can be obtained with a simple change of variable, later in the procedure, it will be very useful given that the equations implemented in the code are derived from this last expression of the Hamiltonian. Therefore in the following equation, it has been reported the above-mentioned expression of the second-order term:

$$H^{(\mathrm{II})} = -\varepsilon_\delta^2 \frac{\eta\, q^2}{m}\frac{\partial}{\partial\mu}\langle\tilde{\phi}^2\rangle_\theta = -\varepsilon_\delta^2 \frac{q\eta}{B}\frac{1}{\rho}\frac{\partial}{\partial\rho}\langle\tilde{\phi}^2\rangle_\theta. \tag{2.66}$$

Naturally one has to remember that while turning the independent variable μ into ρ other contributions will have a different expression in H_0 but they will be constant as well, this won't influence at all the equations.

2.7.3 Third order

From Order 2 we have three residual terms of order $\varepsilon\varepsilon_\delta^2$, we should account for them in the third order:

$$-\varepsilon\varepsilon_\delta^2\Big[\Big\{S_1, V\Big\}_\perp - \frac{1}{2}\Big\{S_1, \tilde{V}\Big\}_\perp + \Big\{S_1, H_0\Big\}_\mathrm{t}\Big] = -\varepsilon\varepsilon_\delta^2\Big[\Big\{S_1, V\Big\}_\perp - \frac{1}{2}\Big\{S_1, \tilde{V}\Big\}_\perp + \frac{\partial S_1}{\partial t}\Big].$$

It is time now to develop the third order in ε_δ, the next equation is the full expression:

$$\begin{aligned}\varepsilon_\delta^3\Big[&-\varepsilon\Big\{S_3, H_0\Big\} - \varepsilon\Big\{S_2, V\Big\} + \frac{\varepsilon^2}{2}\Big\{S_2, \Big\{S_1, H_0\Big\}\Big\} + \frac{\varepsilon^2}{2}\Big\{S_1, \Big\{S_2, H_0\Big\}\Big\} + \\ &+ \frac{\varepsilon^2}{2}\Big\{S_1, \Big\{S_1, V\Big\}\Big\} - \frac{\varepsilon^3}{6}\Big\{S_1, \Big\{S_1, \Big\{S_1, H_0\Big\}\Big\}\Big\} + \\ &- \frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1, V\Big\}_\perp + \frac{\varepsilon}{2\,\varepsilon_\delta}\Big\{S_1, \tilde{V}\Big\}_\perp - \frac{\varepsilon}{\varepsilon_\delta}\frac{\partial S_1}{\partial t}\Big].\end{aligned} \tag{2.67}$$

through the usual property of H_0 and given that all the Poisson Brackets can be developed as in Eq. [2.40], a lot of terms are cancelled because they are of order $\varepsilon\varepsilon_\delta^3$ or higher. It is possible also to take advantage of the perturbation theory as we did in the previous subsections obtaining:

$$\begin{aligned}\varepsilon_\delta^3\Big[&-\Big\{S_3, H_0\Big\}_\mathrm{g} - \Big\{S_2, \langle V\rangle\Big\}_\mathrm{g} - \Big\{S_2, \tilde{V}\Big\}_\mathrm{g} + \frac{1}{2}\Big\{S_2, \tilde{V}\Big\}_\mathrm{g} + \frac{1}{2}\Big\{S_1, \Big\{S_2, H_0\Big\}_\mathrm{g}\Big\}_\mathrm{g} + \\ &+ \frac{1}{2}\Big\{S_1, \Big\{S_1, \langle V\rangle\Big\}_\mathrm{g}\Big\}_\mathrm{g} + \frac{1}{2}\Big\{S_1, \Big\{S_1, \tilde{V}\Big\}_\mathrm{g}\Big\}_\mathrm{g} - \frac{1}{6}\Big\{S_1, \Big\{S_1, \tilde{V}\Big\}_\mathrm{g}\Big\}_\mathrm{g} + \\ &- \frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1, \langle V\rangle\Big\}_\perp - \frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1, \tilde{V}\Big\}_\perp + \frac{\varepsilon}{2\,\varepsilon_\delta}\Big\{S_1, \tilde{V}\Big\}_\perp - \frac{\varepsilon}{\varepsilon_\delta}\frac{\partial S_1}{\partial t}\Big].\end{aligned} \tag{2.68}$$

Now it has to be considered that a generic term like $\Big\{S_1, F\Big\}$ with F a non-fluctuating term is a fluctuating term itself, therefore it will be inserted in the equation for the indirect definition of S_3 with which we will cancel all the third order fluctuating terms. Another useful relation we have already used comes from $\Big\{S_1, G\Big\}$ with G a fluctuating term, in fact, the latter can be written as:

$$\Big\{S_1, G\Big\} = \Big\{S_1, G\Big\}^{(0)} + \Big\{S_1, G\Big\}^{(\sim)}. \tag{2.69}$$

As it is known from the second order procedure also all the terms like the last one of the previous equation will be inserted in the equation to cancel the fluctuations, this is associated with a decrease in the number of the terms, of which expression is required to be calculated. By applying these relationships to the single PB and adding up similar terms it is possible to obtain:

$$\begin{aligned}
&\varepsilon_\delta^3\Big[-\Big\{S_3,H_0\Big\}_{\mathrm{g}}-\Big\{S_2,\langle V\rangle\Big\}_{\mathrm{g}}-\frac{1}{2}\Big\{S_2,\tilde{V}\Big\}_{\mathrm{g}}^{(0)}-\frac{1}{2}\Big\{S_2,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}+\frac{1}{2}\Big\{S_1,\Big\{S_2,H_0\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}+\\
&+\frac{1}{2}\Big\{S_1,\Big\{S_1,\langle V\rangle\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}+\frac{1}{3}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(0)}\Big\}_{\mathrm{g}}+\frac{1}{3}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}-\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\langle V\rangle\Big\}_{\perp}+\\
&-\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\tilde{V}\Big\}_{\perp}^{(0)}-\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\tilde{V}\Big\}_{\perp}^{(\sim)}-\frac{\varepsilon}{\varepsilon_\delta}\frac{\partial S_1}{\partial t}\Big].
\end{aligned}\tag{2.70}$$

It is necessary to write more terms with the using of the specific relation from Eq. [2.69]:

$$\begin{aligned}
&\varepsilon_\delta^3\Big[-\Big\{S_3,H_0\Big\}_{\mathrm{g}}-\Big\{S_2,\langle V\rangle\Big\}_{\mathrm{g}}-\frac{1}{2}\Big\{S_2,\tilde{V}\Big\}_{\mathrm{g}}^{(0)}-\frac{1}{2}\Big\{S_2,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}+\frac{1}{2}\Big\{S_1,\Big\{S_2,H_0\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(0)}+\\
&+\frac{1}{2}\Big\{S_1,\Big\{S_2,H_0\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(\sim)}+\frac{1}{2}\Big\{S_1,\Big\{S_1,\langle V\rangle\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(0)}+\frac{1}{2}\Big\{S_1,\Big\{S_1,\langle V\rangle\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(\sim)}+\frac{1}{3}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(0)}\Big\}_{\mathrm{g}}+\\
&+\frac{1}{3}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}^{(0)}+\frac{1}{3}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}^{(\sim)}-\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\langle V\rangle\Big\}_{\perp}+\\
&-\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\tilde{V}\Big\}_{\perp}^{(0)}-\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\tilde{V}\Big\}_{\perp}^{(\sim)}-\frac{\varepsilon}{\varepsilon_\delta}\frac{\partial S_1}{\partial t}\Big].
\end{aligned}\tag{2.71}$$

I can cancel the fluctuating terms with a specific definition of S_3 through the following equation:

$$\begin{aligned}
&-\Big\{S_3,H_0\Big\}_{\mathrm{g}}-\Big\{S_2,\langle V\rangle\Big\}_{\mathrm{g}}-\frac{1}{2}\Big\{S_2,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}+\frac{1}{2}\Big\{S_1,\Big\{S_2,H_0\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(\sim)}+\frac{1}{2}\Big\{S_1,\Big\{S_1,\langle V\rangle\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(\sim)}+\\
&+\frac{1}{3}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(0)}\Big\}_{\mathrm{g}}+\frac{1}{3}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}^{(\sim)}-\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\langle V\rangle\Big\}_{\perp}+\\
&-\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\tilde{V}\Big\}_{\perp}^{(\sim)}-\frac{\varepsilon}{\varepsilon_\delta}\frac{\partial S_1}{\partial t}=0.
\end{aligned}\tag{2.72}$$

In fact, as we said $\Big\{S_1,F\Big\}$ with F a non-fluctuating term is a fluctuating term itself. The remaining terms in the ε_δ third order are:

$$\begin{aligned}
&\varepsilon_\delta^3\Big[-\frac{1}{2}\Big\{S_2,\tilde{V}\Big\}_{\mathrm{g}}^{(0)}+\frac{1}{2}\Big\{S_1,\Big\{S_2,H_0\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(0)}+\frac{1}{2}\Big\{S_1,\Big\{S_1,\langle V\rangle\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(0)}+\\
&+\frac{1}{3}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}^{(0)}-\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\tilde{V}\Big\}_{\perp}^{(0)}\Big].
\end{aligned}\tag{2.73}$$

Like it has been done with the term of the second order it is necessary to develop those terms, they can't be implemented otherwise. It is possible to start developing the various terms after putting the equation in order:

$$\varepsilon_\delta^3H^{(\mathrm{III})}=\varepsilon_\delta^3\Big[H_1^{(\mathrm{III})}+H_2^{(\mathrm{III})}+H_3^{(\mathrm{III})}+H_4^{(\mathrm{III})}+H_5^{(\mathrm{III})}\Big].\tag{2.74}$$

The first term of Eq. [2.73] we are trying to develop is $H_1^{(\mathrm{III})}$:

$$H_1^{(\mathrm{III})} = -\frac{1}{2}\Big\{S_2, \tilde{V}\Big\}_{\mathrm{g}}^{(0)}. \tag{2.75}$$

Starting from the inner it is possible to simply develop the gyro-bracket, and apply the identity that has been applied for the second order:

$$\Big\{S_2, \tilde{V}\Big\}_{\mathrm{g}} = -\Big\{\tilde{V}, S_2\Big\}_{\mathrm{g}} = -\frac{q}{m}\Big(\frac{\partial \tilde{V}}{\partial \theta}\frac{\partial S_2}{\partial \mu} - \frac{\partial \tilde{V}}{\partial \mu}\frac{\partial S_2}{\partial \theta}\Big) = -\frac{q}{m}\Big[\frac{\partial}{\partial \mu}\Big(\tilde{V}\frac{\partial S_2}{\partial \theta}\Big) - \frac{\partial}{\partial \theta}\Big(\tilde{V}\frac{\partial S_2}{\partial \mu}\Big)\Big]. \tag{2.76}$$

Taking advantage of the gyro-average it is possible to use $\int \frac{\partial}{\partial \theta}\Big(\square\Big)d\theta = 0$ so that it is possible to write:

$$-\frac{1}{2}\Big\{S_2, \tilde{V}\Big\}_{\mathrm{g}}^{(0)} = \frac{1}{2}\frac{q}{m}\frac{\partial}{\partial \mu}\Big(\tilde{V}\frac{\partial S_2}{\partial \theta}\Big)^{(0)}. \tag{2.77}$$

The expression of $\frac{\partial S_2}{\partial \theta}$ is required, the only way to obtain it is from the Eq. [2.57] we introduced the latter for the indirect definition of S_2 while calculating the expression of the second order term. Starting from that equation we write:

$$\Big\{S_2, H_0\Big\}_{\mathrm{g}} = -\frac{1}{2}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)} - \Big\{S_1, \langle V\rangle_\theta\Big\}_{\mathrm{g}}. \tag{2.78}$$

Starting from the first component of the right-hand side of Eq. [2.78] it is possible to write:

$$\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)} = \Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}} - \Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(0)}. \tag{2.79}$$

We already calculate the second term of the RHS of Eq. [2.79]:

$$\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(0)} = -\frac{1}{B}\frac{\partial}{\partial \mu}\langle \tilde{V}^2\rangle_\theta. \tag{2.80}$$

The other needs to be calculated with the usual procedure, developing the gyro-brackets, and applying the same identity we used before but not having the average, in this case, will complicate things:

$$\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}} = -\Big\{\tilde{V}, S_1\Big\}_{\mathrm{g}} = -\frac{q}{m}\Big(\frac{\partial \tilde{V}}{\partial \theta}\frac{\partial S_1}{\partial \mu} - \frac{\partial \tilde{V}}{\partial \mu}\frac{\partial S_1}{\partial \theta}\Big) = -\frac{q}{m}\Big[\frac{\partial}{\partial \mu}\Big(\tilde{V}\frac{\partial S_1}{\partial \theta}\Big) - \frac{\partial}{\partial \theta}\Big(\tilde{V}\frac{\partial S_1}{\partial \mu}\Big)\Big]. \tag{2.81}$$

It is known the expression of $\frac{\partial S_1}{\partial \theta}$ from Eq. [2.48] and it can be easily calculate the expression of $\frac{\partial S_1}{\partial \mu}$ given the definition of S_1 coming from Eq. [2.49]:

$$\frac{\partial S_1}{\partial \mu} = \frac{m}{qB}\frac{\partial}{\partial \mu}\int \tilde{V} d\theta. \tag{2.82}$$

It is then possible to calculate the first term of the RHS of Eq. [2.79]:

$$\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}} = -\frac{1}{B}\Big[\frac{\partial}{\partial \mu}\Big(\tilde{V}^2\Big) - \frac{\partial}{\partial \theta}\Big(\tilde{V}\frac{\partial}{\partial \mu}\int \tilde{V} d\theta\Big)\Big]. \tag{2.83}$$

The full expression of Eq. [2.79] is:

$$\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}=-\frac{1}{B}\Big[\frac{\partial}{\partial\mu}\Big(\tilde{V}^2\Big)-\frac{\partial}{\partial\theta}\Big(\tilde{V}\frac{\partial}{\partial\mu}\int\tilde{V}\,d\theta\Big)-\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle_\theta\Big]. \tag{2.84}$$

It is now possible to develop the second component of the right-hand side of Eq. [2.78]:

$$\Big\{S_1,\langle V\rangle\Big\}_{\mathrm{g}}=\frac{q}{m}\Big(\frac{\partial S_1}{\partial\theta}\frac{\partial\langle V\rangle}{\partial\mu}-\frac{\partial S_1}{\partial\mu}\frac{\partial\langle V\rangle}{\partial\theta}\Big)=\frac{q}{m}\Big(\frac{\partial S_1}{\partial\theta}\frac{\partial\langle V\rangle}{\partial\mu}\Big). \tag{2.85}$$

Giving that $\frac{\partial\langle V\rangle}{\partial\theta}=0$. It is known the expression of $\frac{\partial S_1}{\partial\theta}$ from Eq. [2.48] and it is possible to substitute it obtaining:

$$\Big\{S_1,\langle V\rangle\Big\}_{\mathrm{g}}=\frac{1}{B}\tilde{V}\frac{\partial\langle V\rangle}{\partial\mu}. \tag{2.86}$$

It is possible to join the two components coming from Eq. [2.86] and from Eq. [2.84] obtaining the expression of the RHS of Eq. [2.78]:

$$\begin{aligned}\Big\{S_2,H_0\Big\}_{\mathrm{g}}&=-\frac{1}{2}\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}-\Big\{S_1,\langle V\rangle_\theta\Big\}_{\mathrm{g}}=\\&=-\frac{1}{B}\tilde{V}\frac{\partial\langle V\rangle}{\partial\mu}+\frac{1}{2}\frac{1}{B}\Big[\frac{\partial}{\partial\mu}\Big(\tilde{V}^2\Big)-\frac{\partial}{\partial\theta}\Big(\tilde{V}\frac{\partial}{\partial\mu}\int\tilde{V}\,d\theta\Big)-\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle_\theta\Big].\end{aligned} \tag{2.87}$$

Now it is required to develop $\Big\{S_2,H_0\Big\}_{\mathrm{g}}$ itself, in fact, we are able to obtain the expression of $\frac{\partial S_2}{\partial\theta}$ using also the Eq. [2.87]. We start with the usual development:

$$\Big\{S_2,H_0\Big\}_{\mathrm{g}}=\frac{q}{m}\Big(\frac{\partial S_2}{\partial\theta}\frac{\partial H_0}{\partial\mu}-\frac{\partial S_2}{\partial\mu}\frac{\partial H_0}{\partial\theta}\Big)=\frac{q}{m}\Big(\frac{\partial S_2}{\partial\theta}\frac{\partial H_0}{\partial\mu}\Big)=\frac{qB}{m}\Big(\frac{\partial S_2}{\partial\theta}\Big). \tag{2.88}$$

Giving that $\frac{\partial H_0}{\partial\theta}=0$ and $\frac{\partial H_0}{\partial\mu}=B$. As it has been said now we are able to obtain the explicit expression of $\frac{\partial S_2}{\partial\theta}$ taking advantage of Eq. [2.87]:

$$\frac{\partial S_2}{\partial\theta}=\frac{m}{qB^2}\Big[-\tilde{V}\frac{\partial\langle V\rangle}{\partial\mu}+\frac{1}{2}\frac{\partial}{\partial\mu}\Big(\tilde{V}^2\Big)-\frac{1}{2}\frac{\partial}{\partial\theta}\Big(\tilde{V}\frac{\partial}{\partial\mu}\int\tilde{V}\,d\theta\Big)-\frac{1}{2}\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle_\theta\Big]. \tag{2.89}$$

Now we want to find the final expression of $H_1^{(\mathrm{III})}$ substituting Eq. [2.89] in Eq. [2.77] and by applying the gyro-average in order to simplify the expression by cancelling the terms containing $\frac{\partial}{\partial\theta}$:

$$-\frac{1}{2}\Big\{S_2,\tilde{V}\Big\}_{\mathrm{g}}^{(0)}=\frac{1}{2}\frac{q}{m}\frac{\partial}{\partial\mu}\Big(\tilde{V}\frac{\partial S_2}{\partial\theta}\Big)^{(0)}=\frac{1}{2B^2}\frac{\partial}{\partial\mu}\Big(-\tilde{V}^2\frac{\partial\langle V\rangle}{\partial\mu}+\frac{1}{2}\tilde{V}\frac{\partial\tilde{V}^2}{\partial\mu}-\frac{1}{2}\tilde{V}\frac{\partial\langle\tilde{V}^2\rangle_\theta}{\partial\mu}\Big)^{(0)}. \tag{2.90}$$

The average on theta can enter into the derivatives in Eq. [2.90] giving us the final expression:

$$H_1^{(\mathrm{III})}=\frac{1}{2B^2}\frac{\partial}{\partial\mu}\Big(-\langle\tilde{V}^2\rangle\frac{\partial\langle V\rangle}{\partial\mu}+\frac{1}{2}\langle\tilde{V}\rangle\frac{\partial\langle\tilde{V}^2\rangle}{\partial\mu}-\frac{1}{2}\langle\tilde{V}\rangle\frac{\partial\langle\tilde{V}^2\rangle}{\partial\mu}\Big). \tag{2.91}$$

It is time to develop $H_2^{(\mathrm{III})}$ from Eq. [2.73];

$$H_2^{(\mathrm{III})} = +\frac{1}{2}\Big\{S_1, \Big\{S_2, H_0\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(0)}. \tag{2.92}$$

It is necessary to take advantage of the Eq. [2.57] through which it has been defined indirectly S_2:

$$\Big\{S_2, H_0\Big\}_{\mathrm{g}} = -\frac{1}{2}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)} - \Big\{S_1, \langle V\rangle_\theta\Big\}_{\mathrm{g}}.$$

We substitute Eq. [2.57] in Eq. [2.92] obtaining two contributions:

$$\frac{1}{2}\Big\{S_1, \Big\{S_2, H_0\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(0)} = -\frac{1}{4}\Big\{S_1, \Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}^{(0)} - \frac{1}{2}\Big\{S_1, \Big\{S1, \langle V\rangle_\theta\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(0)}. \tag{2.93}$$

We notice that the second component of the RHS of Eq. [2.93]is cancelled by the third term of the third order equation (Eq. [2.73]). In fact we notice the expression of $H_3^{(\mathrm{III})}$:

$$H_3^{(\mathrm{III})} = +\frac{1}{2}\Big\{S_1, \Big\{S_1, \langle V\rangle\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}^{(0)}. \tag{2.94}$$

This means that developing only the first component of the RHS of Eq. [2.93] gives us the expression of $H_2^{(\mathrm{III})} + H_3^{(\mathrm{III})}$:

$$H_2^{(\mathrm{III})} + H_3^{(\mathrm{III})} = -\frac{1}{4}\Big\{S_1, \Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}^{(0)}. \tag{2.95}$$

Like we did before we try to develop Eq. [2.95] starting from the inner part:

$$\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)} = \Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}} - \Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(0)}. \tag{2.96}$$

The expression of component $\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(0)}$ of Eq. [2.96] is known from Eq. [2.62]:

$$\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)} = \Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}} + \frac{1}{B}\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle. \tag{2.97}$$

It is necessary to calculate the first component of the RHS of Eq. [2.97] through the usual procedure:

$$\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}} = \frac{q}{m}\Big(\frac{\partial S_1}{\partial\theta}\frac{\partial\tilde{V}}{\partial\mu} - \frac{\partial S_1}{\partial\mu}\frac{\partial\tilde{V}}{\partial\theta}\Big) = -\frac{q}{m}\Big(\frac{\partial S_1}{\partial\mu}\frac{\partial\tilde{V}}{\partial\theta} - \frac{\partial S_1}{\partial\theta}\frac{\partial\tilde{V}}{\partial\mu}\Big) = -\frac{q}{m}\Big(\frac{\partial}{\partial\mu}\Big(\tilde{V}\frac{\partial S_1}{\partial\theta}\Big) + - \frac{\partial}{\partial\theta}\Big(\tilde{V}\frac{\partial S_1}{\partial\mu}\Big)\Big) = -\frac{1}{B}\frac{\partial}{\partial\mu}\tilde{V}^2 + \frac{q}{m}\frac{\partial}{\partial\theta}\Big(\tilde{V}\frac{\partial S_1}{\partial\mu}\Big). \tag{2.98}$$

The final expression of the Eq. [2.97] then is:

$$\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)} = \frac{1}{B}\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle - \frac{1}{B}\frac{\partial}{\partial\mu}\tilde{V}^2 + \frac{q}{m}\frac{\partial}{\partial\theta}\Big(\tilde{V}\frac{\partial S_1}{\partial\mu}\Big). \tag{2.99}$$

It is necessary to apply the second gyro-bracket to the expression of the Eq. [2.99] in order to calculate $\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}$. By taking advantage of the usual identity, of the definition of $\frac{\partial S_1}{\partial\theta}$, of the gyro-average and the periodic boundary conditions it is possible to obtain an intermediate expression of $H_2^{(\mathrm{III})}+H_3^{(\mathrm{III})}$:

$$H_2^{(\mathrm{III})}+H_3^{(\mathrm{III})}=-\frac{1}{4}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}^{(0)}=\frac{1}{2\pi B}\int_0^{2\pi}\frac{\partial}{\partial\mu}\Big(\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\tilde{V}\Big)d\theta. \tag{2.100}$$

Making explicit the term inside the integral with the substitution of Eq. [2.99] and by applying the average it is possible to obtain:

$$H_2^{(\mathrm{III})}+H_3^{(\mathrm{III})}=-\frac{1}{B}\frac{\partial}{\partial\mu}\Big(\frac{1}{B}\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle\langle\tilde{V}\rangle-\frac{1}{B}\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle\langle\tilde{V}\rangle+\frac{q}{2\pi\ m}\int_0^{2\pi}\frac{\partial}{\partial\theta}\Big(\tilde{V}^2\frac{\partial S_1}{\partial\mu}\Big)d\theta\Big)=0. \tag{2.101}$$

So the conclusion is that they don't give any contribution to the new expression of the Hamiltonian:

$$H_2^{(\mathrm{III})}+H_3^{(\mathrm{III})}=0. \tag{2.102}$$

The next term we are trying to develop is $H_4^{(\mathrm{III})}$:

$$H_4^{(\mathrm{III})}=\frac{1}{3}\Big\{S_1,\Big\{S_1,\tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}\Big\}_{\mathrm{g}}^{(0)}. \tag{2.103}$$

We have already developed this term following the same procedure of $H_2^{(\mathrm{III})}+H_3^{(\mathrm{III})}$, the final result is taken by Eq. [2.100] and Eq. [2.40] giving that the final result is zero:

$$H_4^{(\mathrm{III})}=0. \tag{2.104}$$

Now only $H_5^{(\mathrm{III})}$ is missing, this is the one of order $\varepsilon\varepsilon_\delta^2$, the latter is different from the others given that it doesn't have the gyro-brackets acting on it but only the "perpendicular-brackets", the developing procedure will be slightly different from the previous ones. As usual, we start with the expression of the term itself:

$$H_5^{(\mathrm{III})}=-\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\tilde{V}\Big\}_{\perp}^{(0)}. \tag{2.105}$$

We develop the "perpendicular brackets" following the expression introduced in Sec.2.6:

$$\Big\{F,G\Big\}_{\perp}=\Big\{F,G\Big\}_{(\chi_1,\chi_2)}=-\frac{1}{qB}\Big(\frac{\partial F}{\partial\chi_1}\frac{\partial G}{\partial\chi_2}-\frac{\partial F}{\partial\chi_2}\frac{\partial G}{\partial\chi_1}\Big). \tag{2.106}$$

Using Eq. [2.106] it is possible to obtain:

$$\Big\{S_1,\tilde{V}\Big\}_{\perp}=-\frac{1}{qB}\Big(\frac{\partial S_1}{\partial\chi_1}\frac{\partial\tilde{V}}{\partial\chi_2}-\frac{\partial S_1}{\partial\chi_2}\frac{\partial\tilde{V}}{\partial\chi_1}\Big). \tag{2.107}$$

The expression of S_1 is known from Eq. [2.49], and it is possible to calculate the various partial derivatives:

$$\frac{\partial S_1}{\partial\chi_1}=\frac{m}{qB}\int\frac{\partial\tilde{V}}{\partial\chi_1}d\theta\ ;\qquad\frac{\partial S_1}{\partial\chi_2}=\frac{m}{qB}\int\frac{\partial\tilde{V}}{\partial\chi_2}d\theta\ ;$$

It is possible to substitute these last expressions in Eq. [2.107]:

$$\Big\{S_1,\tilde{V}\Big\}_{\perp}=-\frac{m}{q^2B^2}\Big(\frac{\partial\tilde{V}}{\partial\chi_2}\int\frac{\partial\tilde{V}}{\partial\chi_1}d\theta-\frac{\partial\tilde{V}}{\partial\chi_1}\int\frac{\partial\tilde{V}}{\partial\chi_2}d\theta\Big). \tag{2.108}$$

The expression of this term then should be the following:

$$H_5^{(\mathrm{III})}=-\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\Big\{S_1,\tilde{V}\Big\}_{\perp}^{(0)}=\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\frac{m}{q^2B^2}\Big\langle\Big(\frac{\partial\tilde{V}}{\partial\chi_2}\int\frac{\partial\tilde{V}}{\partial\chi_1}d\theta-\frac{\partial\tilde{V}}{\partial\chi_1}\int\frac{\partial\tilde{V}}{\partial\chi_2}d\theta\Big)\Big\rangle. \tag{2.109}$$

We have to concentrate on the terms inside the average, the only way to annul this average is:

$$\Big\langle\frac{\partial\tilde{V}}{\partial\chi_2}\int\frac{\partial\tilde{V}}{\partial\chi_1}d\theta\Big\rangle=\Big\langle\frac{\partial\tilde{V}}{\partial\chi_2}\frac{\partial}{\partial\chi_1}\int\tilde{V}d\theta\Big\rangle. \tag{2.110}$$

Now it shall be done a little recap of all the terms that have been calculated:

$$\varepsilon_\delta^3H^{(\mathrm{III})}=\varepsilon_\delta^3\Big[H_1^{(\mathrm{III})}+H_2^{(\mathrm{III})}+H_3^{(\mathrm{III})}+H_4^{(\mathrm{III})}+H_5^{(\mathrm{III})}\Big].$$

$$H_1^{(\mathrm{III})}=\frac{1}{2B^2}\frac{\partial}{\partial\mu}\Big(-\langle\tilde{V}^2\rangle\frac{\partial\langle V\rangle}{\partial\mu}+\frac{1}{2}\langle\tilde{V}\rangle\frac{\partial\langle\tilde{V}^2\rangle}{\partial\mu}-\frac{1}{2}\langle\tilde{V}\rangle\frac{\partial\langle\tilde{V}^2\rangle}{\partial\mu}\Big).$$

$$H_2^{(\mathrm{III})}+H_3^{(\mathrm{III})}=0.$$

$$H_4^{(\mathrm{III})}=0.$$

$$H_5^{(\mathrm{III})}=\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\frac{m}{q^2B^2}\Big(\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_2}\int\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_1}d\theta-\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_1}\int\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_2}d\theta\Big).$$

The final expression of the third-order contribution to the modified expression of the Hamiltonian is:

$$\begin{aligned}\varepsilon_\delta^3H^{(\mathrm{III})}=&\varepsilon_\delta^3\frac{1}{2B^2}\Big[\frac{\partial}{\partial\mu}\Big(-\langle\tilde{V}^2\rangle\frac{\partial\langle V\rangle}{\partial\mu}+\frac{1}{2}\langle\tilde{V}\rangle\frac{\partial\langle\tilde{V}^2\rangle}{\partial\mu}-\frac{1}{2}\langle\tilde{V}\rangle\frac{\partial\langle\tilde{V}^2\rangle}{\partial\mu}\Big)+\\&+\frac{\varepsilon}{\varepsilon_\delta}\frac{m}{q^2}\Big(\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_2}\int\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_1}d\theta-\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_1}\int\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_2}d\theta\Big)\Big].\end{aligned} \tag{2.111}$$

Some terms cancel each other:

$$\varepsilon_\delta^3H^{(\mathrm{III})}=\varepsilon_\delta^3\Big[\frac{1}{2B^2}\frac{\partial}{\partial\mu}\Big(-\langle\tilde{V}^2\rangle\frac{\partial\langle V\rangle}{\partial\mu}\Big)+\frac{1}{2}\frac{\varepsilon}{\varepsilon_\delta}\frac{m}{q^2B^2}\Big(\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_2}\int\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_1}d\theta-\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_1}\int\frac{\partial\langle\tilde{V}\rangle}{\partial\chi_2}d\theta\Big)\Big]. \tag{2.112}$$

This last expression of the third-order term is not that complex to implement. The first contribution is very similar to the second order term, it is just necessary to apply it a second time the derivative, the current version has μ as an independent variable but when it will be implemented μ will be turned in ρ like what happened $H^{(\mathrm{II})}$. On the other hand, the second contribution is more difficult to implement given that it is necessary to compute the primitive of $\langle\tilde{\phi}\rangle$. It is possible to make a quick recap of the final full expression of the modified Hamiltonian with μ as the independent variable (later will be reported the expression in ρ), the notation helps us differentiate the origin of each term:

$$H' = H^{(0)} + H^{(\mathrm{I})} + H^{(\mathrm{II})} + H^{(\mathrm{III})}. \tag{2.113}$$

$$H^{(0)} = H_0 = \mu B + \frac{1}{2} m \boldsymbol{v}_{\|}^2 + k. \tag{2.114}$$

$$H^{(\mathrm{I})} = \varepsilon_\delta q \langle \phi \rangle_\theta. \tag{2.115}$$

$$H^{(\mathrm{II})} = -\varepsilon_\delta^2 \frac{\eta\, q^2}{m} \frac{\partial}{\partial \mu} \langle \tilde{\phi}^2 \rangle_\theta. \tag{2.116}$$

$$H^{(\mathrm{III})} = \varepsilon_\delta^3 \Big[-\frac{q^3}{2B^2} \frac{\partial}{\partial \mu} \Big(\langle \tilde{\phi}^2 \rangle \frac{\partial \langle \phi \rangle}{\partial \mu} \Big) + \frac{1}{2} \frac{\varepsilon}{\varepsilon_\delta} \frac{m}{B^2} \Big(\frac{\partial \langle \tilde{\phi} \rangle}{\partial \chi_2} \int \frac{\partial \langle \tilde{\phi} \rangle}{\partial \chi_1} d\theta - \frac{\partial \langle \tilde{\phi} \rangle}{\partial \chi_1} \int \frac{\partial \langle \tilde{\phi} \rangle}{\partial \chi_2} d\theta \Big) \Big]. \tag{2.117}$$

2.8 Guiding centres governing equations

The expression of the modified Hamiltonian, as anticipated, is used to calculate the evolution in time of variables of great interest, for a generic observable we remember:

$$\dot{F} = \frac{\partial F}{\partial t} = \Big\{ F, H \Big\}. \tag{2.118}$$

This generic expression can be used in order to study the evolution in time of various quantities, and independent variables as well. Given that we are interested in the trajectories of the guiding centre themselves and that χ_1 and χ_2 coordinates are independent variables it is possible to obtain two differential equations in time on how the GC will move in space once one has set the other parameters of the system. The equation will be solved through the integration in time of the equation. Before starting the procedure used to obtain the equation for χ_1 and χ_2 coordinates one has to remember that the Poisson Bracket obtained in Eq. [2.35] is canonical, this means that by applying the PB to one of the independent variables some contributions are zero given that if we have F and G independent variables then we are sure that $\frac{\partial F}{\partial G} = \frac{\partial G}{\partial F} = 0$. In the previous section the PB has been subdivided into various contributions, it is possible to recall Eq. [2.39]:

$$\Big\{ F, G \Big\} = \frac{1}{\varepsilon} \Big\{ F, G \Big\}_{\mathrm{g}} + \Big\{ F, G \Big\}_{\perp} + \varepsilon_\delta \Big\{ F, G \Big\}_{\|} + \varepsilon_\delta \Big\{ F, G \Big\}_{\mathrm{t}}.$$

It is possible also to remind the definition of the sub-bracket of interest for obtaining these specific equations giving that the bracket will be applied to χ_1 and χ_2 coordinates:

$$\Big\{ F, G \Big\}_{\perp} = \Big\{ F, G \Big\}_{(\chi_1, \chi_2)} = -\frac{1}{qB} \Big(\frac{\partial F}{\partial \chi_1} \frac{\partial G}{\partial \chi_2} - \frac{\partial F}{\partial \chi_2} \frac{\partial G}{\partial \chi_1} \Big).$$

Before starting the procedure it is required to make some useful considerations, that's to better understand the meaning of the operations involved in the equations and to clarify the entire procedure. There are some similarities between the starting and the modified expression of the Hamiltonian:

$$H = H_0 + q \varepsilon_\delta \phi(\chi_1, \chi_2, \chi_3, \mu, \theta, \tau).$$

$$H' = H_0 + q\varepsilon_\delta \psi(\chi_1, \chi_2, \chi_3, \mu, \tau).$$

The structure is exactly the same, ψ is a modified potential and indeed it is the potential felt by the guiding centre, it is not the same potential felt by the particles, it is corrected. If we imagine studying the trajectories of the particles and guiding centre in the same conditions the final results should be a correspondence between the trajectories of gyro-kinetic GC and fully-kinetic GC (calculated through the change of variable once the trajectory of the particle itself is known). From now on ψ is defined as *Guiding Center Potential*. From what we have just said it is valid:

$$\varepsilon_\delta \psi = \varepsilon_\delta q \langle \phi \rangle_\theta - \varepsilon_\delta^2 \frac{q\eta}{B} \frac{1}{\rho} \frac{\partial}{\partial \rho} \langle \tilde{\phi}^2 \rangle_\theta. \tag{2.119}$$

The resulting potential is considering the so-called *FLR effects*, we imagine the guiding centre (non-physical object) and the particle that rotates around it, the movement of the system needs to follow the particle driven by the known physics and equations, the particle is moving very quickly and it feels the potential in multiple positions ad different instants of time, the value of the potential may be different in some of these points. The presence of the gyro-average inside ψ is the key point, it is accounting for what ϕ a particle feels during each rotation around the GC, it is clear that the final result is a mean value and naturally some peak values will be flattened by this operation. The operation turns out to be a damping, the resulting potential is a smoother version of the initial one. As we will see the entire operation will result in multiplication by a coefficient ≤ 1. This topic will be treated in greater detail later in the manuscript. Now it is possible to start the procedure to obtain the equations, let's first introduce some definitions:

$$A = \frac{\varepsilon_\delta}{B}. \tag{2.120}$$

$$\Phi = A\phi. \tag{2.121}$$

$$\Psi = A\psi. \tag{2.122}$$

It is possible to introduce A in the equations by dividing Eq. [2.119] for B, amplitude of the magnetic field:

$$\frac{\varepsilon_\delta}{B}\psi = \frac{\varepsilon_\delta}{B} q \langle \phi \rangle_\theta - \frac{\varepsilon_\delta^2}{B^2} q\eta \frac{1}{\rho} \frac{\partial}{\partial \rho} \langle \tilde{\phi}^2 \rangle_\theta. \tag{2.123}$$

We introduce the definition of A directly into the gyro averages:

$$A\psi = q\langle A\phi \rangle_\theta - q\eta \frac{1}{\rho} \frac{\partial}{\partial \rho} \langle (A\tilde{\phi})^2 \rangle_\theta. \tag{2.124}$$

By using the definition introduced with Eq. [2.121] and Eq. [2.122] the resulting equation for the guiding center potential is the following:

$$\Psi = q\langle \Phi \rangle_\theta - q\eta \frac{1}{\rho} \frac{\partial}{\partial \rho} \langle (\tilde{\Phi})^2 \rangle_\theta. \tag{2.125}$$

After having obtained this last expression for the guiding centre potential it is possible to obtain the two governing equations for the evolution of GC coordinates, as usual, it is necessary to apply the PB to the two independent variables:

$$\dot{\chi}_1 = \Big\{\chi_1, H'\Big\} = \Big\{\chi_1, H'\Big\}_{\perp} = \Big\{\chi_1, H'\Big\}_{(\chi_1,\chi_2)} = -\frac{1}{qB}\Big(\frac{\partial \chi_1}{\partial \chi_1}\frac{\partial H'}{\partial \chi_2} - \frac{\partial \chi_1}{\partial \chi_2}\frac{\partial H'}{\partial \chi_1}\Big). \tag{2.126}$$

$$\dot{\chi}_2 = \Big\{\chi_2, H'\Big\} = \Big\{\chi_2, H'\Big\}_{\perp} = \Big\{\chi_2, H'\Big\}_{(\chi_1,\chi_2)} = -\frac{1}{qB}\Big(\frac{\partial \chi_2}{\partial \chi_1}\frac{\partial H'}{\partial \chi_2} - \frac{\partial \chi_2}{\partial \chi_2}\frac{\partial H'}{\partial \chi_1}\Big). \tag{2.127}$$

Eq. [2.126] and Eq. [2.127] are simplified by the fact that $\frac{\partial \chi_1}{\partial \chi_2} = \frac{\partial \chi_2}{\partial \chi_1} = 0$, in addition to that, the only term inside H' that is dependent on χ_2 and χ_1 is Ψ indeed, the definition of the latter comes from the fact that the PBs are introducing the product of $\frac{1}{B}$ with ψ (remember Eq. [2.123]), this is what we have in reality inside H', it is immediate to obtain:

$$\dot{\chi}_1 = -\frac{1}{q}\frac{\partial \Psi}{\partial \chi_2}. \tag{2.128}$$

$$\dot{\chi}_2 = \frac{1}{q}\frac{\partial \Psi}{\partial \chi_1}. \tag{2.129}$$

By applying the definition from Eq. [2.125] it is easy to obtain:

$$\dot{\chi}_1 = -\frac{\partial}{\partial \chi_2}\Big[\langle \Phi \rangle_\theta - \eta\frac{1}{\rho}\frac{\partial}{\partial \rho}\langle (\tilde{\Phi})^2 \rangle_\theta\Big]. \tag{2.130}$$

$$\dot{\chi}_2 = \frac{\partial}{\partial \chi_1}\Big[\langle \Phi \rangle_\theta - \eta\frac{1}{\rho}\frac{\partial}{\partial \rho}\langle (\tilde{\Phi})^2 \rangle_\theta\Big]. \tag{2.131}$$

Two further definitions are now introduced:

$$\Psi_{GC_1} = \langle \Phi \rangle_\theta. \tag{2.132}$$

$$\Psi_{GC_2} = -\eta\frac{1}{\rho}\frac{\partial}{\partial \rho}\langle (\tilde{\Phi})^2 \rangle_\theta. \tag{2.133}$$

Together with the validity of the following obvious relation:

$$\Psi = q\Big[\Psi_{GC_1} + \Psi_{GC_2}\Big]. \tag{2.134}$$

It is natural that Eq. [2.128] and Eq. [2.129] become then:

$$\dot{\chi}_1 = -\frac{\partial \Psi_{GC_1}}{\partial \chi_2} - \frac{\partial \Psi_{GC_2}}{\partial \chi_2}. \tag{2.135}$$

$$\dot{\chi}_2 = +\frac{\partial \Psi_{GC_1}}{\partial \chi_1} + \frac{\partial \Psi_{GC_2}}{\partial \chi_1}. \tag{2.136}$$

2.8.1 Non-dimensional governing equations for guiding centres

In order to implement Eq. [2.135] and Eq. [2.136] it is necessary to obtain the non-dimensional version of these last ones. Once we have done that it will be possible to define 3 non-dimensional parameters inside the code ($\hat{A}$, $\hat{\eta}$ and $\hat{\rho}$) to fix physical conditions governing the movement of particles. We remember in fact that in this specific system, μ is a constant and consequently also the *Larmor Radius* ρ (check Eq. [2.28] and Eq. [2.121]). η is the so-called *real parameter* we introduced in Eq. [2.64] while A is the amplitude of the potential we introduced in Eq. [2.120] and Eq. [2.121]. It is necessary to define the usual non-dimensional parameters:

$$\begin{gathered}\hat{\chi}_{\_} = 2\pi \frac{\chi_{\_}}{L} \ ; \ \hat{t} = 2\pi \frac{t}{T} \ ; \ \hat{\phi} = \frac{\phi}{\phi_0} \ ; \ \phi_0 = \frac{L^2 B_0}{T 2\pi} \ ; \ \hat{B} = \frac{B}{B_0} \ ; \\ \hat{A} = \frac{\varepsilon_\delta}{\hat{B}} \ ; \ \hat{\rho} = 2\pi \frac{\rho}{L} \ ; \ \hat{\eta} = 2\pi \frac{\eta}{T}.\end{gathered} \tag{2.137}$$

The starting equation are Eq. [2.135] and Eq. [2.136] in which we apply the definitions from Eq. [2.132] and Eq. [2.133]:

$$\dot{\chi}_1 = -\frac{\partial \langle \Phi \rangle_\theta}{\partial \chi_2} - \frac{\partial}{\partial \chi_2}\Big(-\eta \frac{1}{\rho} \frac{\partial}{\partial \rho} \langle (\tilde{\Phi})^2 \rangle_\theta \Big). \tag{2.138}$$

$$\dot{\chi}_2 = +\frac{\partial \langle \Phi \rangle_\theta}{\partial \chi_1} + \frac{\partial}{\partial \chi_1}\Big(-\eta \frac{1}{\rho} \frac{\partial}{\partial \rho} \langle (\tilde{\Phi})^2 \rangle_\theta \Big). \tag{2.139}$$

We introduce the definitions of the non-dimensional parameters:

$$\frac{L}{T}\frac{\partial \hat{\chi}_1}{\partial \hat{t}} = -\frac{2\pi \varepsilon_\delta \phi_0}{LB}\frac{\partial \langle \hat{\phi} \rangle_\theta}{\partial \hat{\chi}_2} - \frac{(2\pi)^3 \varepsilon_\delta^2 T \phi_0^2}{L^3 2\pi B^2}\frac{\partial}{\partial \hat{\chi}_2}\Big(-\hat{\eta} \frac{1}{\hat{\rho}} \frac{\partial}{\partial \hat{\rho}} \langle (\tilde{\hat{\phi}}^2) \rangle_\theta \Big). \tag{2.140}$$

$$\frac{L}{T}\frac{\partial \hat{\chi}_2}{\partial \hat{t}} = +\frac{2\pi \varepsilon_\delta \phi_0}{LB}\frac{\partial \langle \hat{\phi} \rangle_\theta}{\partial \hat{\chi}_1} + \frac{(2\pi)^3 \varepsilon_\delta^2 T \phi_0^2}{L^3 2\pi B^2}\frac{\partial}{\partial \hat{\chi}_1}\Big(-\hat{\eta} \frac{1}{\hat{\rho}} \frac{\partial}{\partial \hat{\rho}} \langle (\tilde{\hat{\phi}})^2 \rangle_\theta \Big). \tag{2.141}$$

We introduce the definition of ϕ_0 and we simplify some terms:

$$\frac{\partial \hat{\chi}_1}{\partial \hat{t}} = -\frac{\varepsilon_\delta B_0}{B}\frac{\partial \langle \hat{\phi} \rangle_\theta}{\partial \hat{\chi}_2} - \frac{\varepsilon_\delta^2 B_0^2}{B^2}\frac{\partial}{\partial \hat{\chi}_2}\Big(-\hat{\eta} \frac{1}{\hat{\rho}} \frac{\partial}{\partial \hat{\rho}} \langle (\tilde{\hat{\phi}})^2 \rangle_\theta \Big). \tag{2.142}$$

$$\frac{\partial \hat{\chi}_2}{\partial \hat{t}} = +\frac{\varepsilon_\delta B_0}{B}\frac{\partial \langle \hat{\phi} \rangle_\theta}{\partial \hat{\chi}_1} + \frac{\varepsilon_\delta^2 B_0^2}{B^2}\frac{\partial}{\partial \hat{\chi}_1}\Big(-\hat{\eta} \frac{1}{\hat{\rho}} \frac{\partial}{\partial \hat{\rho}} \langle (\tilde{\hat{\phi}})^2 \rangle_\theta \Big). \tag{2.143}$$

It is possible to introduce the definition of $\hat{B}$ and consequently of $\hat{A}$:

$$\dot{\hat{\chi}}_1 = -\hat{A}\frac{\partial \langle \hat{\phi} \rangle_\theta}{\partial \hat{\chi}_2} - \hat{A}^2 \frac{\partial}{\partial \hat{\chi}_2}\Big(-\hat{\eta} \frac{1}{\hat{\rho}} \frac{\partial}{\partial \hat{\rho}} \langle (\widetilde{\hat{\phi}})^2 \rangle_\theta \Big). \tag{2.144}$$

$$\dot{\hat{\chi}}_2 = +\hat{A}\frac{\partial \langle \hat{\phi} \rangle_\theta}{\partial \hat{\chi}_1} + \hat{A}^2 \frac{\partial}{\partial \hat{\chi}_1}\Big(-\hat{\eta} \frac{1}{\hat{\rho}} \frac{\partial}{\partial \hat{\rho}} \langle (\widetilde{\hat{\phi}})^2 \rangle_\theta \Big). \tag{2.145}$$

As we did in the dimensional equations it is possible to introduce $\hat{\Phi} = \hat{A}\hat{\phi}$:

$$\dot{\hat{\chi}}_1 = -\frac{\partial\langle\hat{\Phi}\rangle_\theta}{\partial\hat{\chi}_2} - \frac{\partial}{\partial\hat{\chi}_2}\Big(-\hat{\eta}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\langle(\tilde{\hat{\Phi}})^2\rangle_\theta\Big). \tag{2.146}$$

$$\dot{\hat{\chi}}_2 = +\frac{\partial\langle\hat{\Phi}\rangle_\theta}{\partial\hat{\chi}_1} + \frac{\partial}{\partial\hat{\chi}_1}\Big(-\hat{\eta}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\langle(\tilde{\hat{\Phi}})^2\rangle_\theta\Big). \tag{2.147}$$

Finally the definition of $\hat{\Psi}$, it is the same of Ψ (from Eq. [2.125]) but it includes $\hat{\Phi}$, after that it is necessary to introduce $\hat{\Psi}_{GC_1}$ and $\hat{\Psi}_{GC_2}$ from the non-dimensional version of Eq. [2.132] and Eq. [2.133]:

$$\dot{\hat{\chi}}_1 = -\frac{\partial\hat{\Psi}_{GC_1}}{\partial\hat{\chi}_2} - \frac{\partial\hat{\Psi}_{GC_2}}{\partial\hat{\chi}_2}. \tag{2.148}$$

$$\dot{\hat{\chi}}_2 = +\frac{\partial\hat{\Psi}_{GC_1}}{\partial\hat{\chi}_1} + \frac{\partial\hat{\Psi}_{GC_2}}{\partial\hat{\chi}_1}. \tag{2.149}$$

2.9 Full orbits governing equations

In full orbit mode, the code calculates the trajectories of the particles without any approximation coming from the gyro-kinetic approach to the problem. In fact, as we anticipated in the first section it is possible to integrate the equations of motion obtaining the solution in the 6 dimension phase space. The geometry simplification applies also in this case so we are interested in the solution in the 4 dimension phase space. The starting point system is the same as the complete model we introduced at the beginning:

$$H = \frac{1}{2}m\boldsymbol{v}^{\mathbf{2}} + q\phi. \tag{2.150}$$

$$\Big\{F,G\Big\}= \frac{1}{m}\Big(\frac{\partial F}{\partial\mathbf{x}}\frac{\partial G}{\partial\mathbf{v}} - \frac{\partial F}{\partial\mathbf{v}}\frac{\partial G}{\partial\mathbf{x}}\Big)+\frac{qB}{m^2}\Big(\frac{\partial F}{\partial v}\times\frac{\partial G}{\partial v}\Big). \tag{2.151}$$

Is possible to obtain the extended form of the Poisson Bracket of the system simply by doing both the dot and the cross product:

$$\Big\{F,G\Big\}= \frac{1}{m}\Big(\frac{\partial F}{\partial x}\frac{\partial G}{\partial v_x} - \frac{\partial F}{\partial v_x}\frac{\partial G}{\partial x}\Big)+\frac{1}{m}\Big(\frac{\partial F}{\partial y}\frac{\partial G}{\partial v_y} - \frac{\partial F}{\partial v_y}\frac{\partial G}{\partial y}\Big)+\frac{qB}{m^2}\Big(\frac{\partial F}{\partial v_x}\frac{\partial G}{\partial v_y} - \frac{\partial F}{\partial v_y}\frac{\partial G}{\partial v_x}\Big). \tag{2.152}$$

It is necessary to study the evolution in time of the observable x, y, v_x and v_y by taking advantage of $\dot{F} = \Big\{F,H\Big\}$. through some simple calculations it is possible to obtain the expression of the derivatives respect to the time of the observable of interest:

$$\begin{cases} \dot{x} = v_x, \\ \\ \dot{y} = v_y, \\ \\ \dot{v_x} = -\frac{q}{m}\frac{\partial\phi}{\partial x} + \frac{qB}{m}v_y, \\ \\ \dot{v_y} = -\frac{q}{m}\frac{\partial\phi}{\partial y} - \frac{qB}{m}v_x. \end{cases} \tag{2.153}$$

These are the so-called equations of motion that we would like to integrate into the code in order to obtain the desired result.

2.9.1 Non-dimensional governing equations for full orbits

To maintain the coherence to what has been done with the guiding centre it is necessary to proceed with the non-dimensionalization of the equations of the motion. It is necessary to define the non-dimensional quantities of interest:

$$\begin{gathered}\hat{x}=\frac{x}{L}2\pi \ \ ; \ \hat{y}=\frac{y}{L}2\pi \ \ ; \ \hat{t}=\frac{t}{T}2\pi \ \ ; \ \hat{v_x}=\frac{v_x}{L}T \ \ ; \\ \hat{v_y}=\frac{v_y}{L}T \ \ ; \ \hat{\eta}=\frac{\eta}{T}2\pi \ \ ; \ \hat{\phi}=\frac{\phi}{\phi_0} \ \ ; \ \hat{B}=\frac{B}{B_0}.\end{gathered} \tag{2.154}$$

It is of great importance the definition of ϕ_0 we do it using the following definition:

$$\phi_0=\frac{L^2B_0}{2\pi T}.$$

In the equations, we have made the same assumptions on the potential so that we will have $\phi=\varepsilon_\delta\phi$. It is now possible to start to develop the equation, the procedure will be done for the first equation but it is the same for the second one:

$$\frac{\partial\hat{v_x}}{\partial\hat{t}}\frac{L}{T^2}2\pi=-\frac{q}{m}\phi_0\frac{2\pi}{L}\frac{\partial(\varepsilon_\delta\hat{\phi})}{\partial\hat{x}}+\frac{qB}{m}\frac{L}{T}\hat{v_y}. \tag{2.155}$$

In the definition of ϕ_0 it is possible to insert the definition of B_0:

$$\text{with } B_0=\frac{B}{\hat{B}}\Rightarrow\phi_0=\frac{L^2B}{2\pi T\hat{B}}. \tag{2.156}$$

We substitute the definition of ϕ_0 in the equation of interest:

$$\frac{\partial\hat{v_x}}{\partial\hat{t}}\frac{L}{T^2}2\pi=-\frac{q}{m}\frac{L^2B}{2\pi T\hat{B}}\frac{2\pi}{L}\frac{\partial(\varepsilon_\delta\hat{\phi})}{\partial\hat{x}}+\frac{qB}{m}\frac{L}{T}\hat{v_y}. \tag{2.157}$$

It is possible to simplify the equation and bring $\hat{B}$ inside the partial derivative with respect to $\hat{x}$:

$$\frac{\partial\hat{v_x}}{\partial\hat{t}}\frac{2\pi}{T}=-\frac{qB}{m}\frac{\partial(\frac{\varepsilon_\delta\hat{\phi}}{\hat{B}})}{\partial\hat{x}}+\frac{qB}{m}\hat{v_y}. \tag{2.158}$$

Knowing that $\frac{qB}{m}=\frac{1}{2\eta}$ we introduce η in the equation, and we use it to introduce $\hat{\eta}$:

$$\frac{\partial\hat{v_x}}{\partial\hat{t}}\frac{2\pi}{T}=-\frac{1}{2\eta}\frac{\partial(\frac{\varepsilon_\delta\hat{\phi}}{\hat{B}})}{\partial\hat{x}}+\frac{1}{2\eta}\hat{v_y}=-\frac{1}{2\hat{\eta}}\frac{2\pi}{T}\frac{\partial(\frac{\varepsilon_\delta\hat{\phi}}{\hat{B}})}{\partial\hat{x}}+\frac{1}{2\hat{\eta}}\frac{2\pi}{T}\hat{v_y}. \tag{2.159}$$

The final simplified equations are the following:

$$\frac{\partial\hat{v_x}}{\partial\hat{t}}=-\frac{1}{2\hat{\eta}}\frac{\partial(\frac{\varepsilon_\delta\hat{\phi}}{\hat{B}})}{\partial\hat{x}}+\frac{1}{2\hat{\eta}}\hat{v_y}. \tag{2.160}$$

$$\frac{\partial \hat{v_y}}{\partial \hat{t}} = -\frac{1}{2\hat{\eta}}\frac{\partial(\frac{\varepsilon_\delta \hat{\phi}}{\hat{B}})}{\partial \hat{y}} - \frac{1}{2\hat{\eta}}\hat{v_x}. \tag{2.161}$$

It is possible to perform a re-scaling on a non-dimensional parameter in order to take advantage of the definition of a certain parameter. Starting from the definition of the Larmor radius we perform the non-dimensionalization:

$$\rho = \frac{m}{|q|B}v_\perp = 2|\eta|v_\perp. \tag{2.162}$$

With the definition of:

$$\hat{\rho} = \frac{\rho}{L} \ \ ; \ \hat{v}_\perp = v_\perp\frac{T}{L}. \tag{2.163}$$

It is easy to obtain:

$$\hat{\rho} = 2|\hat{\eta}|v_\perp. \tag{2.164}$$

I divide both the equations for $\hat{v_\perp}$ and I take advantage of the last equation:

$$\frac{\partial \hat{v}_x}{\partial \hat{t}}\frac{1}{\hat{v}_\perp} = -\frac{1}{2|\hat{\eta}|}\frac{1}{\hat{v}_\perp}\frac{\partial(\frac{\varepsilon_\delta \hat{\phi}}{\hat{B}})}{\partial \hat{x}} + \frac{1}{2|\hat{\eta}|}\frac{1}{\hat{v}_\perp}\hat{v}_y. \tag{2.165}$$

$$\frac{\partial \hat{v}_y}{\partial \hat{t}}\frac{1}{\hat{v}_\perp} = -\frac{1}{2|\hat{\eta}|}\frac{1}{\hat{v}_\perp}\frac{\partial(\frac{\varepsilon_\delta \hat{\phi}}{\hat{B}})}{\partial \hat{y}} - \frac{1}{2|\hat{\eta}|}\frac{1}{\hat{v}_\perp}\hat{v}_x. \tag{2.166}$$

In addition to the definition of the non-dimensional Larmor radius, I get an advantage by exploiting two more definitions:

$$\hat{V}_x = \frac{\hat{v_x}}{\hat{v_\perp}} \ \ ; \ \hat{V}_y = \frac{\hat{v_y}}{\hat{v_\perp}}. \tag{2.167}$$

Those two last parameters are non-dimensional but they are also re-scaled on $\hat{v_\perp}$. This fact must be taken into account later. To implement easily $|\hat{\eta}|$ in the code it is possible to simply use: $|\hat{\eta}| = \hat{\eta} sign(\hat{\eta})$. It is also possible to use the definition of the parameter $\hat{A} = \frac{\varepsilon_\delta}{\hat{B}}$ to obtain the final expressions of the equation:

$$\frac{\partial \hat{V}_x}{\partial \hat{t}} = -sign(\hat{\eta})\frac{\hat{A}}{\hat{\rho}}\frac{\partial \hat{\phi}}{\partial \hat{x}} + \frac{1}{2|\hat{\eta}|}\hat{V}_y. \tag{2.168}$$

$$\frac{\partial \hat{V}_y}{\partial \hat{t}} = -sign(\hat{\eta})\frac{\hat{A}}{\hat{\rho}}\frac{\partial \hat{\phi}}{\partial \hat{y}} - \frac{1}{2|\hat{\eta}|}\hat{V}_x. \tag{2.169}$$

As we did for Guiding centres it is possible to introduce $\hat{\Phi} = \hat{A}\hat{\phi}$ obtaining the version of the equations that will be implemented in the code::

$$\dot{\hat{V}}_x = -sign(\hat{\eta})\frac{1}{\hat{\rho}}\frac{\partial \hat{\Phi}}{\partial \hat{x}} + \frac{1}{2|\hat{\eta}|}\hat{V}_y. \tag{2.170}$$

$$\dot{\hat{V}}_y = -sign(\hat{\eta})\frac{1}{\hat{\rho}}\frac{\partial \hat{\Phi}}{\partial \hat{y}} - \frac{1}{2|\hat{\eta}|}\hat{V}_x. \tag{2.171}$$

These are the equations that will be implemented together with the ones for x and y:

$$\dot{\hat{x}} = \frac{\hat{\rho}}{2|\hat{\eta}|}\hat{V}_x \tag{2.172}$$

$$\dot{\hat{y}} = \frac{\hat{\rho}}{2|\hat{\eta}|}\hat{V}_y \tag{2.173}$$

In our case we would like to choose the Larmor radius as a thermal Larmor radius, this means that the $v_\perp$ is defined as the thermal velocity:

$$v_\perp = \sqrt{\frac{K_B T}{m}} \quad \Rightarrow \quad \rho = \frac{m}{|q|B}v_\perp = \frac{\sqrt{mK_BT}}{|q|B} = \rho_{th}. \tag{2.174}$$

2.9.2 Re-scaling factor

In addition to that, from the equation of velocity, it is possible to establish the re-scaling factor for which we should account when we want to confront the velocity of the Guiding Center and the velocity of the corresponding particle. As we have seen in fact, we are not calculating $\hat{v}_x$ but $\hat{V}_x$ in the differential equations. Given that the equations have been divided for $v_\perp$ this is the factor in which we are interested. The relation between $\hat{v}_x$ but $\hat{V}_x$ is described by the following equation:

$$\hat{v}_x = \dot{\hat{x}} = \hat{V}_x \hat{v}_\perp. \tag{2.175}$$

The re-scaling parameter is fixed, it can't be different for each particle. Given the validity of Eq. [2.174] it is straightforward:

$$\hat{v}_\perp = \frac{\hat{\rho}}{2|\hat{\eta}|}. \tag{2.176}$$

In order to fix it we can easily choose it following the next definition:

$$\hat{v}_\perp = \frac{\hat{\rho}_{dict}}{2|\hat{\eta}_{dict}|}. \tag{2.177}$$

After having computed full orbits of particles, we would like to obtain the trajectories of the corresponding guiding centres, once having done that, the comparison between the results from the two different approaches can be better performed. In order to compare the results, we have to account for the fact we are not calculating strictly $\hat{v}_x$ and $\hat{v}_y$:

$$\hat{\rho} = 2|\hat{\eta}|\sqrt{\hat{v}_x^2 + \hat{v}_y^2} = 2|\hat{\eta}|\hat{v}_\perp\sqrt{\hat{V}_x^2 + \hat{V}_y^2} = \hat{\rho}_{dict}\sqrt{\hat{V}_x^2 + \hat{V}_y^2}. \tag{2.178}$$

Another point to discuss is the initialization of $\hat{v}_\perp$ in the code. Given the definition of the dimensional Larmor radius and consequently, of the non-dimensional Larmor radius, it is possible to calculate what should be the value of $\hat{v}_\perp$ given a couple of $\hat{\rho}$ and $\hat{\eta}$ defined in the code:

$$\hat{v}_\perp = \frac{\hat{\rho}}{2|\hat{\eta}|} \Rightarrow \hat{v}_{\perp 0} = \frac{\hat{\rho}_{dict}}{2|\hat{\eta}_{dict}|}. \tag{2.179}$$

This last equation has to be valid at the first instant of time otherwise the initial conditions are wrong. It has to be reminded of the validity of Eq. [2.175] for $\hat{v}_\perp$. Let's define some additional parameters to be more clear in the next passages:

$\hat{v}_{\perp\ real}$: It is the real transverse velocity that has to satisfy the condition introduced in Eq. [2.179].

$\hat{V}_{\perp\ calc}$: It is the transverse velocity calculated in the code through $\hat{V}_x$ and $\hat{V}_y$, we need to remember that those are re-scaled on the chosen $\hat{v}_\perp$.

$\hat{v}_\perp$: It is the scaling factor of the equations, we remember it is defined as in Eq. [2.177] through the use of $\hat{\rho}_{dict}$ and $\hat{\eta}_{dict}$.

Taking as a reference Eq. [2.175] it is possible to write:

$$\hat{v}_{\perp\ real} = \hat{V}_{\perp\ calc}\hat{v}_\perp. \tag{2.180}$$

Then the validity of the equation on initial conditions forces the value of $\hat{V}_{\perp\ calc}$ to 1 as done in the code.

2.10 Non-dimensional potential model

Like anticipated the turbulent potential is modelled so that it is equipped with the main characteristics of a real potential within a tokamak. It is not a real potential map given that the geometry considered is idealised with respect to a possible real one. The equation for the non-dimensional potential is the following:

$$\hat{\phi} = \sum_{\substack{n,m=1 \\ n^2+m^2 \leq M^2}}^{M} \frac{1}{(n^2+m^2)^{3/2}} \sin\left(n\hat{x} + m\hat{y} - \varphi_{nm} - \hat{t}\right). \tag{2.181}$$

From the previous sections, we have then the validity of:

$$\hat{\Phi} = \frac{\varepsilon_\delta}{\hat{B}}\hat{\phi}. \tag{2.182}$$

The latter will be the potential inserted in the equations, the one containing also the $\hat{A}$:

$$\hat{\Phi} = \sum_{\substack{n,m=1 \\ n^2+m^2 \leq M^2}}^{M} \frac{\hat{A}}{(n^2+m^2)^{3/2}} \sin\left(n\hat{x} + m\hat{y} - \varphi_{nm} - \hat{t}\right). \tag{2.183}$$

Given that the potential is non-dimensional and that it is periodic, the definitions of the non-dimensional parameters introduced previously grant a periodicity in space and time with a period of 2π. The so-defined potential is represented in a square of side 2π in the transverse plane (x, y). The above-mentioned square will be called *fundamental square* from now on. The periodicity assures that it is possible to calculate the potential in the square to calculate the trajectories of particles also outside of it, this aspect will be deepened later but is very important for the implementation in the code. The expression of the potential in Eq. [2.183] is not convenient and it is quite long, it can't be implemented efficiently inside the code. The idea is to write the potential in terms of a complex potential $\hat{\Phi}_c$ in order to take advantage of the fast implementation and of its easy writing. From the complex analysis it is well known the following formula:

$$\Phi = \mathfrak{Im}\{\Phi_c(x,y)e^{-jt}\} = \mathfrak{Im}\{\Phi_c(x,y)\}\cos(t) - \mathfrak{Re}\{\Phi_c(x,y)\}\sin(t). \tag{2.184}$$

In the latter equation, real and imaginary parts are indicated by $\mathfrak{Re}\{\}$ and $\mathfrak{Im}\{\}$ respectively. Together with this it is also possible to write the complex potential Φ_c in the following way:

$$\Phi_c = \sum_{\substack{n,m=1 \\ n^2+m^2\leq M^2}}^{M} K_{nm}e^{j(nx+my)}. \tag{2.185}$$

through the obvious definition of K_{nm}:

$$K_{nm} = \frac{Ae^{\varphi_{nm}}}{(n^2+m^2)^{3/2}}. \tag{2.186}$$

The potential with this shape can be directly inserted inside the equations of full orbit mode, we already have everything we need in order to run the code and calculate the trajectories of particles in the designated potential. On the other hand, for Guiding Center Mode we need further manipulations to the potential in order to obtain the ready-to-implement expressions, it is necessary to proceed with the change of coordinates from Eq. [2.14] and Eq. [2.15] in order to rewrite Eq. [2.184], Eq. [2.185] and Eq. [2.186]:

$$\begin{gathered}\Phi(x,y,t) = \Phi(\chi_1+\rho\cos(\theta), \chi_2-\rho\sin(\theta), t) = \\ = \sum_{\substack{n,m=1 \\ n^2+m^2\leq M^2}}^{M} \frac{A}{(n^2+m^2)^{3/2}}\sin\left(n(\chi_1+\rho\cos(\theta)) + m(\chi_2-\rho\sin(\theta)) - \varphi_{nm} - t\right).\end{gathered} \tag{2.187}$$

$$\begin{aligned}&\Phi(x,y,t) = \Phi(\chi_1+\rho\cos(\theta), \chi_2-\rho\sin(\theta), t) = \\ &= \mathfrak{Im}\{\Phi_c(\chi_1+\rho\cos(\theta), \chi_2-\rho\sin(\theta))\}\cos(t) + \\ &- \mathfrak{Re}\{\Phi_c(\chi_1+\rho\cos(\theta), \chi_2-\rho\sin(\theta))\}\sin(t).\end{aligned} \tag{2.188}$$

The expression of the complex potential is:

$$\begin{aligned}&\Phi_c(x,y) = \Phi(\chi_1+\rho\cos(\theta), \chi_2-\rho\sin(\theta)) = \\ &= \sum_{\substack{n,m=1 \\ n^2+m^2\leq M^2}}^{M} K_{nm}e^{j(nx+my)}e^{j(n\rho cos(\theta)-m\rho\sin(\theta))}.\end{aligned} \tag{2.189}$$

Now that the complex potential is available it is possible to find an alternative expression of the guiding centre potential, using the last three equations it is possible to simplify the expression of Ψ_{GC_1} and of Ψ_{GC_2}. The use of Φ_c is very useful given that we have to deal with a gyro-average applied to the potential itself. The procedure foresees the simplification of the two orders of the guiding centre potential separately so that the passages are more clear.

2.11 First-order guiding centre potential

Like what has been done with Φ, Ψ_{GC_1} can be expressed in terms of a complex guiding centre potential $\Psi_{GC_{1,c}}$:

$$\Psi_{GC_1} = \Psi_{GC_{1,1}} = \Im\mathfrak{m}\{\Psi_{GC_{1,c}} e^{-jt}\}. \tag{2.190}$$

Given the validity of Eq. [2.132] it is possible to write:

$$\Psi_{GC_{1,c}} = \langle \Phi_c(\chi_1 + \rho\cos(\theta), \chi_2 - \rho\sin(\theta)) \rangle. \tag{2.191}$$

Given the validity of Eq. [2.189] and that the average is made on θ it is possible to obtain a simpler expression:

$$\Psi_{GC_{1,c}} = \sum_{\substack{n,m=1 \\ n^2+m^2 \leq M^2}}^{M} K_{nm} e^{j(nx+my)} \left\langle e^{j(n\rho cos(\theta) - m\rho\sin(\theta))} \right\rangle. \tag{2.192}$$

The average acts only on the exponential containing the dependency of θ, this will simplify further calculations. It is now time to actively perform the gyro-average and look for the final expression of the first-order guiding centre potential. The desired result should be a sort of operator, With the application of the latter to the complex potential we should be able to account for the effect of the average made on θ. Knowing the effect of the *FLR* effects we already know what to expect from the operator, like we anticipated the final results will be a smoother potential which means that all the peaks will be reduced, the amplitude of each mode will be lowered, the operator will result in a coefficient lower than 1. In order to perform the average it is necessary to introduce the Bessel function for the first kind $\mathcal{J}_\nu$:

$$\mathcal{J}_\nu(s) = \int_{-\pi}^{+\pi} e^{j(s\ sin\beta - \nu\beta)} d\beta. \tag{2.193}$$

$$\mathcal{J}_\nu(s) = \sum_{k=0}^{\infty} (-1)^k \frac{(s/2)^{2k+\nu}}{k!\ \Gamma(k+\nu+1)}. \tag{2.194}$$

In these equations the following definitions are valid:

- s is a generic variable.
- ν is an integer number constituting the order of the Bessel function.
- Γ is the so-called Gamma function.

While considering the Bessel function for the first kind and of order 0 ($\nu = 0$) the two previous equations become:

$$\mathcal{J}_0(s) = \int_{0}^{2\pi} e^{j(s\ sin\beta)} d\beta. \tag{2.195}$$

$$\mathcal{J}_0(s) = \sum_{k=0}^{\infty} (-1)^k \frac{(s/2)^{2k+\nu}}{(k!)^2}. \tag{2.196}$$

In order to obtain Eq. [2.196] one has to take advantage of the following useful relation:

$$\Gamma(a+1) = a!. \tag{2.197}$$

At this point it is possible to develop the gyro-average acting on Eq. [2.192]:

$$\begin{aligned} &\Big\langle e^{j(n\rho cos(\theta)-m\rho\sin{(\theta)})}\Big\rangle = \Big\langle e^{j\frac{\sqrt{n^2+m^2}}{\sqrt{n^2+m^2}}(n\rho cos(\theta)-m\rho\sin{(\theta)})}\Big\rangle = \\ &=\Big\langle \exp\Big[j\rho\sqrt{n^2+m^2}\Big(\frac{n}{\sqrt{n^2+m^2}}\cos{(\theta)}-\frac{m}{\sqrt{n^2+m^2}}\sin{(\theta)}\Big)\Big]\Big\rangle. \end{aligned} \tag{2.198}$$

It is possible to express the coefficients multiplying $\cos{(\theta)}$ and $\sin{(\theta)}$ in terms of trigonometric functions as well:

$$\cos{(\theta_{nm})} = \frac{n}{\sqrt{n^2+m^2}}. \tag{2.199}$$

$$\sin{(\theta_{nm})} = \frac{m}{\sqrt{n^2+m^2}}. \tag{2.200}$$

At this point it is possible to apply a useful relation for the cosine of a sum:

$$\cos{(\theta+\theta_{nm})} = \cos\theta\cos\theta_{nm} - \sin\theta\sin\theta_{nm}. \tag{2.201}$$

With the application of these last three relations Eq. [2.198] becomes:

$$\Big\langle e^{j(n\rho cos(\theta)-m\rho\sin{(\theta)})}\Big\rangle = \Big\langle \exp\Big[j\rho\sqrt{n^2+m^2}\Big(\cos{(\theta+\theta_{nm})}\Big)\Big]\Big\rangle. \tag{2.202}$$

It is known then that:

$$\sin(\beta) = \cos{(\beta+\pi/2)}. \tag{2.203}$$

A specific value of β has to be chosen so you can take advantage of the definition of the Bessel function of order zero, looking at Eq.2.195 it is clear that the $\frac{\pi}{2}$ has to be cancelled, in order to do that the β needs to satisfy some criteria:

$$\beta = \theta + \theta_{nm} - \pi/2. \tag{2.204}$$

$$d\beta = d\theta. \tag{2.205}$$

So that in the Eq. [2.195] we have:

$$\sin\beta = \cos{(\beta+\pi/2)} = \cos{(\theta+\theta_{nm})}. \tag{2.206}$$

By applying Eq. [2.195] and by performing these last passages Eq. [2.202] can be developed:

$$\Big\langle e^{j(n\rho cos(\theta)-m\rho\sin{(\theta)})}\Big\rangle = \int_0^{2\pi} e^{j\rho\sqrt{n^2+m^2}(\cos\theta+\theta_{nm})}d\theta = \mathcal{J}_0(\rho\sqrt{n^2+m^2}). \tag{2.207}$$

In order to lighten the notation, it can be introduced an additional definition:

$$s_{nm} = \sqrt{n^2+m^2}. \tag{2.208}$$

It is now possible to introduce the expression of the average inside the starting equation, with the substitution inside Eq. [2.192] we obtain:

$$\Psi_{GC_{1,c}} = \sum_{\substack{n,m=1 \\ n^2+m^2 \leq M^2}}^{M} K_{nm} \mathcal{J}_0(\rho s_{nm}) e^{j(n\chi_1 + m\chi_2)}. \tag{2.209}$$

It is important to mention that, since the gyro-average has been performed and θ has been averaged out, then ρ has become a parameter, it is no more variable. We remember that from Eq. [2.65], once the Lie transform has been applied as well as the order reduction we checked that the *magnetic moment* or the *Larmor radius* would have become a parameter, ρ is no more considered as a variable (like χ_1 and χ_2 for example). In addition to that, we can now evaluate the global effect of the *FLR* effects for the first order, it is clear that we are applying a coefficient to each mode of the potential, it is an additional coefficient with respect to the already present K_{nm}. Given the trend of the Bessel function of the first kind and order zero Fig. [2.1] it is clear that whether is the value of ρ parameter and even for modes with low s_{nm} we will have a decrease in the amplitude of the guiding potential for every mode.

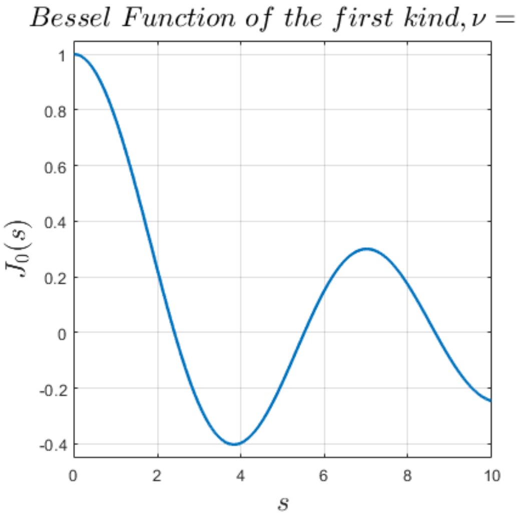


Figure 2.1: Bessel function of the first kind of order zero.

2.12 Second-order guiding center potential

It is now time to repeat the procedure for the second order term, in this case, it is more difficult given that we have to deal with the fluctuating part of the potential. The starting point is always the same relation, writing the potential in terms of a complex one:

$$\Psi_{GC_2} = \Im\mathfrak{m}\{\Psi_{GC_{2,c}} e^{-jt}\}. \tag{2.210}$$

In this case, however, we can not proceed directly, as anticipated we have to deal with the fluctuating part of the potential. We recall Eq. [2.133]:

$$\Psi_{GC_2} = -\eta \frac{1}{\rho}\frac{\partial}{\partial \rho}\langle (\tilde{\Phi})^2\rangle_\theta. \tag{2.211}$$

To simplify the implementation it is possible to write $\tilde{\Phi}$ in the following way:

$$\langle \tilde{\Phi}^2\rangle = \langle \Phi^2\rangle - \langle \Phi\rangle^2. \tag{2.212}$$

This causes us to work with two terms instead of one but it will be easier to deal with two non-fluctuating terms instead of a single fluctuating one, the expression of the second order guiding-center potential changes, Eq. [2.211] becomes:

$$\Psi_{GC_2} = -\eta \frac{1}{\rho}\frac{\partial}{\partial \rho}(\langle \Phi^2\rangle - \langle \Phi\rangle^2) = -\eta \frac{1}{\rho}\frac{\partial}{\partial \rho}\langle \Phi^2\rangle + \eta \frac{1}{\rho}\frac{\partial}{\partial \rho}\langle \Phi\rangle^2. \tag{2.213}$$

we can now introduce a new symbology, accounting for the two terms on the right-hand side:

$$\Psi^{(\mathrm{I})}_{GC_2} = -\eta \frac{1}{\rho}\frac{\partial}{\partial \rho}\langle \Phi^2\rangle. \tag{2.214}$$

$$\Psi^{(\mathrm{II})}_{GC_2} = +\eta \frac{1}{\rho}\frac{\partial}{\partial \rho}\langle \Phi\rangle^2. \tag{2.215}$$

So that Eq. [2.213] becomes now:

$$\Psi_{GC_2} = \Psi^{(\mathrm{I})}_{GC_2} + \Psi^{(\mathrm{II})}_{GC_2}. \tag{2.216}$$

In order to better understand the following passages it is also convenient to introduce a new definition for the product of the complex potential Φ_c with the exponential accounting for the time dependence e^{-jt}:

$$\Psi_{c,\tau}(\chi_1, \chi_2, \rho, \theta, t) = \Phi_{c,}(\chi_1, \chi_2, \rho, \theta)e^{-jt}. \tag{2.217}$$

As for the Poisson Brackets, τ accounts for the time dependence. At this point, it is possible to take advantage of useful relations coming from complex analysis, from Eq. [2.184] we have:

$$\Phi = \Im\mathfrak{m}\{\Psi_{c,\tau}\}. \tag{2.218}$$

It is easy to write:

$$\Phi = \frac{\Psi_{c,\tau} - \Psi^*_{c,\tau}}{2j}. \tag{2.219}$$

And it is possible to calculate Φ^2:

$$\Phi^2 = \frac{\Psi^2_{c,\tau} + \Psi^{*2}_{c,\tau} - 2\Psi_{c,\tau}\Psi^*_{c,\tau}}{-4}. \tag{2.220}$$

It is possible to develop the individual terms:

$$\Psi^2_{c,\tau} = \Re\mathfrak{e}\{\Psi_{c,\tau}\}^2 - \Im\mathfrak{m}\{\Psi_{c,\tau}\}^2 + 2j\Re\mathfrak{e}\{\Psi_{c,\tau}\}\Im\mathfrak{m}\{\Psi_{c,\tau}\}. \tag{2.221}$$

$$\Psi^{*2}_{c,\tau} = \mathfrak{Re}\{\Psi_{c,\tau}\}^2 + \mathfrak{Im}\{\Psi_{c,\tau}\}^2 - 2j\mathfrak{Re}\{\Psi_{c,\tau}\}\mathfrak{Im}\{\Psi_{c,\tau}\}. \tag{2.222}$$

$$\begin{aligned}\Psi_{c,\tau}\Psi^{*}_{c,\tau} &= \mathfrak{Re}\{\Psi_{c,\tau}\}^2 + \mathfrak{Im}\{\Psi_{c,\tau}\}^2 + j\left[\mathfrak{Re}\{\Psi_{c,\tau}\}\mathfrak{Im}\{\Psi_{c,\tau}\}+\right.\\ &\left.- \mathfrak{Im}\{\Psi_{c,\tau}\}Re\{\Psi_{c,\tau}\}\right]= \mathfrak{Re}\{\Psi_{c,\tau}\}^2 + \mathfrak{Im}\{\Psi_{c,\tau}\}^2.\end{aligned} \tag{2.223}$$

We put everything together so that Eq. [2.220] becomes:

$$\begin{aligned}\Phi^2 = \frac{1}{2}\mathfrak{Im}\{\Psi_{c,\tau}\}^2 &= \frac{1}{2}\mathfrak{Re}\{\Psi_{c,\tau}\}^2 + \frac{1}{2}\mathfrak{Im}\{\Psi_{c,\tau}\}^2 - \frac{1}{2}\mathfrak{Re}\{\Psi_{c,\tau}\}^2 =\\ &= \frac{1}{2}|\Psi_{c,\tau}|^2 - \frac{1}{2}\mathfrak{Re}\{\Psi^2_{c,\tau}\}.\end{aligned} \tag{2.224}$$

In the case of Eq. [2.214], to obtain the expression of the desired term it is necessary to simply apply the gyro-average to Eq. [2.224]:

$$\langle\Phi^2\rangle = \frac{1}{2}\langle|\Psi_{c,\tau}|^2\rangle - \frac{1}{2}\mathfrak{Re}\{\langle\Psi^2_{c,\tau}\rangle\}. \tag{2.225}$$

On the other hand in the case of the term inside Eq. [2.215] it is necessary to apply the operator directly to $\Psi_{c,\tau}$. Can be easily proven that if we repeat the calculations required to obtain Eq. [2.224] after having applied the operator we would obtain the exact same result but with the $\langle\ \rangle$ operator applied directly to $\Psi_{c,\tau}$ in the equation:

$$\langle\Phi\rangle^2 = \frac{1}{2}|\langle\Psi_{c,\tau}\rangle|^2 - \frac{1}{2}\mathfrak{Re}\{\langle\Psi_{c,\tau}\rangle^2\}. \tag{2.226}$$

it is now possible to write the new expression of Eq. [2.214] and Eq. [2.215] through Eq. [2.225] and Eq. [2.226]:

$$\Psi^{(\mathrm{I})}_{GC_2} = -\frac{\eta}{2\rho}\frac{\partial}{\partial\rho}\Big(\langle|\Psi_{c,\tau}|^2\rangle - \mathfrak{Re}\{\langle\Psi^2_{c,\tau}\rangle\}\Big). \tag{2.227}$$

$$\Psi^{(\mathrm{II})}_{GC_2} = +\frac{\eta}{2\rho}\frac{\partial}{\partial\rho}\Big(|\langle\Psi_{c,\tau}\rangle|^2 - \mathfrak{Re}\{\langle\Psi_{c,\tau}\rangle^2\}\Big). \tag{2.228}$$

Eq. [2.228] can be further developed, when one has to proceed with the derivation he has to account that he is deriving the square of a function and that $|\langle\Psi_{c,\tau}\rangle|^2$ can be further developed as well:

$$|\Psi_{c,\tau}|^2 = \Psi_{c,\tau}\Psi^*_{c,\tau} \quad\Rightarrow\quad |\langle\Psi_{c,\tau}\rangle|^2 = \langle\Psi_{c,\tau}\rangle\langle\Psi^*_{c,\tau}\rangle. \tag{2.229}$$

For this term, the rule of the derivation of a product will be followed. Before completing any other steps it is convenient to introduce a further notation, we consider a global operator, which includes the partial derivation with respect to ρ, directly acting on the various potentials:

$$\mathbb{J}(\square) = \frac{1}{\rho}\frac{\partial}{\partial\rho}\langle\square\rangle. \tag{2.230}$$

With the introduction of this notation inside Eq. [2.227] and Eq. [2.228] while considering also the new expression of $|\langle\Psi_{c,\tau}\rangle|^2$ and the derivation rules we obtain:

$$\Psi^{(\mathrm{I})}_{GC_2} = -\frac{\eta}{2}\Big(\mathbb{J}(|\Psi_{c,\tau}|^2) - \mathfrak{Re}\{\mathbb{J}(\Psi^2_{c,\tau})\}\Big). \tag{2.231}$$

$$\Psi^{(\mathrm{II})}_{GC_2} = +\frac{\eta}{2}\Big(\langle\Psi_{c,\tau}\rangle\mathbb{J}(\Psi^*_{c,\tau})+\langle\Psi^*_{c,\tau}\rangle\mathbb{J}(\Psi_{c,\tau})-\Re\mathfrak{e}\{2\langle\Psi_{c,\tau}\rangle\mathbb{J}(\Psi_{c,\tau})\}\Big). \tag{2.232}$$

At this point we can put everything together re-writing Eq. [2.216]:

$$\begin{split}\Psi_{GC_2} = \Psi^{(\mathrm{I})}_{GC_2} + \Psi^{(\mathrm{II})}_{GC_2} = +\frac{\eta}{2}\Big(-\mathbb{J}(|\Psi_{c,\tau}|^2)+\langle\Psi_{c,\tau}\rangle\mathbb{J}(\Psi^*_{c,\tau})+\langle\Psi^*_{c,\tau}\rangle\mathbb{J}(\Psi_{c,\tau})+\\ +\Re\mathfrak{e}\{\mathbb{J}(\Psi^2_{c,\tau})-2\langle\Psi_{c,\tau}\rangle\mathbb{J}(\Psi_{c,\tau})\}\Big).\end{split} \tag{2.233}$$

Before the implementation, it can be useful to further re-label the terms of the first and second order, to make immediate the association to the respective temporal dependence. As we know, the following relations will be implemented in the code:

$$\Phi = \Im\mathfrak{m}\{\Phi_c(x,y)e^{-jt}\}. \tag{2.234}$$

$$\Psi_{GC_1} = \Psi_{GC_{1,1}} = \Im\mathfrak{m}\{\Psi_{GC_{1,c}}e^{-jt}\}. \tag{2.235}$$

$$\Phi = \Im\mathfrak{m}\{\Psi_{c,\tau}\}. \tag{2.236}$$

This last Eq. will be inserted in every term of Ψ_{GC_2}. It can be easily proven that some of those terms won't be time-dependent because of the multiplication for their conjugate. It is necessary to recall Euler's formula:

$$e^{jt} = \cos(t) + j\sin(t). \tag{2.237}$$

Consequently, we have:

$$e^{-jt} = \cos(-t) + j\sin(-t) = \cos(t) - j\sin(t). \tag{2.238}$$

With the use of simple trigonometric formulas. through the definition of $\Psi_{c,\tau}$ from Eq. [2.217] we can write easily that:

$$\Psi_{c,\tau} = \Phi_c\cos(t) - j\Phi_c\sin(t). \tag{2.239}$$

In case we try to calculate its conjugate we have:

$$\Psi^*_{c,\tau} = \Phi_c\cos(t) + j\Phi_c\sin(t) = \Phi_c e^{jt}. \tag{2.240}$$

This means that in every case we have a product between $\Psi_{c,\tau}$ and its conjugate the time dependence is cancelled. Following this simple logic, it is possible to write the global guiding centre potential as:

$$\Psi = \Psi_{GC_{1,1}} + \Psi_{GC_{2,0}} + \Psi_{GC_{2,2}}. \tag{2.241}$$

The first number in the subscript indicates the order of the individual terms while the second one indicates the factor multiplying time in the exponential, we can resume the expressions of the three terms:

$$\Psi_{GC_{1,1}} = \Im\mathfrak{m}\{\Psi_{GC_{1,c}}e^{-1jt}\}. \tag{2.242}$$

$$\Psi_{GC_{2,0}} = +\frac{\eta}{2}\Big(-\mathbb{J}(|\Phi_c|^2)+\langle\Phi_c\rangle\mathbb{J}(\Phi^*_c)+\langle\Phi^*_c\rangle\mathbb{J}(\Phi_c)\Big)e^{-0jt}. \tag{2.243}$$

$$\Psi_{GC_{2,2}} = +\frac{\eta}{2}\Re\mathfrak{e}\Big\{\big[\mathbb{J}(\Phi_c^2) - 2\langle\Phi_c\rangle\mathbb{J}(\Phi_c)\big]e^{-2jt}\Big\}. \tag{2.244}$$

The last thing to do is obtain the expression of the operator $\mathbb{J}(\Box)$, knowing from Eq. [2.209] the expression of $\langle\ \rangle$:

$$\langle\Box\rangle = \sum_{\substack{n,m=1\\ n^2+m^2\leq M^2}}^{M} \Box_{nm}\mathcal{J}_0(\rho s_{nm})e^{j(n\chi_1+m\chi_2)}. \tag{2.245}$$

And the expression of the operator $\mathbb{J}(\Box)$ from Eq. [2.230]:

$$\mathbb{J}(\Box) = \frac{1}{\rho}\frac{\partial}{\partial\rho}\langle\Box\rangle. \tag{2.246}$$

Thanks to the properties of the Bessel functions it is possible to obtain the desired expression simply deriving with respect to ρ, it is possible to use the following useful relation:

$$\frac{\partial\mathcal{J}_0(s)}{\partial s} = -\mathcal{J}_1(s). \tag{2.247}$$

By applying Eq. [2.247] the expression of $\mathbb{J}(\Box)$ can be easily obtained:

$$\mathbb{J}(\Box) = -\sum_{\substack{n,m=1\\ n^2+m^2\leq M^2}}^{M} \Box_{nm}\frac{s_{nm}}{\rho}\mathcal{J}_1(\rho s_{nm})e^{j(n\chi_1+m\chi_2)}. \tag{2.248}$$

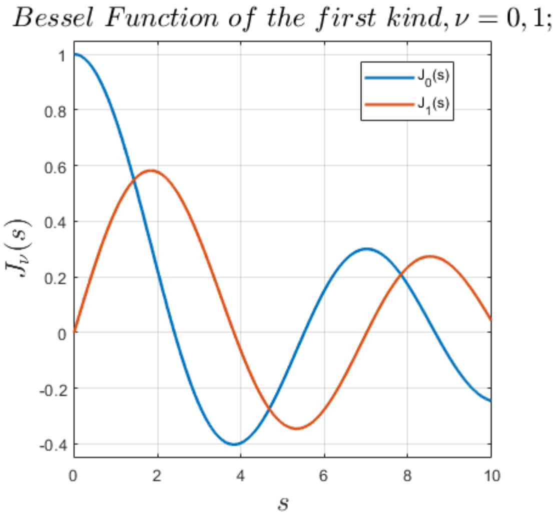


Figure 2.2: Bessel function of the first kind of order zero and one.

While considering $\Box$ a generic quantity (e.g. Φ_c) while $\Box_{nm}$ its related coefficient (e.g. K_{nm}). Naturally, it is necessary a link between the two quantities, this can be done through the *Fast Fourier Transform* algorithm in two dimensions and its inverse, this algorithm is very powerful and efficient with respect to a standard Fourier transform. In case $\rho = 0$ a specific limit is valid, $\mathbb{J}(\square)$ exploits this limit only in this particular case:

$$\lim_{\rho \to 0} \mathcal{J}_1(\rho s_{nm}) = \frac{s_{nm}^2}{2}. \tag{2.249}$$

The real equations implemented in Guiding-Center mode derive directly from Eq. [2.148] and Eq. [2.149] but the new notation has been introduced:

$$\dot{\chi}_1 = -\frac{\partial\ \Psi_{GC_{1,1}}}{\partial \chi_2} - \frac{\partial \Psi_{GC_{2,0}}}{\partial \chi_2} - \frac{\partial \Psi_{GC_{2,2}}}{\partial \chi_2}. \tag{2.250}$$

$$\dot{\chi}_2 = +\frac{\partial \Psi_{GC_{1,1}}}{\partial \chi_1} + \frac{\partial \Psi_{GC_{2,0}}}{\partial \chi_1} + \frac{\partial \Psi_{GC_{2,2}}}{\partial \chi_1}. \tag{2.251}$$

2.13 Integration of energy

In order to understand if the computation of particles trajectory is correct, for both full orbits and guiding centres, it is possible to study the evolution of the Hamiltonian, the latter is invariant for the considered system and it is interesting to check if it is conserved or not. Therefore we are interested in analysing the trend of total energy over time for each particle. In order to do that, it is necessary to compute the trend of k over time for each particle. While tracing the trajectories of a bunch of particles, we should see a certain convergence in the trend of energy while reducing the value of the time step for both the Guiding Center and the particles since the reduction of the time step is associated with a qualitatively better computation. This kind of analysis will be the key to the choice of time-step values for both approaches.

2.13.1 Guiding centres k equation

We consider the expression of gyrokinetic Hamiltonian containing the explicit dependence on ρ:

$$H' = H_0 + \varepsilon_\delta q\langle\phi\rangle_\theta - \varepsilon_\delta^2 \frac{\eta q}{B}\frac{1}{\rho}\frac{\partial\langle\phi^2\rangle}{\partial\rho}. \tag{2.252}$$

With

$$H_0 = H_0(\rho, v_\parallel, k) = \frac{q^2 B^2}{2m}\rho^2 + \frac{1}{2}mv_\parallel^2 + k.$$

We would like to study the evolution of k:

$$\dot{k} = \Big\{k, H\Big\} = \Big(\frac{\partial k}{\partial t}\frac{\partial H}{\partial k} - \frac{\partial k}{\partial k}\frac{\partial H}{\partial t}\Big) = -\frac{\partial H}{\partial t}. \tag{2.253}$$

Time and energy are independent variables so the term $\frac{\partial k}{\partial t} = 0$. Using the expression of the modified Hamiltonian and knowing that H_0 is time independent it is possible to write that:

$$-\frac{\partial H}{\partial t} = -\varepsilon_\delta q\frac{\partial\langle\phi\rangle}{\partial t} + \varepsilon_\delta^2 \frac{\eta q}{B}\frac{1}{\rho}\frac{\partial}{\partial\rho}\frac{\partial\langle\phi^2\rangle}{\partial t}. \tag{2.254}$$

Guiding centres non-dimensional k equation

Now it is possible to start the procedure to obtain the non-dimensional equation, we introduce the non-dimensional parameters:

$$\begin{gathered}\hat{t}=2\pi\frac{t}{T}\ \ ;\ \hat{\phi}=\frac{\phi}{\phi_0}\ \ ;\ \hat{H}=\frac{H}{H_0}\ \ ;\\ \hat{B}=\frac{B}{B_0}\ \ ;\ \hat{A}=\frac{\varepsilon_\delta}{\hat{B}}\ \ ;\ \hat{\eta}=2\pi\frac{\eta}{T}\ \ ;\ \hat{\rho}=2\pi\frac{\rho}{L}.\end{gathered} \tag{2.255}$$

It is possible to proceed as usual:

$$-\frac{2\pi H_0}{T}\frac{\partial\hat{H}}{\partial\hat{t}}=-\varepsilon_\delta\frac{2\pi q\phi_0}{T}\frac{\partial\langle\hat{\phi}\rangle}{\partial\hat{t}}+\varepsilon_\delta^2\hat{\eta}\frac{Tq(2\pi)^2\phi_0^2}{2\pi BL^2T}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\frac{\partial\langle\hat{\phi}^2\rangle}{\partial\hat{t}}. \tag{2.256}$$

By simplifying we obtain:

$$-H_0\frac{\partial\hat{H}}{\partial\hat{t}}=-\varepsilon_\delta q\phi_0\frac{\partial\langle\hat{\phi}\rangle}{\partial\hat{t}}+\varepsilon_\delta^2\hat{\eta}\frac{2\pi Tq}{BL^2}\phi_0^2\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\frac{\partial\langle\hat{\phi}^2\rangle}{\partial\hat{t}}. \tag{2.257}$$

By substituting the values of ϕ_0 and H_0 we have:

$$-\frac{mL^2}{T^2}\frac{\partial\hat{H}}{\partial\hat{t}}=-\varepsilon_\delta q\frac{L^2B_0}{2\pi T}\frac{\partial\langle\hat{\phi}\rangle}{\partial\hat{t}}+\varepsilon_\delta^2\hat{\eta}\frac{2\pi qTL^4B_0^2}{(2\pi)^2BL^2T^2}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\frac{\partial\langle\hat{\phi}^2\rangle}{\partial\hat{t}}. \tag{2.258}$$

By simplifying we obtain:

$$-\frac{m}{T}\frac{\partial\hat{H}}{\partial\hat{t}}=-\varepsilon_\delta q\frac{B_0}{2\pi}\frac{\partial\langle\hat{\phi}\rangle}{\partial\hat{t}}+\varepsilon_\delta^2\hat{\eta}\frac{qB_0^2}{2\pi B}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\frac{\partial\langle\hat{\phi}^2\rangle}{\partial\hat{t}}. \tag{2.259}$$

I reorder the LHS:

$$-\frac{\partial\hat{H}}{\partial\hat{t}}=-\varepsilon_\delta q\frac{B_0T}{2\pi m}\frac{\partial\langle\hat{\phi}\rangle}{\partial\hat{t}}+\varepsilon_\delta^2\hat{\eta}\frac{qB_0^2T}{B2\pi m}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\frac{\partial\langle\hat{\phi}^2\rangle}{\partial\hat{t}}. \tag{2.260}$$

I take advantage of $B_0=\frac{B}{\hat{B}}$:

$$-\frac{\partial\hat{H}}{\partial\hat{t}}=-\frac{\varepsilon_\delta}{\hat{B}}q\frac{BT}{2\pi m}\frac{\partial\langle\hat{\phi}\rangle}{\partial\hat{t}}+\frac{\varepsilon_\delta^2}{\hat{B}^2}\hat{\eta}\frac{qBT}{2\pi m}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\frac{\partial\langle\hat{\phi}^2\rangle}{\partial\hat{t}}. \tag{2.261}$$

I use the definition of $\hat{A}$:

$$-\frac{\partial\hat{H}}{\partial\hat{t}}=-\hat{A}q\frac{BT}{2\pi m}\frac{\partial\langle\hat{\phi}\rangle}{\partial\hat{t}}+\hat{A}^2\hat{\eta}\frac{qBT}{2\pi m}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\frac{\partial\langle\hat{\phi}^2\rangle}{\partial\hat{t}}. \tag{2.262}$$

With $\frac{qB}{m}=\frac{1}{2\eta}$:

$$-\frac{\partial\hat{H}}{\partial\hat{t}}=-\hat{A}\frac{1}{2\eta}\frac{T}{2\pi}\frac{\partial\langle\hat{\phi}\rangle}{\partial\hat{t}}+\hat{A}^2\hat{\eta}\frac{1}{2\eta}\frac{T}{2\pi}\frac{1}{\hat{\rho}}\frac{\partial}{\partial\hat{\rho}}\frac{\partial\langle\hat{\phi}^2\rangle}{\partial\hat{t}}. \tag{2.263}$$

Taking advantage of the definition of $\hat{\eta}$:

$$-\frac{\partial \hat{H}}{\partial \hat{t}} = -\hat{A}\frac{1}{2\hat{\eta}}\frac{\partial \langle \hat{\phi} \rangle}{\partial \hat{t}} + \hat{A}^2 \hat{\eta} \frac{1}{2\hat{\eta}}\frac{1}{\hat{\rho}}\frac{\partial}{\partial \hat{\rho}}\frac{\partial \langle \hat{\phi}^2 \rangle}{\partial \hat{t}}. \tag{2.264}$$

The final form of the equation is:

$$-\frac{\partial \hat{H}}{\partial \hat{t}} = -\frac{\hat{A}}{2\hat{\eta}}\frac{\partial \langle \hat{\phi} \rangle}{\partial \hat{t}} + \frac{\hat{A}^2}{2}\frac{1}{\hat{\rho}}\frac{\partial}{\partial \hat{\rho}}\frac{\partial \langle \hat{\phi}^2 \rangle}{\partial \hat{t}}. \tag{2.265}$$

This equation is very easy to implement given that we have already the expression of the operator required for the implementation: In the code we have defined:

$$\hat{\psi} = \hat{A}\hat{\phi}. \tag{2.266}$$

The equation becomes:

$$-\frac{\partial \hat{H}}{\partial \hat{t}} = -\frac{1}{2\hat{\eta}}\frac{\partial \langle \hat{\psi} \rangle}{\partial \hat{t}} + \frac{1}{2}\frac{1}{\hat{\rho}}\frac{\partial}{\partial \hat{\rho}}\frac{\partial \langle \hat{\psi}^2 \rangle}{\partial \hat{t}}. \tag{2.267}$$

It is known that:

$$\langle \hat{\psi} \rangle = \Im\mathfrak{m}\Big\{ \Psi_{GC_{1,1}} e^{-jt} \Big\}. \tag{2.268}$$

And that:

$$\frac{1}{\hat{\rho}}\frac{\partial \langle \hat{\hat{\psi}}^2 \rangle}{\partial \hat{\rho}} = -\Re\mathfrak{e}\Big\{ \Psi_{GC_{2,2}} e^{-2jt} \Big\}\frac{1}{\hat{\eta}}. \tag{2.269}$$

The $\frac{1}{\hat{\eta}}$ at the end is because of the definition of $\Psi_{GC_{2,2}}$ in the code. We don't multiply for 2 because, in the definition of $\Psi_{GC_{2,2}}$, we already have that, thanks to how we develop $\langle \hat{\psi}^2 \rangle$. It is now required to calculate the expression of the derivatives:

$$\frac{\partial \langle \hat{\psi} \rangle}{\partial \hat{t}} = -\Im\mathfrak{m}\Big\{ j\ \Psi_{GC_{1,1}} e^{-jt} \Big\} = -\Re\mathfrak{e}\Big\{ \Psi_{GC_{1,1}} e^{-jt} \Big\}. \tag{2.270}$$

$$\begin{aligned} \frac{\partial}{\partial \hat{t}}\frac{1}{\hat{\rho}}\frac{\partial \langle \hat{\hat{\psi}}^2 \rangle}{\partial \hat{\rho}} = -2\Re\mathfrak{e}\Big\{ j\ \Psi_{GC_{2,2}} e^{-2jt} \Big\}\frac{1}{\hat{\eta}} = -2\Re\mathfrak{e}\Big\{ j\ \Psi_{GC_{2,2}} e^{-2jt} \Big\}\frac{1}{\hat{\eta}} = \\ = +2\ \Im\mathfrak{m}\Big\{ \Psi_{GC_{2,2}} e^{-2jt} \Big\}\frac{1}{\hat{\eta}}. \end{aligned} \tag{2.271}$$

Now we have to insert the two terms inside the equation:

$$-\frac{1}{2\hat{\eta}}\frac{\partial \langle \hat{\psi} \rangle}{\partial \hat{t}} = +\Re\mathfrak{e}\Big\{ \Psi_{GC_{1,1}} e^{-jt} \Big\}\frac{1}{2\hat{\eta}}. \tag{2.272}$$

$$+\frac{1}{2}\frac{1}{\hat{\rho}}\frac{\partial}{\partial \hat{\rho}}\frac{\partial \langle \hat{\psi}^2 \rangle}{\partial \hat{t}} = \Im\mathfrak{m}\Big\{ \Psi_{GC_{2,2}} e^{-2jt} \Big\}\frac{1}{\hat{\eta}}. \tag{2.273}$$

Then the equation to implement in the code is:

$$\hat{\dot{k}} = +\Re\mathfrak{e}\Big\{ \Psi_{GC_{1,1}} e^{-jt} \Big\}\frac{1}{2\hat{\eta}} + \ \Im\mathfrak{m}\Big\{ \Psi_{GC_{2,2}} e^{-2jt} \Big\}\frac{1}{\hat{\eta}}. \tag{2.274}$$

In order to don't have any problem with the eta at the denominator, it is possible to re-scale the equation given that we are interested in the variation of this quantity, the evaluation will be made on h/h_0. It is possible to multiply the equation for $2\hat{\eta}$ so that divergence is avoided when $\hat{\eta}$ is close to 0 and especially when we want to evaluate the first order through $\hat{\eta} = 0$.

$$2\hat{\eta}\ \dot{\hat{k}} = \Re\mathfrak{e}\Big\{\Psi_{GC_{1,1}}e^{-jt}\Big\}+2\ \Im\mathfrak{m}\Big\{\Psi_{GC_{2,2}}e^{-2jt}\Big\}. \tag{2.275}$$

2.13.2 Full orbits k equation

Passing to the equation for the full orbits the procedure is quite the same but first, it is necessary to consider the autonomized version of the system, we start from:

$$H = \frac{1}{2}m\boldsymbol{v}^{\mathbf{2}} + q\phi. \tag{2.276}$$

$$\Big\{F, G\Big\}= \frac{1}{m}\Big(\frac{\partial F}{\partial \mathbf{x}}\frac{\partial G}{\partial \mathbf{v}} - \frac{\partial F}{\partial \mathbf{v}}\frac{\partial G}{\partial \mathbf{x}}\Big)+\frac{qB}{m^2}\Big(\frac{\partial F}{\partial v} \times \frac{\partial G}{\partial v}\Big). \tag{2.277}$$

Following the autonomization we easily obtain:

$$H = \frac{1}{2}m\boldsymbol{v}^{\mathbf{2}} + q\phi + k. \tag{2.278}$$

$$\Big\{F, G\Big\}= \frac{1}{m}\Big(\frac{\partial F}{\partial \mathbf{x}}\frac{\partial G}{\partial \mathbf{v}} - \frac{\partial F}{\partial \mathbf{v}}\frac{\partial G}{\partial \mathbf{x}}\Big)+\frac{qB}{m^2}\Big(\frac{\partial F}{\partial v} \times \frac{\partial G}{\partial v}\Big)+\Big(\frac{\partial F}{\partial t}\frac{\partial G}{\partial k} - \frac{\partial F}{\partial k}\frac{\partial G}{\partial t}\Big). \tag{2.279}$$

The expression of $\dot{k}$ is the same as the previous case:

$$\dot{k} = -\frac{\partial H}{\partial t}.$$

But as we know now the expression of H is different and simpler. It is possible to calculate the derivative of H with respect to time but it is necessary to remember that $\mathbf{v}$ does not depend on time, they are independent variables so the derivative of respect to the other is zero, the result is simply:

$$-\frac{\partial H}{\partial t} = -q\frac{\partial \phi}{\partial t}. \tag{2.280}$$

Full orbits non-dimensional k equation

It is now possible to proceed with the non-dimensionalisation of this equation, By substituting all the non-dimensional parameters we obtain the:

$$-\frac{2\pi H_0}{T}\frac{\partial \hat{H}}{\partial \hat{t}} = -q\phi_0\frac{2\pi}{T}\frac{\partial \hat{\phi}}{\partial \hat{t}}. \tag{2.281}$$

With some simplifications we obtain:

$$-H_0\frac{\partial \hat{H}}{\partial \hat{t}} = -q\phi_0\frac{\partial \hat{\phi}}{\partial \hat{t}}. \tag{2.282}$$

We substitute the values of H_0 and ϕ_0 and we remember the assumption that give us $\phi = \varepsilon_\delta \phi$:

$$-\frac{mL^2}{T^2}\frac{\partial \hat{H}}{\partial \hat{t}} = -q\frac{L^2 B_0}{2\pi T}\frac{\partial \varepsilon_\delta \hat{\phi}}{\partial \hat{t}}. \tag{2.283}$$

By simplifying the equation we obtain:

$$-\frac{m}{T}\frac{\partial \hat{H}}{\partial \hat{t}} = -q\frac{B_0}{2\pi}\frac{\partial \varepsilon_\delta \hat{\phi}}{\partial \hat{t}}. \tag{2.284}$$

And then

$$-\frac{\partial \hat{H}}{\partial \hat{t}} = -\frac{qB_0 T}{2\pi m}\frac{\partial \varepsilon_\delta \hat{\phi}}{\partial \hat{t}}. \tag{2.285}$$

We take advantage of $B_0 = \frac{B}{\hat{B}}$ and of the definition of $\hat{A}$ obtaining:

$$-\frac{\partial \hat{H}}{\partial \hat{t}} = -\frac{qBT}{2\pi m}\hat{A}\frac{\partial \hat{\phi}}{\partial \hat{t}}. \tag{2.286}$$

With $\frac{qB}{m} = \frac{1}{2\eta}$:

$$-\frac{\partial \hat{H}}{\partial \hat{t}} = -\frac{1}{2\eta}\frac{T}{2\pi}\hat{A}\frac{\partial \hat{\phi}}{\partial \hat{t}}. \tag{2.287}$$

Taking advantage of the definition of $\hat{\eta}$:

$$-\frac{\partial \hat{H}}{\partial \hat{t}} = -\frac{1}{2\hat{\eta}}\frac{2\pi}{T}\frac{T}{2\pi}\hat{A}\frac{\partial \hat{\phi}}{\partial \hat{t}}. \tag{2.288}$$

We simply obtain:

$$-\frac{\partial \hat{H}}{\partial \hat{t}} = -\frac{\hat{A}}{2\hat{\eta}}\frac{\partial \hat{\phi}}{\partial \hat{t}}. \tag{2.289}$$

Given that:

$$\frac{\partial k}{\partial t} = \frac{2\pi m L^2}{T^3}\frac{\partial \hat{k}}{\partial \hat{t}} = -\frac{\partial H}{\partial t} = -\frac{2\pi m L^2}{T^3}\frac{\partial \hat{H}}{\partial \hat{t}}. \tag{2.290}$$

The procedure to obtain the non-dimensional equation is exactly the same. Now it is necessary to implement the equation inside the full orbit mode. In order to do that it is necessary to remember that like before it is valid $\hat{A}\hat{\phi} = \hat{\psi}$ so that the equation becomes:

$$-\frac{\partial \hat{H}}{\partial \hat{t}} = -\frac{1}{2\hat{\eta}}\frac{\partial \hat{\psi}}{\partial \hat{t}}. \tag{2.291}$$

In this case, we don't have any average applied on ψ so we can take advantage of:

$$\hat{\psi} = \mathfrak{Im}\Big\{\phi_c e^{-jt}\Big\}. \tag{2.292}$$

Now we look for the expression of the derivative:

$$\frac{\partial \hat{\psi}}{\partial \hat{t}} = -\mathfrak{Im}\Big\{j\,\phi_c e^{-jt}\Big\} = -\mathfrak{Re}\Big\{\phi_c e^{-jt}\Big\}. \tag{2.293}$$

So that the equation for $\dot{k}$ in f.o. mode becomes:

$$\dot{k} = \mathfrak{Re}\Big\{\phi_c e^{-jt}\Big\}\frac{1}{2\hat{\eta}}. \tag{2.294}$$

2.14 Total energy equations

Once the trend of k has been obtained over time for each trajectory it is possible to assemble it together with other contributions in order to obtain the trend over time of the total energy (Hamiltonian) associated with each particle. This quantity should be invariant for the system (particle), it is possible to check the accuracy of the computation through the analysis of this trend. Since we have two modalities, with different expressions of the Hamiltonian, it is necessary to obtain two different non-dimensional equations so that the contributions obtained from the integration of the other equations can be inserted in a coherent way. For Ions and GC respectively, the following are the expression of the Hamiltonians:

$$H = \frac{1}{2} m \boldsymbol{v^2} + q\phi + k. \tag{2.295}$$

$$H' = \mu B + \frac{1}{2} m \boldsymbol{v}_\parallel^2 + k + \varepsilon_\delta q \langle \phi \rangle_\theta - \varepsilon_\delta^2 \frac{\eta}{m} \frac{q^2}{} \frac{\partial}{\partial \mu} \langle \tilde{\phi}^2 \rangle_\theta. \tag{2.296}$$

2.14.1 Guiding centres H' Equation

The reduced expression of the modified Hamiltonian is:

$$H' = k + q\varepsilon_\delta \langle \phi \rangle - \frac{\varepsilon_\delta^2 q \eta}{B\rho} \frac{\partial}{\partial \rho} \langle \tilde{\phi}^2 \rangle. \tag{2.297}$$

As usual, we introduce non-dimensional quantities in the equation:

$$\frac{mL^2}{T^2} \hat{H} = \frac{mL^2}{T^2} \hat{k} + q\varepsilon_\delta \phi_0 \langle \hat{\phi} \rangle - \frac{\varepsilon_\delta^2 q \phi_0^2 T (2\pi)^2}{B 2\pi L^2} \frac{\hat{\eta}}{\hat{\rho}} \frac{\partial}{\partial \rho} \langle \hat{\tilde{\phi}}^2 \rangle. \tag{2.298}$$

It is possible to introduce the definition of ϕ_0:

$$\frac{mL^2}{T^2} \hat{H} = \frac{mL^2}{T^2} \hat{k} + \varepsilon_\delta \frac{qL^2 B_0}{2\pi T} \langle \hat{\phi} \rangle - \varepsilon_\delta^2 \frac{qL^4 B_0^2 T 2\pi}{B (2\pi)^2 T^2 L^2} \frac{\hat{\eta}}{\hat{\rho}} \frac{\partial}{\partial \rho} \langle \hat{\tilde{\phi}}^2 \rangle. \tag{2.299}$$

With some simplifications we are able to obtain:

$$\frac{mL^2}{T^2} \hat{H} = \frac{mL^2}{T^2} \hat{k} + \varepsilon_\delta \frac{qL^2 B_0}{2\pi T} \langle \hat{\phi} \rangle - \varepsilon_\delta^2 \frac{qL^2 B_0^2}{B 2\pi T} \frac{\hat{\eta}}{\hat{\rho}} \frac{\partial}{\partial \rho} \langle \hat{\tilde{\phi}}^2 \rangle. \tag{2.300}$$

We continue to simplify the equation:

$$\hat{H} = \hat{k} + \varepsilon_\delta \frac{qB_0 T}{2\pi m} \langle \hat{\phi} \rangle - \varepsilon_\delta^2 \frac{qB_0^2 T}{2\pi B m} \frac{\hat{\eta}}{\hat{\rho}} \frac{\partial}{\partial \rho} \langle \hat{\tilde{\phi}}^2 \rangle. \tag{2.301}$$

We take advantage of the definition of $\hat{B}$:

$$\hat{H} = \hat{k} + \frac{\varepsilon_\delta}{\hat{B}} \frac{qB}{m} \frac{T}{2\pi} \langle \hat{\phi} \rangle - \frac{\varepsilon_\delta^2}{\hat{B}^2} \frac{qB}{m} \frac{T}{2\pi} \frac{\hat{\eta}}{\hat{\rho}} \frac{\partial}{\partial \rho} \langle \hat{\tilde{\phi}}^2 \rangle. \tag{2.302}$$

Now we introduce $\hat{A}$ and η through $\frac{1}{2\eta} = \frac{qB}{m}$:

$$\hat{H} = \hat{k} + \hat{A} \frac{1}{2\eta} \frac{T}{2\pi} \langle \hat{\phi} \rangle - \hat{A}^2 \frac{1}{2\eta} \frac{T}{2\pi} \frac{\hat{\eta}}{\hat{\rho}} \frac{\partial}{\partial \rho} \langle \hat{\tilde{\phi}}^2 \rangle. \tag{2.303}$$

And then the non-dimensional η other than $\hat{\psi}$:

$$\hat{H} = \hat{k} + \frac{1}{2\hat{\eta}}\frac{2\pi}{T}\frac{T}{2\pi}\langle\hat{\psi}\rangle - \frac{1}{2\hat{\eta}}\frac{2\pi}{T}\frac{T}{2\pi}\frac{\hat{\eta}}{\hat{\rho}}\frac{\partial}{\partial\rho}\langle\hat{\tilde{\psi}}^2\rangle. \tag{2.304}$$

We remember the implemented version of the two terms:

$$\langle\hat{\psi}\rangle = \mathfrak{Im}\Big[\Psi_{gc11}\ e^{-jt}\Big]. \tag{2.305}$$

$$-\frac{\hat{\eta}}{\hat{\rho}}\frac{\partial}{\partial\rho}\langle\hat{\tilde{\psi}}^2\rangle = \Psi_{gc20} -\ \mathfrak{Re}\Big[\Psi_{gc22}\ e^{-2jt}\Big]. \tag{2.306}$$

Introducing these last definitions the equation becomes:

$$\hat{H} = \hat{k} + \frac{1}{2\hat{\eta}}\ \mathfrak{Im}\Big[\Psi_{gc11}\ e^{-jt}\Big]+\frac{1}{2\hat{\eta}}\Psi_{gc20} - \frac{1}{2\hat{\eta}}\ \mathfrak{Re}\Big[\Psi_{gc22}\ e^{-2jt}\Big]. \tag{2.307}$$

Before implementing we have to remember that our equation for $\hat{k}$ has been re-scaled, from that equation we are obtaining $2\hat{\eta}\hat{k}$ indeed. We have to adapt also this equation by multiplying it for $2\hat{\eta}$ so that we have:

$$2\hat{\eta}\ \hat{H} = 2\hat{\eta}\ \hat{k} + \mathfrak{Im}\Big[\Psi_{gc11}\ e^{-jt}\Big]+\Psi_{gc20} - \mathfrak{Re}\Big[\Psi_{gc22}\ e^{-2jt}\Big]. \tag{2.308}$$

This is the equation that should be implemented in the code.

2.14.2 Full orbits H Equation

All the contributions inside the Hamiltonian expression can vary, as we said we have the following equation:

$$H = k + \frac{1}{2}mv_{real}^2 + q\phi. \tag{2.309}$$

Before implementing it is necessary to obtain the non-dimensional equation, it is necessary to use the definition of the usual non-dimensional parameters as we did before, with the introduction of these last the equation becomes:

$$\frac{mL^2}{T^2}\hat{H} = \frac{mL^2}{T^2}\hat{k} + \frac{1}{2}\frac{mL^2}{T^2}\hat{v}_{real}^2 + q\phi_0\hat{\phi}. \tag{2.310}$$

By taking advantage of the definition of ϕ_0 we obtain:

$$\frac{mL^2}{T^2}\hat{H} = \frac{mL^2}{T^2}\hat{k} + \frac{1}{2}\frac{mL^2}{T^2}\hat{v}_{real}^2 + \frac{qL^2B_0}{2\pi T}\hat{\phi}. \tag{2.311}$$

With some simplification it is possible to obtain:

$$\hat{H} = \hat{k} + \frac{1}{2}\hat{v}_{real}^2 + \frac{qB_0T}{2\pi m}\hat{\phi}. \tag{2.312}$$

We take advantage of the definition of $\hat{B}$ and we remember the assumption on the potential, in fact, it is necessary to use $\hat{\phi} = \varepsilon_\delta\hat{\phi}$ in order to obtain:

$$\hat{H} = \hat{k} + \frac{1}{2}\hat{v}_{real}^2 + \frac{qB\varepsilon_\delta}{m\hat{B}}\frac{T}{2\pi}\hat{\phi}. \tag{2.313}$$

It is possible to use various definitions: $\hat{A} = \frac{\varepsilon_\delta}{\hat{B}}$, $\frac{qB}{m} = \frac{1}{2\eta}$ and $\hat{\eta} = \eta\frac{2\pi}{T}$:

$$\hat{H} = \hat{k} + \frac{1}{2}\hat{v}_{real}^2 + \frac{1}{2\hat{\eta}}\frac{2\pi}{T}\frac{T}{2\pi}\hat{A}\hat{\phi}. \tag{2.314}$$

As usual we introduce ψ:

$$\hat{H} = \hat{k} + \frac{1}{2}\hat{v}_{real}^2 + \frac{1}{2\hat{\eta}}\hat{\psi}. \tag{2.315}$$

Given the validity of Eq. [2.176] and Eq. [2.174] we have:

$$\hat{H} = \hat{k} + \frac{1}{2}\frac{\hat{\rho}^2}{4\hat{\eta}^2} + \frac{1}{2\hat{\eta}}\hat{\psi}. \tag{2.316}$$

We have to consider the re-scaling we have done in the equations of ions, it is valid Eq. [2.178] so that:

$$\hat{H} = \hat{k} + \frac{1}{8}\frac{\hat{\rho}_{dict}^2}{\hat{\eta}^2}(\hat{V}_x^2 + \hat{V}_y^2) + \frac{1}{2\hat{\eta}}\hat{\psi}. \tag{2.317}$$

Considering that V_x and V_y are the variables calculated from *eq ions* and available after the integration of the equations of motion.

2.15 Magnetic moment expression

After having computed the full trajectories of particles, it is possible to check if, after having moved to the gyrokinetic framework the magnetic moment is well conserved. As demonstrated previously in the manuscript, μ has become a parameter, it is constant for the system in the gyrokinetic model. This is not necessarily true for the full-orbits, the definition of the latter is:

$$\mu = \frac{m(v_x^2 + v_y^2)}{2B}. \tag{2.318}$$

It is possible to obtain the expression of the magnetic moment in the obtained reduced system through the application of the usual Lie Transform. From Appendix A.2 we recall that the sign of the Lie Transform applied to one of the independent variables has to be the opposite of one of the transformations we performed to obtain the no more θ dependent Hamiltonian, the latter is an observable in fact. Having said that it is possible to perform the usual procedure, this time we neglect terms of order three or higher, that is because in the code only terms up to order 2 are currently implemented. The procedure starts as usual:

$$\mu' = e^{\varepsilon L_s}\mu. \tag{2.319}$$

With

$$e^{\varepsilon L_s} = 1 + \varepsilon L_s + \frac{\varepsilon^2}{2}L_s^2 + O(\varepsilon^3). \tag{2.320}$$

And the usual definition of generating function:

$$S(z) = S_0 + \varepsilon_\delta S_1 + \varepsilon_\delta^2 S_2 + O(\varepsilon_\delta^3). \tag{2.321}$$

we remember the expression of the Liouville operator:

$$L_s^2(F) = \Big\{S, \Big\{S, F\Big\}\Big\}. \tag{2.322}$$

We obtain the first raw expression of the magnetic moment:

$$\mu' = e^{-\varepsilon L_s}\mu = \mu + \varepsilon\Big\{S, \mu\Big\} + \frac{\varepsilon^2}{2}\Big\{S, \Big\{S, \mu\Big\}\Big\} + O(\varepsilon^3). \tag{2.323}$$

We are able to develop the two concerning terms, the first gives us:

$$\varepsilon\Big\{S, \mu\Big\} = \varepsilon\Big\{\varepsilon_\delta S_1 + \varepsilon_\delta^2 S_2, \mu\Big\} = \varepsilon_\delta\Big\{S_1, \mu\Big\}_{\mathrm{g}} + \varepsilon_\delta^2\Big\{S_2, \mu\Big\}_{\mathrm{g}}. \tag{2.324}$$

As we can see, second order in ε_δ comes out from this term. Following the same logic, in the second component of Eq.[2.323] only one term doesn't have to be neglected:

$$\frac{\varepsilon^2}{2}\Big\{S, \Big\{S, \mu\Big\}\Big\} = \frac{\varepsilon_\delta^2}{2}\Big\{S_1, \Big\{S_1, \mu\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}. \tag{2.325}$$

The expression of magnetic will be composed by the sum of different order terms like what has been done with H':

$$\mu' = \mu + \mu^{(\mathrm{I})} + \mu^{(\mathrm{II})} + O(\varepsilon_\delta^3). \tag{2.326}$$

We develop the first-order term:

$$\mu^{(\mathrm{I})} = \varepsilon_\delta\Big\{S_1, \mu\Big\}_{\mathrm{g}}. \tag{2.327}$$

We recall the definition of H_0 we used in the procedure of H:

$$H_0 = H_0(\mu, v_\parallel, k) = \mu B + \frac{1}{2}m\boldsymbol{v}_\parallel^2 + k.$$

We apply the definition of gyro-brackets, and only one term inside the brackets survives since μ is an independent variable, we also notice that:

$$\frac{\partial \mu}{\partial \mu} = \frac{1}{B}\frac{\partial H_0}{\partial \mu}. \tag{2.328}$$

We use the RHS of the previous equation because in the gyro-brackets appears also the expression of the known term $\frac{\partial S_1}{\partial \theta}$, thanks to this it is possible to substitute from Eq.[2.48] obtaining:

$$\mu^{(\mathrm{I})} = \varepsilon_\delta\frac{q}{B}\tilde{\phi} = \frac{\varepsilon_\delta}{B}\tilde{V}. \tag{2.329}$$

The second term can be approached in the same way:

$$\mu^{(\mathrm{II})} = \varepsilon_\delta^2\Big\{S_2, \mu\Big\}_{\mathrm{g}} + \frac{\varepsilon_\delta^2}{2}\Big\{S_1, \Big\{S_1, \mu\Big\}_{\mathrm{g}}\Big\}_{\mathrm{g}}. \tag{2.330}$$

While working in the brackets, we take advantage again of $\mu = \frac{H_0}{B}$ and of the expression, we just obtained for the first-order thanks to which we obtain:

$$\mu^{(\mathrm{II})} = \frac{\varepsilon_\delta^2}{B}\Big\{S_2, H_0\Big\}_{\mathrm{g}} + \frac{\varepsilon_\delta^2}{2B}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}. \tag{2.331}$$

The first term of the right-hand side of the previous equation can be further expanded thanks to the use of Eq. [2.209] it is easy to obtain:

$$\frac{\varepsilon_\delta^2}{B}\Big\{S_2, H_0\Big\}_{\mathrm{g}} = -\frac{\varepsilon_\delta^2}{B}\Big\{S_1, \langle V\rangle\Big\}_{\mathrm{g}} - \frac{\varepsilon_\delta^2}{2B}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)}. \tag{2.332}$$

The full expression of the second-order term is the following:

$$\mu^{(\mathrm{II})} = -\frac{\varepsilon_\delta^2}{B}\Big\{S_1, \langle V\rangle\Big\}_{\mathrm{g}} - \frac{\varepsilon_\delta^2}{2B}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(\sim)} + \frac{\varepsilon_\delta^2}{2B}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}. \tag{2.333}$$

It is necessary to further develop this term through the use of the theory of linear perturbation, (use of Eq. [2.208]), we easily obtain:

$$\mu^{(\mathrm{II})} = -\frac{\varepsilon_\delta^2}{B}\Big\{S_1, \langle V\rangle\Big\}_{\mathrm{g}} + \frac{\varepsilon_\delta^2}{2B}\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(0)}. \tag{2.334}$$

Now it is necessary to introduce two definitions we introduced during the procedure to obtain H' up to the third order, the two definitions come from Eq. [2.86] and Eq. [2.211]:

$$\Big\{S_1, \langle V\rangle\Big\}_{\mathrm{g}} = \frac{1}{B}\tilde{V}\frac{\partial\langle V\rangle}{\partial\mu}.$$

$$\Big\{S_1, \tilde{V}\Big\}_{\mathrm{g}}^{(0)} = \frac{1}{B}\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle_\theta.$$

Thanks to these definitions the new expression of the second-order term becomes:

$$\mu^{(\mathrm{II})} = -\frac{\varepsilon_\delta^2}{B^2}\tilde{V}\frac{\partial\langle V\rangle}{\partial\mu} + \frac{\varepsilon_\delta^2}{2B^2}\frac{\partial}{\partial\mu}\langle\tilde{V}^2\rangle_\theta. \tag{2.335}$$

We consider $V = q\phi$ so that we have the following final expression:

$$\mu^{(\mathrm{II})} = -\frac{\varepsilon_\delta^2 q^2}{B^2}\tilde{\phi}\frac{\partial\langle\phi\rangle}{\partial\mu} + \frac{\varepsilon_\delta^2 q^2}{2B^2}\frac{\partial}{\partial\mu}\langle\tilde{\phi}^2\rangle_\theta. \tag{2.336}$$

The final expression of the magnetic moment in the gyro-kinetic framework is the following:

$$\mu' = \frac{\varepsilon_\delta}{B}\tilde{V} - \frac{\varepsilon_\delta^2 q^2}{B^2}\tilde{\phi}\frac{\partial\langle\phi\rangle}{\partial\mu} + \frac{\varepsilon_\delta^2 q^2}{2B^2}\frac{\partial}{\partial\mu}\langle\tilde{\phi}^2\rangle_\theta + O(\varepsilon_\delta^3). \tag{2.337}$$

The non-dimensional expression of this equation can be obtained through the usual procedure, the definition of the non-dimensional parameters is always the same, and the required additional definition is the one of the non-dimensional magnetic moment $\hat{\mu}$:

$$\hat{\mu} = \frac{m}{q^2 B}\mu. \tag{2.338}$$

The non-dimensional expression of the magnetic moment in the gyro-kinetic framework is:

$$\hat{\mu}' = \hat{\mu} + 2\hat{\eta}\hat{A}\hat{\tilde{\phi}} + 2\hat{\eta}^2\hat{A}^2\Big(\frac{\partial}{\partial\hat{\mu}}\langle\hat{\tilde{\phi}}^2\rangle - 2\hat{\tilde{\phi}}\frac{\partial}{\partial\hat{\mu}}\langle\phi\rangle\Big) + O(\hat{\eta}^3\hat{A}^3). \tag{2.339}$$

Chapter 3

Description of the code

3.1 Structure of the code

This chapter presents the three Python scripts with which it is possible to replicate all the simulations I performed during the work. One can then analyse the results in an output from the code with external scripts, I personally performed the post-process of the results using MATLAB but it is possible to do the same also with Python or C++. It is important to post-process the results after the main computation so that the required computational time is lower. The code has been improved throughout the internship and several modifications have been carried out. In the following pages, it will be described only the final version of the scripts and their basic functioning. In order to better understanding the implementation of equations and of other tools, it is required knowledge of the basics of programming in Python. Nevertheless, the following analysis will be quite short and readable, in addition to that one can find the description of the built-in functions on official online documentation as [Fou23], [dev23] and [com23].

3.1.1 Operation of the code

The code can work in two modalities, Guiding-Center and full-orbit, as anticipated in the previous chapter, the equations that will be integrated are different and also other parts of the code are different. Once the mode is fixed it is possible to choose three categories of simulations:

Potentials Allows one to plot the map of the potential (standard one, GC-potential, first-order GC-potential, second-order GC-potential).

Poincare Allows one to plot directly out of the simulation the Poincaré section (temporal evolution of particles' position in the transverse plane).

Diffusion Allows one to plot the trend in time of some interesting quantities and the related fittings describing how the various particles are diffusing.

One doesn't have to choose the mode while running **Potentials** given that the code won't compute the trajectories of the particles. This is the only case in which it is not required to set the Modality in which the simulation needs to be carried out. The code runs by exploiting three different scripts in Python:

`gc2d_dict.py` This file constitutes the interface with the user, where all the parameters of the simulation can be set and modified as desired and necessary.

`gc2d.py` This is the main file, it is the one that is executed by the user, and it interacts with both the other files. It contains the equations required to study the dynamics of GC and particles (Potentials computation, Governing equations for GC, G.E. for full orbit, full-orbit to GC equations, compute energy, compute magnetic moment).

`gc2d_modules.py` This file contains all the instructions that the computer must follow in order to carry out the simulations, it initialises variables, calls functions as the integrator, the latter has to integrate in time the equations in order to compute the particles' position evolution. It stores data and provides outputs in the form of `.mat` and `.txt` files as well as plotting interesting trends related to the specific category of simulation while storing pictures as `.pdf` or `.png`.

The post-process files are written in MATLAB, it has been chosen to use that language because of its versatility, and power and especially because of the interactive mode while working on `.fig` files. Except for some simple handling of vectors no computations are performed during the execution of these scripts, all the data are extracted from the output files of the Python code.

3.2 Description of `gc2d_dict.py`

As anticipated in the previous section, the user can set all the parameters value through this file, except for particular cases this is the only file that has to be modified in order to run a specific case. Modifications to the other scripts are required only in case one wants to test a new feature or a modification to the code. Given that the code is available in a GitHub repository, the folder containing all the scripts contains also `README.md`. In the latter, all the parameters are briefly described, in addition to that, if a single parameter has to be defined with some specific values it presents all the possible options so that an inexperienced user can work efficiently as well. In the following rows, all the parameters are briefly described:

- `Potential:` a string, it can be chosen as 'turbulent' or 'KMdCN', it determines the set of equations that will be integrated. Concerning this work, it has always been chosen as 'turbulent' and never changed.

- `Method:` a string, this parameter set the mode and the category of the simulation, it can be set as a combination of words:

  - `'potentials':` This category of simulation can be performed only if `Potential` = 'turbulent'. The simulation output is a `.mat` containing the previously mentioned potentials fields and a `.gif` containing the animation of the potential, it is periodic in time, and an entire period is shown.

  - `Method` finishing with `'_gc'`: set the Guiding-Center mode, the code computes the trajectories of GC using the related set of equations.
  - `Method` finishing with `'_fo'`: set the full-orbit mode, the code firstly computes full orbits using the related set of equations and then calculates the trajectories of the corresponding GC, whichever category of simulation is performed only data about GC path will be stored in the output.
  - `Method` starting with `'diffusion'`: computes the trajectories of Guiding-centres and it calculates the related diffusion coefficients. It plots the trend of r_2 for ballistic and diffusive GC. The output is composed of a `.mat` and a `.txt` file, in the latter only data about percentages of types of particles and diffusion coefficients are stored.
  - `Method` starting with `'poincare'`: computes the trajectories of Guiding-centres in the corresponding mode and it plots these in the transverse plane at the end of each time cycle (every period of the potential). The output is composed of a `.mat` file containing the trajectories data.

- `FLR`: a tuple of two strings, the possible values you can insert are `'all'`, `'pade'` and `'none'`. It can be chosen whether to take into account *FLR* effects or not in addition to considering an approximation of these. In case `'none'` is set in both the strings then the position of *guiding centes* is confused with one of the corresponding particles. It is like $\rho = 0$ so that in the first-order one has simply $\mathcal{J}_0(0) = 1$ while in the second order you use $\mathcal{J}_1(0) = s_{nm}^2/2$. In case `'all'` is set the full Bessel functions are used in the computations. In case `'pade'` is set a *padé* approximation for the *FLR effects* is used.

- `A`: float or array of floats, it is the non-dimensional amplitude of the turbulent potential, [theory: $A = \varepsilon_\delta/B$]. As anticipated by the type of value that can be assigned one can choose one or multiple values to this parameter (the same will be valid for `rho` and `eta`). This interesting case can be exploited through the use of `numpy.linspace` function. If `A` is an array, then running the code will result in running multiple simulations, one for each value of `A`. One has also to remember that if two or more parameters are set as arrays then the number of sub-simulations executed accounts for all the possible combinations of all the parameters set as arrays. This number can grow very quickly when the user decides to use `numpy.linspace` function on three parameters.

- `rho`: float or array of floats, it is the non-dimensional *Larmor radius* associated with the test particles, in case off full-orbit mode it corresponds to the *thermal Larmor radius*. Its value can be set in the same way as `A`. If the user chose to set `rho` $= 0$ then the code is automatically setting `FLR` $= ('none','none')$ and vice versa.

- `eta`: float or array of floats, it is the non-dimensional *real parameter*, it takes into account the influence of the second-order of Ψ in the equations. Its value can be set in the same way as `A`. Its value can be negative in order to study the motion of negatively charged particles.

- `Ntraj:` integer, it fixes the number of trajectories that has to be integrated.
- `Tf:` integer, it is the number of temporal cycles (potential periods) that are considered for the integration of trajectories.
- `threshold:` float, it is a value used to discriminate between trapped and untrapped particles. The recommended value is 4.
- `TwoStepIntegration:` boolean, this parameter is used in order to reduce the computational time of the simulation. If `TwoStepIntegration` $= True$ then the code computes the trajectories of all particles for the first T_{mid} temporal cycles, it removes the trapped and continues the integration only for untrapped until the end of the simulation. In case `TwoStepIntegration` $= False$ then the code computes all the trajectories for all T_f periods.
- `Tmid:` integer, it is the number of cycles for which all trajectories are computed when `TwoStepIntegration` $= True$.
- `TimeStep:` float, it is the value of time-step used by the integrator of equations in the code. Its value it's crucial for the quality of the results, recommended values:
  - 5×10^{-3} for Guiding-Center mode.
  - 5×10^{-4} for full-orbit mode.
- `check_energy:` boolean, in case it is $True$ the autonomous system is integrated, the equation of energy for the corresponding mode is integrated. If `SaveData` $=$ $True$ the `.mat` file contains also the trend of the total energy associated with each trajectory.
- `init:` string, the possible value the user can insert are `'fixed'` and `'random'`. In the second case the position of all particles is chosen randomly while in the first case, they are positioned on a grid, they are equally spaced one from the other. If `init` $=$ `'fixed'` the number of integrated trajectories can be slightly different, for obvious reasons it is rounded to `Ntraj` $= int(\sqrt{\texttt{Ntraj}^2})$.
- `SaveData:` boolean, the user can choose to save data in output from the simulations. If it is `True` the results are saved in `.mat` file while the plots are saved as `fig_extension:` files. In case `Method` is starting with `'diffusion'` the `.txt` file is always generated.
- `PlotResults:` boolean, the user can choose to plot poincaré section and r_2 trends or not. It is useful when the user has to run a big number of sub-simulations or in case the code is executed on a cluster, in this case, the graphics card may not be available resulting in an error from the code.
- `Parallelization:` tuple made of a boolean and an integer, extremely useful while running sub-simulations, if the first element of the tuple is `True` then the following integer value fixes the number of processes used to solve the simulations. This

number is equal to the number of sub-simulation performed simultaneously. In case the second element of the tuple is set as `'all'` then the computer chose the number as the highest possible. This number coincides with the number of logical cores available on the computer ($2 \times$ *number of physical cores*).

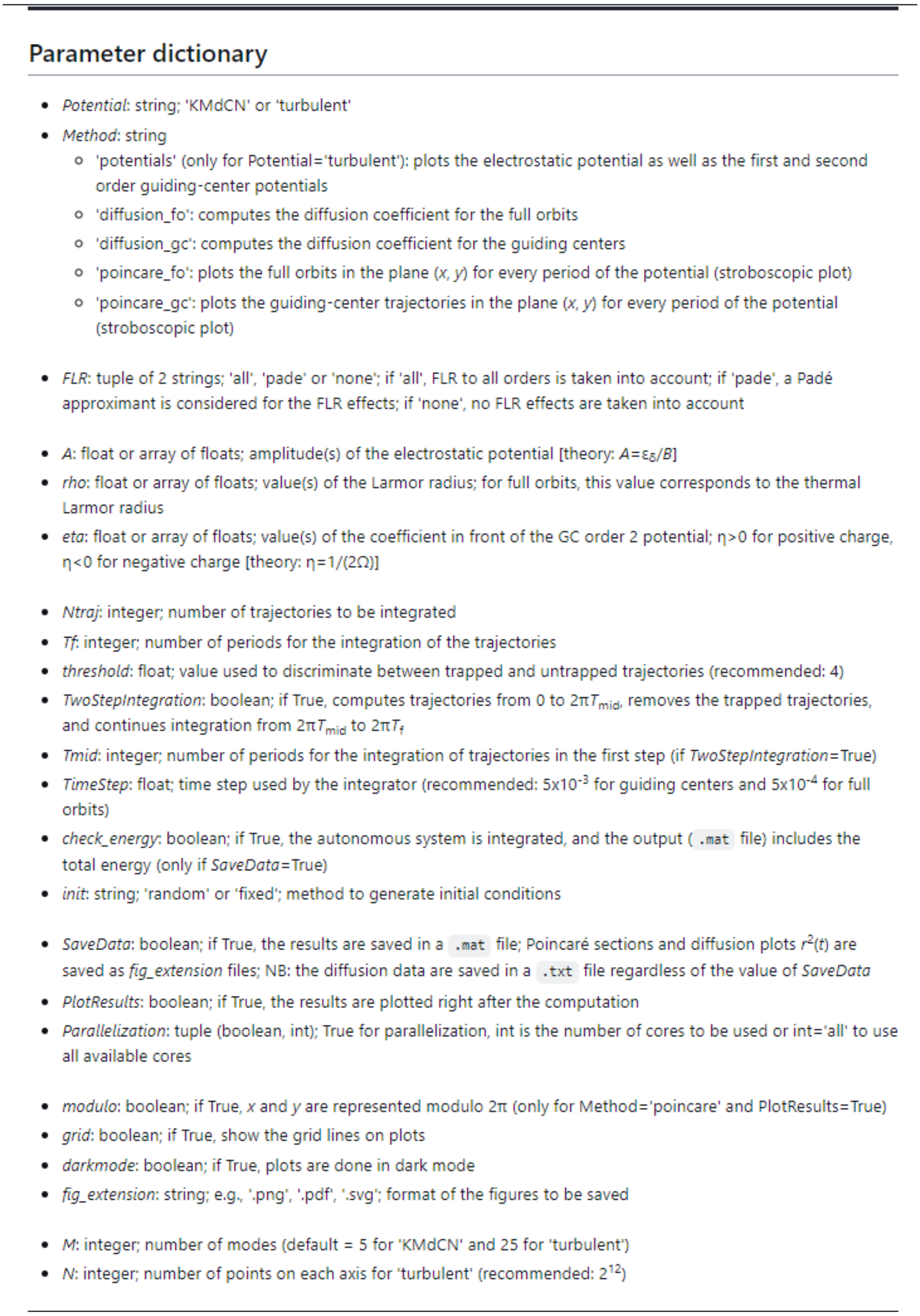
- `modulo:` boolean, if it is set to `True`, `Method` is starting with `'poincare'` and `PlotResults` = `True` then the plots are referred to the fundamental square. In case `modulo` is set to `False` the code produces standard plots but it is always possible to obtain the plots referred to the square while it is not necessarily true to the contrary.
- `grid:` boolean, when this parameter is set to `True`, grid lines are showed on plots.
- `darkmode:` boolean, when this parameter is set to `True`, plots are done in dark mode. All the plots have a black background and a different colour scheme.
- `fig_extension:` string, it coincides with the format of the figures to be saved when `SaveResults` is set to `True`. Some examples of possible formats are `'.png'`, `'.pdf'` and `'.svg'`.
- `M:` integer, number of modes of the potential. default value is 25 when `Potential` = `'turbulent'`.
- `N:` integer, it is the number of points on the axes of the transverse plane, and the grid on which the potential is calculated is generated from this spatial discretization. It is also equal to the number of Fourier coefficients associated with the numerical implemented version of the potential. The default value is 2^{12}.

Fig. 3.1 shows the Parameter dictionary, which is available at the GitHub repository.

3.3 Description of `gc2d.py`

This file is the file that is executed by the user. It contains the physics of the problem, the procedure to calculate the potentials, the governing equations integrated in order to obtain the trajectories and other useful functions. They are all exploited in order to complete some tasks that appear in `gc2d_modules.py`. In the following rows are listed all the definitions of functions and classes that may be used during execution, the most important parts of some of them will be briefly described in specific subsections:

- `Main():` function, it is the main function of the code.
- `run_case(dict):` function, it determines the use of class `GC2Dt` in the case `Potential` = `'turbulent'`.
- `___init___`(*dict*): function, it is the class used to compute all the potentials and their derivatives with respect to x, y and time.
- `eqn(t, y):` function, it contains the governing equations for both Guiding-center and Full-orbit mode that will be integrated by the integrator.



Parameter dictionary

- *Potential*: string; 'KMdCN' or 'turbulent'
- *Method*: string
  - 'potentials' (only for Potential='turbulent'): plots the electrostatic potential as well as the first and second order guiding-center potentials
  - 'diffusion_fo': computes the diffusion coefficient for the full orbits
  - 'diffusion_gc': computes the diffusion coefficient for the guiding centers
  - 'poincare_fo': plots the full orbits in the plane (*x*, *y*) for every period of the potential (stroboscopic plot)
  - 'poincare_gc': plots the guiding-center trajectories in the plane (*x*, *y*) for every period of the potential (stroboscopic plot)

- *FLR*: tuple of 2 strings; 'all', 'pade' or 'none'; if 'all', FLR to all orders is taken into account; if 'pade', a Padé approximant is considered for the FLR effects; if 'none', no FLR effects are taken into account

- *A*: float or array of floats; amplitude(s) of the electrostatic potential [theory: $A=\varepsilon_\delta/B$]
- *rho*: float or array of floats; value(s) of the Larmor radius; for full orbits, this value corresponds to the thermal Larmor radius
- *eta*: float or array of floats; value(s) of the coefficient in front of the GC order 2 potential; $\eta>0$ for positive charge, $\eta<0$ for negative charge [theory: $\eta=1/(2\Omega)$]

- *Ntraj*: integer; number of trajectories to be integrated
- *Tf*: integer; number of periods for the integration of the trajectories
- *threshold*: float; value used to discriminate between trapped and untrapped trajectories (recommended: 4)
- *TwoStepIntegration*: boolean; if True, computes trajectories from 0 to $2\pi T_{\mathrm{mid}}$, removes the trapped trajectories, and continues integration from $2\pi T_{\mathrm{mid}}$ to $2\pi T_f$
- *Tmid*: integer; number of periods for the integration of trajectories in the first step (if *TwoStepIntegration*=True)
- *TimeStep*: float; time step used by the integrator (recommended: $5\mathrm{x}10^{-3}$ for guiding centers and $5\mathrm{x}10^{-4}$ for full orbits)
- *check_energy*: boolean; if True, the autonomous system is integrated, and the output (`.mat` file) includes the total energy (only if *SaveData*=True)
- *init*: string; 'random' or 'fixed'; method to generate initial conditions

- *SaveData*: boolean; if True, the results are saved in a `.mat` file; Poincaré sections and diffusion plots $r^2(t)$ are saved as *fig_extension* files; NB: the diffusion data are saved in a `.txt` file regardless of the value of *SaveData*
- *PlotResults*: boolean; if True, the results are plotted right after the computation
- *Parallelization*: tuple (boolean, int); True for parallelization, int is the number of cores to be used or int='all' to use all available cores

- *modulo*: boolean; if True, *x* and *y* are represented modulo 2π (only for Method='poincare' and PlotResults=True)
- *grid*: boolean; if True, show the grid lines on plots
- *darkmode*: boolean; if True, plots are done in dark mode
- *fig_extension*: string; e.g., '.png', '.pdf', '.svg'; format of the figures to be saved

- *M*: integer; number of modes (default = 5 for 'KMdCN' and 25 for 'turbulent')
- *N*: integer; number of points on each axis for 'turbulent' (recommended: 2^{12})

Figure 3.1: Parameter dictionary contained in the `README.md` associated with the GitHub repository that can be found at `github.com/cchandre/Guiding-Center/`

- `compute_energy(t, sol):` function, it contains the equations used to compute the total energy of each particle for both Guiding-center and Full-orbit mode.
- `fo2gc(t, sol, order):` function, it contains the equations for computing guiding centre positions from corresponding particles' position and velocity in output from Full-orbit mode.
- `compute_mu(t, sol, order):` function, it computes magnetic moment trend from particles' trajectory and velocity in output from Full-orbit mode.

All these functions except for `main()` are part of the class `GC2Dt`.

3.3.1 Description of `main()` function

As obvious, the main function is the one that has to be executed to run a generic case, in the present code, this function is also supposed to set all the parallelization options for the execution of a generic case. As shown in Listing. (3.1) in fact, within `main()` function, the number of workers with which the simulation is going to be exploited is set. This is done firstly through the attribute of multiprocessing class `pool`, set as the desired number of workers while the attribute `map` of the new variable `pool` is the command that executes the specific case with the parameters extracted automatically from `gc2d_dict.py`. In case the parallelization option is off the specific case is executed normally with a single process.

Listing 3.1: Portion of `gc2d.py`. `main()` function brief description.

```python
import multiprocessing
def main():
        if Parallelization[0]:
                if Parallelization[1] == 'all':
                        num_cores = multiprocessing.cpu_count()
                else:
                        num_cores = min(multiprocessing.cpu_count(),
                        ↪  Parallelization[1])
                pool = multiprocessing.Pool(num_cores)
                pool.map(run_case, dict_list)
        else:
                for dict in dict_list:
                        run_case(dict)
        plt.show()
```

3.3.2 Description of `__init__(dict)` function

This function computes and stores the potential and the guiding centre potentials at both orders and their derivatives. It doesn't compute guiding centre potentials in case Full-orbit mode is exploited.

Computation of Potential through Fourier space

Given that the potential is periodic it is possible to take advantage of the frequency domain and Fourier Transform algorithms to calculate and make some operations such as the derivation, in a much more efficient way numerically speaking. As it will be seen in the code, you go from one space to another multiple times through the procedure. It is possible to define Φ_c via a truncated Fourier series. This can be done with the usual Discrete Fourier Transform but in order to make the code more efficient and faster Φ_c is very well approximated through the 2D Fast Fourier Transform algorithm and its inverse. The code shown in Listing. (3.2) starts with the definition of the grid on the physical space. The grid is made by $N \times N$ points. The mesh on the transverse plane can be limited only to the fundamental square thanks to the fact the potential is periodic in time but also in space. In order to study the motion of particles outside the fundamental square it will just be necessary to apply the modulo function in Python (indicated as %). The potential $\Phi(x, y)$ and the guiding centres potential $\Psi(\chi_1, \chi_2)$ depend on different variables but in the code these variables are used interchangeably, that's because the two potentials have the same exponential part. From a basic knowledge of Fourier Analysis it is known that the problem is associated with a Fourier space made of an equal number of wave numbers and related coefficients, in this case, they are $N \times N$. This implies that when Fourier Transform is applied to a $N \times N$ matrix in the physical space the matrix of the Fourier coefficients will have the same dimension. Naturally, in order to have a great accuracy of calculations, N should be very high. Fast Fourier Transform is exploited through the use of a specific `numpy` function, `numpy.fft.fft2` and its inverse `numpy.fft.ifft2`, these are specific to work with $2D$ matrices. It is very important to remember that the potential will have only M modes, this means that the number of coefficients related to Φ_c will be equal to M^2 given that we are considering 2 dimensions. The total number of coefficients will continue to be $N \times N$ but the one not respecting certain conditions will be annulled. It is possible to find the expression of the Fourier coefficients starting from the known expression of the desired potential, we recall Eq.(2.185):

$$\Phi_c = \sum_{\substack{n,m=1 \\ n^2+m^2 \leq M^2}}^{M} K_{nm} e^{j(nx+my)}.$$

From a basic knowledge of Fourier analysis, it is possible to write the expression of generic Fourier coefficients:

$$a_{nm} = \frac{1}{N^2} \sum_{k,l=0}^{N-1} A_{kl} e^{j(kx_n+ly_m)}, . \tag{3.1}$$

As anticipated it is necessary to reproduce Eq.(2.185) through Eq.(3.1). The shape of the latter can be reproduced directly inside Φ_c but a series of conditions must be respected in order to grant the correct result:

$$\Phi_c = \frac{1}{N^2} \sum_{n,m=0}^{N-1} c_{nm} e^{j(nx+my)}. \tag{3.2}$$

the conditions to be met are:

- if $(n, m) < (1, 1) \Rightarrow c_{nm} = 0$.
- if $(n, m) > (M, M) \Rightarrow c_{nm} = 0$.
- if $n^2 + m^2 \geq \Rightarrow c_{nm} = 0$.

In all the other cases the value of a generic c_{nm} is:

$$c_{nm} = N^2 K_{nm}. \tag{3.3}$$

The definition includes the multiplication by N^2 because we would like that c_{nm} to coincide with K_{nm} but a term $1/N^2$ will come from the Fast Fourier Transform. It is important to remember that this latter transformation has been done only in the theory, in reality, we are defining the potential in the physical space starting from the one in the Fourier space, this means that a generic potential has the characteristics of the one in Eq.(3.2) and when we do the inverse transformation (numerically this time) we have to account for the terms that are really present numerically speaking. We are multiplying all the terms for N^2 in order to let c_{nm} coincide numerically with K_{nm}. Before looking at the listing it is necessary to recall the definition of K_{nm}, in fact, the coefficients will be defined from Eq.[2.186]:

$$K_{nm} = \frac{Ae^{\varphi_{nm}}}{(n^2 + m^2)^{3/2}}. \tag{3.4}$$

Before looking at the code some references:

- `phases`: random phases of each of the 625 modes.
- `nm`: grid made of $N \times N$ points.
- `fft_phi`: coefficients c_{nm} not yet multiplied by N^2.
- `phi`: complex potential Φ_c.
- `N`: corresponds to N.
- `M`: corresponds to M.

In Listing. (3.2) the salient parts of the function `__init__(dict)` defining the potential are shown. Note also how the parameters from `gc2d_dict.py` are introduced.

Listing 3.2: Portion of `gc2d.py`.Portion of `__init__(dict)` which defines the potential.

```
import numpy as xp
from numpy.fft import fft2, ifft2, fftfreq
from gc2d_dict import dict_list

def __init__(self, dict):
    for key in dict:
        setattr(self, key, dict[key])
    self.DictParams = dict
    xp.random.seed(27)
    phases = 2 * xp.pi * xp.random.random((self.M, self.M))
    n = xp.meshgrid(xp.arange(1, self.M+1), xp.arange(1, self.M+1), indexing='ij')
    self.xy_ = 2 * (xp.linspace(0, 2 * xp.pi, self.N+1, dtype=xp.float64),)
    nm = xp.meshgrid(fftfreq(self.N, d=1/self.N), fftfreq(self.N, d=1/self.N),
    ↪  indexing='ij')
    sqrt_nm = xp.sqrt(nm[0]**2 + nm[1]**2)
    self.elts_nm = sqrt_nm, xp.angle(nm[0] + 1j * nm[1])
    fft_phi = xp.zeros((self.N, self.N), dtype=xp.complex128)
    fft_phi[1:self.M+1, 1:self.M+1] = (self.A / (n[0]**2 +
    ↪  n[1]**2)**1.5).astype(xp.complex128) * xp.exp(1j * phases)
    fft_phi[sqrt_nm > self.M] = 0
    self.phi = ifft2(fft_phi) * (self.N**2)
```

Computation of First-Order Guiding Center Potential through Fourier space

It is now time to implement the computation of the first-order guiding centre potential. Before doing that it is also necessary to introduce padé approximation. The latter is currently used in some more powerful and complex gyro-kinetic code in order to reduce the computational time whenever the code has to be tested after a modification or a bug correction. In these cases, accuracy is not essential and this kind of approximation can be done. It consists in simplifying the gyro-average operator by approximating the Bessel function through a padé expansion. Firstly, we recall the current expression of the gyro-average operator from Eq. [2.245]:

$$\langle \Box \rangle = \sum_{\substack{n,m=1 \\ n^2+m^2 \leq M^2}}^{M} \Box_{nm} \mathcal{J}_0(\rho s_{nm}) e^{j(n\chi_1 + m\chi_2)}.$$

The advantage of this padé representation is that the code has no longer to compute the full Bessel function, it uses a normal function. The padé expansion of the Bessel function is the following:

$$\mathcal{J}_{pade}(s) = \frac{1}{[1+\frac{s^2}{4}]}. \tag{3.5}$$

As shown in Fig. [3.2] the trend at the very beginning is very similar to a full Bessel function of the first kind and of order 0. As it can be clearly seen the argument can't have a high value otherwise the approximation is not acceptable. This representation is asymptotically correct in the limit of $\rho s_{nm} << 1$ that gives us an interesting condition in which it is valid $\mathcal{J}_0(\rho s_{nm}) \approx \mathcal{J}_{pade}(\rho s_{nm})$. In case of large arguments, $\rho s_{nm} \Rightarrow \infty$ then we have an unfortunate condition in which we have $\mathcal{J}_0(\rho s_{nm}) \approx (2\pi\rho s_{nm})^{1/2} \cos{(\rho s_{nm} - \pi/4)}$ while $\mathcal{J}_{pade}(\rho s_{nm}) \approx 4/(\rho s_{nm})^2$. It can be concluded that, as a drawback, filtering of small scales is required, the number of modes can't be very high otherwise very little and small structures will be overestimated and will have not realistic amplitudes, that's

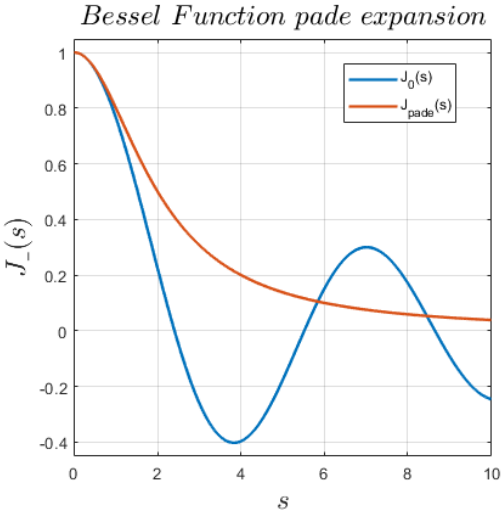


Figure 3.2: Bessel function of the first kind of order zero padé expansion comparison with full Bessel function.

where the unacceptable approximation affects the results. In the code, it will be shown how to implement this tool, it has been used in order to check if also the results from the present code can be affected by such an approximation considering the present number of modes (25).

Going back to the numerical implementation of the full First-order guiding centre potential, the procedure consists of accounting for the gyro-average operator as a coefficient. This coefficient is simply multiplied by `fft_phi`, the coefficients that represent the normal complex potential. Before recalling the expression of the concerning term from Eq. [2.209], it is important to note that, also in this case, we are simply defining $\Psi_{GC_{1,1}}$ directly through an inverse Fast Fourier Transform, we don't generate again the potential from the just obtained `phi`, this is much more efficient.

$$\Psi_{GC_{1,c}} = \sum_{\substack{n,m=1\\ n^2+m^2\leq M^2}}^{M} K_{nm}\mathcal{J}_0(\rho s_{nm})e^{j(n\chi_1+m\chi_2)}.$$

All the above-mentioned features can be found in Listing. (3.3), and the imported libraries are the same imported for Listing. (3.2).

Listing 3.3: Portion of `gc2d.py`.Portion of `__init__(dict)` which defines the First-order guiding center potential.

```
if self.FLR[0] == 'all':
    flr1_coeff = jv(0, self.rho * sqrt_nm)
elif self.FLR[0] == 'pade':
    flr1_coeff = 1 / (1 + self.rho**2 * sqrt_nm**2 / 4)
else:
    flr1_coeff = 1
self.phi_gc1_1 = ifft2(flr1_coeff * fft_phi) * (self.N**2)
```

Computation of First-Order Guiding Center Potential through Fourier space

Like what has been done in the previous paragraph we first recall the expression of the concerning terms from Eq. [2.243] and Eq. [2.244] other than the operator of the gyro-average coming from Eq. [2.248]:

$$\Psi_{GC_{2,0}} = +\frac{\eta}{2}\Big(-\mathbb{J}(|\Phi_c|^2)+\langle\Phi_c\rangle\mathbb{J}(\Phi_c^*)+\langle\Phi_c^*\rangle\mathbb{J}(\Phi_c)\Big)e^{-0jt}.$$

$$\Psi_{GC_{2,2}} = +\frac{\eta}{2}\Re\mathfrak{e}\Big\{\big[\mathbb{J}(\Phi_c^2)-2\langle\Phi_c\rangle\mathbb{J}(\Phi_c)\big]e^{-2jt}\Big\}.$$

$$\mathbb{J}(\Box) = -\sum_{\substack{n,m=1\\ n^2+m^2\le M^2}}^{M} \Box_{nm}\frac{s_{nm}}{\rho}\mathcal{J}_1(\rho s_{nm})e^{j(n\chi_1+m\chi_2)}.$$

The padé expansion of the Bessel function this time is the following:

$$\mathcal{J}_{1\ pade}(s) = -\frac{s_{nm}^2/2}{[1+\frac{(\rho s_{nm})^2}{8}]}. \tag{3.6}$$

The procedure in the code remarks the one for the first-order, obviously the variables this time accounts for multiple terms containing also the first term indeed. One additional operation is the computation of the derivatives that will be necessary for the governing equations. Given that complex potentials have been used the computation is really simple through the use of the function `derivs`. It is also important to remember that, despite the fact the potentials are periodic, in order to take advantage of the Fourier Analysis we wouldn't have considered the full period, the potential produced from `ifft2` function does not account for the top and right sides of the fundamental square, that's why we have some pad operation inside the code. The function `pad` in the code is redefined in order to add one element at the end of each row and on the top of each column, the values of these are equal to the one on the opposite side. After this operation, we have the full periodic potential available. In Listing. (3.4) it is showed the portion of the code that does all the above-mentioned operations.

Listing 3.4: Portion of `gc2d.py`.Portion of `__init__(dict)` which defines the Second-order guiding center potentials.

```
if self.FLR[1] == 'all' and (self.rho != 0):
    flr2_coeff = -sqrt_nm * jv(1, self.rho * sqrt_nm) / self.rho
elif self.FLR[1] == 'pade' and (self.rho != 0):
    flr2_coeff = -(sqrt_nm**2 / 2) / (1 + self.rho**2 * sqrt_nm**2 / 8)
else:
    flr2_coeff = -sqrt_nm**2 / 2
```

```
self.flr2 = lambda psi: ifft2(fft2(psi) * flr2_coeff)
self.phi_gc2_0 = self.eta * (2 * self.phi_gc1_1 * self.flr2(self.phi.conj()) -
↪  self.flr2(xp.abs(self.phi)**2)).real / 2
self.phi_gc2_2 = self.eta * (2 * self.phi_gc1_1 * self.flr2(self.phi) -
↪  self.flr2(self.phi**2)) / 2
stack = (*self.derivs(self.phi_gc1_1), *self.derivs(self.phi_gc2_0),
↪  *self.derivs(self.phi_gc2_2))
self.Dphi = xp.moveaxis(xp.stack(stack), 0, -1)
```

In case the code is running a simulation computing the trajectory of particles it is not necessary to compute guiding centre potentials but only Φ, its derivatives are required, they are calculated inside this function through the use of `derivs` as well.

3.3.3 Description of `eqn(t, y):` function

As anticipated previously this function contains the governing equations of both modalities. Obviously, it uses the potential computed by the precedent function in order to calculate how the particles will move in time. Given that the derivatives are calculated as the derivatives of a complex potential, in order to obtain their values in all the periods it is just needed to multiply for the corresponding exponential accounting for time dependency. In the function, there is a common part, the one responsible for the unpacking of the vector `y` passed to the code when the function is called and the one that applies the modulo function on the coordinates of the trajectory in order to get advantage from the periodicity of the fundamental square. It is also necessary to call a first interpolation in order to know the value of the derivatives on the exact position of the particles from the values available on the potential grid. This built-in function is crucial for the functioning of the code but it is also kinda slow, as it will be seen later in the code, this function is called the least number of times while giving her multiple arguments. It has been noticed that the slowest part of the whole operation is the calling. In Listing. (3.5) it is showed the portion of the code that does all the above-mentioned operations.

Listing 3.5: Portion of `gc2d.py`. Portion of `eqn(t, y)` common to both modes.

```
def eqn(self, t, y):
    vars = xp.split(y, self.dim)
    r = xp.moveaxis(xp.asarray(vars[:2]) % (2 * xp.pi), 0, -1)
    fields = xp.moveaxis(interpn(self.xy_, self.Dphi, r), 0, 1)
    dphidx, dphidy = fields[:2]
```

Implementation of the governing equations for Guiding Center mode

Firstly it is necessary to recall the equations for guiding centre mode from Eq. [2.250] and Eq. [2.251], both obtained in Sec. 2.8:

$$\dot{\chi}_1 = -\frac{\partial\ \Psi_{GC_{1,1}}}{\partial \chi_2} - \frac{\partial \Psi_{GC_{2,0}}}{\partial \chi_2} - \frac{\partial \Psi_{GC_{2,2}}}{\partial \chi_2}.$$

$$\dot{\chi}_2 = +\frac{\partial \Psi_{GC_{1,1}}}{\partial \chi_1} + \frac{\partial \Psi_{GC_{2,0}}}{\partial \chi_1} + \frac{\partial \Psi_{GC_{2,2}}}{\partial \chi_1}.$$

The values of the derivatives are already available, despite this, one has to remember that up to now, we only computed complex potentials, it is necessary to compute the imaginary part of these ones other than multiplying for the corresponding exponential which introduces the temporal dependence. In case it is requested to compute the trend of the energy k over time, it is necessary to integrate (together with the governing equations) the equation for energy, we recall its expression from Eq. [2.274], it has been obtained in Sec. 2.13.1:

$$\dot{\hat{k}} = +\mathfrak{Re}\Big\{\Psi_{GC_{1,1}} e^{-jt}\Big\}\frac{1}{2\hat{\eta}} + \mathfrak{Im}\Big\{\Psi_{GC_{2,2}} e^{-2jt}\Big\}\frac{1}{\hat{\eta}}.$$

While looking at the Listing. (3.6) it is easy to recognize all the portions of the equations. The main difference is that the equations have been split in order to compute only required quantities when one wants to account only for First-order terms. In case also second-order terms need to be computed then a simple sum is performed.

Listing 3.6: Portion of `gc2d.py`. The portion of `eqn(t, y):` containing the implemented governing equations of guiding centre mode together with the equations for energy (k).

```python
if self.Method.endswith('_gc'):
    dy_gc = xp.concatenate((-(dphidy * xp.exp(-1j * t)).imag, (dphidx * xp.exp(-1j * t)).imag), axis=None)
    if self.GCorder == 1:
        if not self.check_energy:
                return dy_gc
        phi = fields[2]
        dk = (phi * xp.exp(-1j * t)).real
        return xp.concatenate((dy_gc, dk), axis=None)
    dphidx_0, dphidy_0, dphidx_2, dphidy_2 = fields[2:6]
    dy_gc += xp.concatenate((-dphidy_0.real + (dphidy_2 * xp.exp(-2j * t)).real, dphidx_0.real - (dphidx_2 * xp.exp(-2j * t)).real), axis=None)
    if self.GCorder == 2:
        if not self.check_energy:
                return dy_gc
        phi1, phi2 = fields[6:8]
        dk = (phi1 * xp.exp(-1j * t)).real + 2 * (phi2 * xp.exp(-2j * t)).imag
        return xp.concatenate((dy_gc, dk), axis=None)
```

Implementation of the governing equations for Full-orbit mode

The governing equations for full orbits can be recovered from Eq. [2.170] and Eq. [2.171], both obtained in Sec. 2.9:

$$\frac{\partial \hat{V}_x}{\partial \hat{t}} = -sign(\hat{\eta})\frac{1}{\hat{\rho}}\frac{\partial \hat{\Phi}}{\partial \hat{x}} + \frac{1}{2|\hat{\eta}|}\hat{V}_y.$$

$$\frac{\partial \hat{V}_y}{\partial \hat{t}} = -sign(\hat{\eta})\frac{1}{\hat{\rho}}\frac{\partial \hat{\Phi}}{\partial \hat{y}} - \frac{1}{2|\hat{\eta}|}\hat{V}_x.$$

As it can be clearly seen, these equations require only Φ and its partial derivatives in space. These equations will be integrated giving us the value of V_x and V_y for each particle for each instant of time. As we remember from Sec.2.9.2 these velocities need to be re-scaled in order to obtain the real values. We recall the following relation from Eq. [2.180] while it is also valid Eq. [2.177]:

$$\hat{v}_{\perp\ real} = \hat{V}_{\perp\ calc}\hat{v}_{\perp} \quad ; \ \hat{v}_{\perp} = \frac{\hat{\rho}_{dict}}{2|\hat{\eta}_{dict}|}.$$

Once we apply these relations to the velocities calculated from Eq. [2.170] and Eq. [2.171] we have the real values of velocities, obviously, these can be integrated as well in order to obtain the coordinates of particles over time. At the end of each integration, we have the new values of velocities on the transverse plane other than the updated position. It is also implemented the equation for computing the energy k of each particle over time. We recall the equation from Eq. [2.294], obtained in Sec. 2.13.2:

$$\dot{k} = \mathfrak{Re}\Big\{\phi_c e^{-jt}\Big\}\frac{1}{2\hat{\eta}}.$$

In Listing. (3.7) it is possible to observe the implemented above-mentioned equations.

Listing 3.7: Portion of gc2d.py. The portion of eqn(t, y): containing the implemented governing equations for full orbit mode together with the equations for energy (k).

```
elif self.Method.endswith('_fo'):
    if self.eta == 0 or self.rho == 0:
    raise ValueError("Eta or Rho cannot be zero for full orbits")
    vx, vy = vars[2:4]
    dvx = -(dphidx * xp.exp(-1j * t)).imag / self.rho * xp.sign(self.eta) + vy / (2 * self.eta)
    dvy = -(dphidy * xp.exp(-1j * t)).imag / self.rho * xp.sign(self.eta) - vx / (2 * self.eta)
    d_ = xp.concatenate((self.rho / (2 * xp.abs(self.eta)) * vx, self.rho / (2 * xp.abs(self.eta)) * vy, dvx, dvy), axis=None)
    if not self.check_energy:
        return d_integr
    phi = fields[2]
    dk = (phi * xp.exp(-1j * t)).real / (2 * self.eta)
    return xp.concatenate((d_, dk), axis=None)
```

3.3.4 Description of compute_energy(t, sol) function

The purpose of this function is to compute the total energy trend of each particle over time. This equation doesn't need to be integrated and it works by exploiting vectors of variables already calculated in the previous integration, extracted from the vector `sol`, this function is called kinda late in the code. Obviously, the equations for guiding centre and full orbit mode are different, the first accounts for terms deriving from guiding centre potentials that obviously are not required in the second one. In Sec.2.14 we called these equations H' and H equations, we are actually computing the values of the modified and of the standard Hamiltonian for each particle over time. We recall the two equations respectively from Eq. [2.308] and Eq. [2.317]

$$2\hat{\eta}\ \hat{H}' = 2\hat{\eta}\ \hat{k} + \mathfrak{Im}\Big[\Psi_{gc11}\ e^{-jt}\Big] + \Psi_{gc20} - \mathfrak{Re}\Big[\Psi_{gc22}\ e^{-2jt}\Big].$$

$$\hat{H} = \hat{k} + \frac{1}{8}\frac{\hat{\rho}^2_{dict}}{\hat{\eta}^2}(\hat{V}_x^2 + \hat{V}_y^2) + \frac{1}{2\hat{\eta}}\hat{\psi}.$$

Also in this case, the implemented version of Eq. [2.308] is split in two, the first part will account only for the terms coming from the First-order while the second one will add the terms coming from the Second-order. This is done again to let the user be able of computing only the equations for the First-Order in order to make some tests on the influence of the second-order terms. One additional feature of the implemented version is that this time, the interpolation is called multiple times without any problem. This is done on purpose, the present equation won't be integrated, this means that a single interpolation function won't be called hundreds of millions of times but only a single time, it is obvious that it won't affect the speed of the code unlike before. This time in order to redirect the function in computing the value of `h`, for a mode other than the other, as shown in Listing. (3.8), it is used the scalar `dim`. It has been computed inside function `___init___` and is modified according to which mode has been executed other than if `check_energy` = `True`, the dimensions of some vectors, commonly used in both modes, like `sol`, are different.

Listing 3.8: Portion of `gc2d.py`. Portion of `compute_energy(t, sol)` containing the implemented equations for H' and H.

```python
def compute_energy(self, t, *sol):
    r = xp.moveaxis(xp.asarray(sol[:2]) % (2 * xp.pi), 0, -1)
    k = sol[-1]
    if self.dim <= 3:
        phi_1 = interpn(self.xy_, self.pad(self.phi_gc1_1), r)
        h = k + (phi_1 * xp.exp(-1j * t)).imag
        if self.GCorder == 1:
                return h
        elif self.GCorder == 2:
                phi_0 = interpn(self.xy_, self.pad(self.phi_gc2_0), r)
                phi_2 = interpn(self.xy_, self.pad(self.phi_gc2_2), r)
                h += phi_0 - (phi_2 * xp.exp(-2j * t)).real
                return h
    else:
        vx, vy = sol[2:4]
        h = k + self.rho**2 / (8 * self.eta**2) * (vx**2 + vy**2) +
        ↪  (interpn(self.xy_, self.pad(self.phi), r) * xp.exp(-1j * t)).imag / (2 *
        ↪  self.eta)
        return h
```

3.3.5 Description of `fo2gc(t, sol, order)` function

The purpose of this function is to compute the position of the correspondent guiding centre from the one of a particle, it is necessary to know the position and the velocity (in the transverse plane) of the starting particle over time in order to compute the trajectory of the respective guiding centre. Again, this function doesn't require to be integrated but works with the data coming out from the integration of the governing equations for full orbit mode. The equations are obtained from the change of coordinates exploited in Sec. 2.3, we recall some useful relations from Eq. [2.11], [2.12], [2.14] and [2.15]:

$$\rho = \frac{m}{|q|B}\sqrt{v_x^2 + v_y^2}.$$

$$\theta = \tan^{-1}\left(\frac{v_x}{v_y}\right).$$

$$x = \chi_1 + \rho\cos\theta.$$

$$y = \chi_2 - \rho\sin\theta.$$

These equations are simply implemented without any peculiarity as shown in Listing. [3.9]:

Listing 3.9: Portion of `gc2d.py`. Portion of `fo2gc(t, sol, order)` containing the implemented concerning equations.

```
def fo2gc(self, t, *sol, order=1):
    x, y, vx, vy = sol[:4]
    v = vy + 1j * vx
    theta, rho = xp.pi + xp.angle(v), self.rho * xp.abs(v)
    x_gc, y_gc = x - rho * xp.cos(theta), y + rho * xp.sin(theta)
    if order <= 1:
        return x_gc, y_gc
```

3.3.6 Description of `compute_mu(t, sol)` function

We already obtained the expression of the magnetic moment in Sec. 2.15, the resulting expression can be recalled from Eq. [2.339]:

$$\hat{\mu}' = \hat{\mu} + 2\hat{\eta}\hat{A}\hat{\tilde{\phi}} + 2\hat{\eta}^2\hat{A}^2\Big(\frac{\partial}{\partial\hat{\mu}}\langle\hat{\tilde{\phi}}^2\rangle - 2\hat{\tilde{\phi}}\frac{\partial}{\partial\hat{\mu}}\langle\phi\rangle\Big) + O(\hat{\eta}^3\hat{A}^3)$$

Also in this case, as shown in Listing. [3.10], the implemented version of the equation is split for the first and second order like did with `compute_energy` function this is done to analyse the influence of the second order terms.

Listing 3.10: Portion of `gc2d.py`. The portion of `compute_mu(t, sol)` containing the implemented concerning equation.

```python
def compute_mu(self, t, *sol, order=1):
    x, y, vx, vy = sol[:4]
    r = xp.moveaxis(xp.asarray((x, y)) % (2 * xp.pi), 0, -1)
    mu = self.rho**2 * (vx**2 + vy**2) / 2
    if order == 0:
        return mu
    r_gc = xp.moveaxis(xp.asarray(self.fo2gc(t, *sol, order=1)) % (2 * xp.pi), 0, -1)
    phi_c = interpn(self.xy_, self.pad(self.phi), r)
    mu += 2 * self.eta * ((phi_c - interpn(self.xy_, self.pad(self.phi_gc1_1), r_gc)) * xp.exp(-1j * t)).imag
    if order == 1:
        return mu
    phi_gc = interpn(self.xy_, self.pad(self.flr2(self.phi)), r_gc)
    phi_2_0 = 2 * phi_c * phi_gc.conj() - interpn(self.xy_, self.pad(self.flr2(xp.abs(self.phi)**2)), r_gc)
    phi_2_2 = 2 * phi_c * phi_gc - interpn(self.xy_, self.pad(self.flr2(self.phi**2)), r_gc)
    mu += self.eta**2 * (phi_2_2 * xp.exp(-2j * t) - phi_2_0).real
    if order == 2:
        return mu
    raise ValueError("compute_mu not available at order {}".format(order))
```

3.4 Description of `gc2d_modules.py`

As anticipated, the present file is the one leading the simulation through all the passages required in order to complete all the desired tasks. It contains all the instructions that will be followed like integrating the equations, calling useful functions and producing plots as required by the user through `gc2d_dict.py`.

3.4.1 Initialization of key vectors

The first important task reserved for this file is initializing vectors required for the integration of the governing equations, for both modalities. First, it is necessary to generate the vector `t_eval`, it contains the value of time at the end of each cycle, and the position of the particles is stored only at these instants, not at the end of each time-step, otherwise, the memory required would be enormous. After this, the vector containing the initial position of particles is defined, and a different procedure is followed in case the user decides to use random initial positions other than fixing the particles on a grid. The dimension of the vector `y0` is different in case the user chose to execute full-orbit mode other than computing the energy for each particle. In this last case, the initial energy of each particle is set to 0, in fact, we are not interested in the real value of the latter, only its evolution over time and conservation. In order to study the full dynamics, each particle requires an initial velocity on the transverse plane. We recall some important relations from Sec.2.9.2 Specifically the Eq. [2.179], [2.180] and [2.177]:

$$\hat{v}_{\perp 0_{real}} = \frac{\hat{\rho}_{dict}}{2|\hat{\eta}_{dict}|}; \qquad \hat{v}_{\perp\ real} = \hat{V}_{\perp\ calc}\hat{v}_{\perp}; \qquad \hat{v}_{\perp} = \frac{\hat{\rho}_{dict}}{2|\hat{\eta}_{dict}|}.$$

It is clear that $\hat{V}_{\perp\ calc}$ has to be equal to 1 initially. Since the distribution of the velocity along x and y axes is not important, in fact, it is only required that:

$$\hat{V}_{\perp\ calc} = \sqrt{\hat{V}_x^2 + \hat{V}_y^2} = 1.$$

we can simply take advantage of some random phases `phi_perp` while defining

$$V_{x_0} = \cos\left(\texttt{phi\_perp}\right); \quad V_{y_0} = \sin\left(\texttt{phi\_perp}\right).$$

so that the previous equation is valid given that:

$$\sqrt{cos(\alpha)^2 + \sin\left(\alpha\right)^2} = 1.$$

In Listing. [3.11] one can find the initialization of all the above-mentioned vectors that will be further used later in the code.

Listing 3.11: Portion of `gc2d_modules.py` containing the initialization of some essential vectors.

```python
elif case.Method in ['poincare_gc', 'poincare_fo', 'diffusion_gc', 'diffusion_fo']:
    t_eval = 2 * xp.pi * xp.arange(0, case.Tf + 1)
    if case.init == 'random':
        y0 = 2 * xp.pi * xp.random.rand(2 * case.Ntraj)
    elif case.init == 'fixed':
        y_vec = xp.linspace(0, 2 * xp.pi, int(xp.sqrt(case.Ntraj)), endpoint=False)
        y_mat = xp.meshgrid(y_vec, y_vec)
        y0 = xp.concatenate((y_mat[0], y_mat[1]), axis=None)
        case.Ntraj = int(xp.sqrt(case.Ntraj))**2
    if case.Method.endswith('_fo'):
        phi_perp = 2 * xp.pi * xp.random.rand(case.Ntraj)
        y0 = xp.concatenate((y0, xp.cos(phi_perp), xp.sin(phi_perp)), axis=None)
    if case.check_energy:
        y0 = xp.concatenate((y0, xp.zeros(case.Ntraj)), axis=None)
    start = time.time()
```

Integration of the governing equations through `scipy.integrate.solve_ivp`

The integration is performed through the use of `scipy.integrate.solve_ivp`. The latter is a solver for *initial values problems*, its functioning is pretty simple, it is necessary to furnish: the equation to be integrated, temporal starting and ending points, the initial value of the solution, the instants at which we want to save data in the solution vector, the maximum value of the time-step and other two parameters, those are supposed to evaluate if the current time-step is too large in order to grant a good quality integration. In the present case, these values are fixed to 1 in order to don't let them vary the temporal discretization, it has to be fixed over all the integration. The equation that is supposed to be integrated in our case is `eqn(t, y)`.

3.4.2 `Trajectory` class

This class is very important in the code, it establishes if a particle is considered `untrapped` or `trapped` other than distinguishing between sub-diffusive and super-diffusive particles. In the code, we refer to sub-diffusive particles as to `diffusive` while to the super-diffusive ones as to `ballistic`. `Trajectory` class stores and computes essential quantities that can be accessed through the attributes of the class itself, this is really convenient and aimed at object programming. In case the category of the simulation is "diffusive" then it computes also diffusion coefficients and some fittings other than a couple of statistical indicators for `diffusive` and `ballistic` particles only, there is no point in doing it for the `trapped` ones. Another feature is that it is also able to set the colour the particle will have in the plots if the user decides to use it.

Function `define_type` description

This function's main task is distinguishing various types of particles. The operation provides to first identify trapped particles, immediately after that, ballistic particles are identified. The remaining particles are considered diffusive. The identification of the first two types is exploited through the check on some criteria. Despite this, the function can work also in an alternate way, whose target is distinguishing between trapped and simply untrapped. This is done through the fact that ballistic particles are untrapped as well, the strategy is then to output ballistic and diffusive as untrapped. This alternative mode is performed when calling the function with a specific argument, the operation always requires that the type of particle to be distinguished is provided. In Listing. [3.12] are shown all the possible ways of calling this function:

Listing 3.12: Portion of `gc2d_modules.py` containing some examples of `define_type` function call.

```python
Trapped = Trajectory(case, t_eval[:case.Tmid+1], sol_, 'trapped')
Untrapped = Trajectory(case, t_eval[:case.Tmid+1], sol_, 'untrapped')

Trapped = Trajectory(case, t_eval, sol_, 'trap')
Diffusive = Trajectory(case, t_eval, sol_, 'diff')
Ballistic = Trajectory(case, t_eval, sol_, 'ball')

class Trajectory:
    def __init__(self, case, t, sol, type):
        sol_gc = sol if case.Method.endswith('_gc') else case.fo2gc(t, *sol)
        if type in ['trap', 'diff', 'ball']:
                type_ = define_type(case, sol_gc, output=['trap', 'diff', 'ball'])
        elif type in ['trapped', 'untrapped']:
                type_ = define_type(case, sol_gc, output=['trapped', 'untrapped',
                ↪  'untrapped'])
```

The first couple of calls is performed during the execution while `TwoStepIntegration` = `True`, at the end of the first integration of governing equations. As it is possible to see, the class redirects the call of `define_type` in different ways, depending on the `type` argument. Obviously, it has to check on some criteria to identify the different types of particles. In brief, first, we need information about how much a single particle has moved during the considered interval of time. This is done simply by calculating for each axe (x and y), the distance between the maximum and the minimum value of the specified coordinate. We obtain a sort of aspect ratio for each trajectory. It is possible to do this by the exploiting *peaktopeak* Python built-in function, it calculates, as an attribute, the difference between the maximum and minimum inside a matrix along the specified axe (`sol.ptp(axis = axis)`), naturally the choice of the axe is done so that it performs the computation for a specific trajectory over all the instants of time considered. Once having done that, two criteria are checked by the `define_type` function. Imagine composing a rectangle (for each trajectory) whose sides are made with the `.pkp` attributes of each trajectory. The resulting rectangle perfectly contains the trajectory. The function computes the diagonal of this rectangle as well as the numerical value of the shape factor (ratio between lengths of sides) and its inverse. These values are finally used in order to distinguish the type of each particle associated with the corresponding trajectory. A particle is considered trapped if the value of the diagonal previously computed is under a certain threshold. This criterion can be considered as a geometrical one, the value of

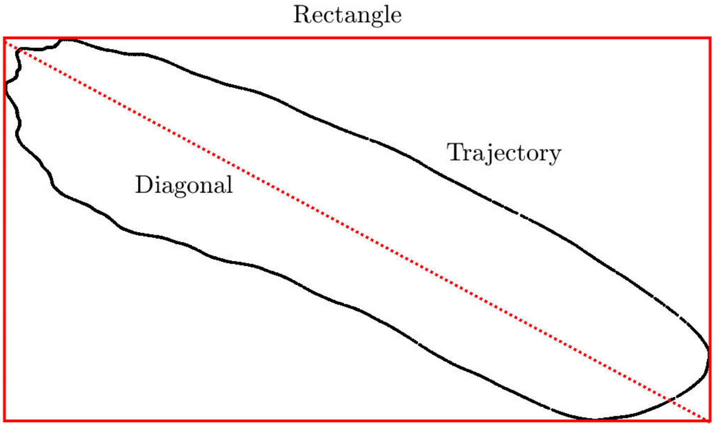


Figure 3.3: Sketch of the geometrical criterion used to distinguish trapped and untrapped particles.

the threshold was discovered as a value for which the percentage of trapped particles remains constant while varying the value of the threshold itself. This has been done in previous work on this topic [Sta22]. A check on the numerical aspect ratio is done for ballistic particles instead, it has been observed that ballistic particles don't have an isotropic behaviour while diffusing, their movement is preferably in one specific direction. This cause to the imaginary rectangle, to be highly elongated resulting in a very high numerical value of the shape factor. Naturally, the check is done also on the inverse of this value, in case the main direction of ballistic particles is along the other axe. The threshold value in this check is chosen as the squared value of the previous one. Also for this criterion, the distinction is based on geometrical analysis. In Listing. [3.13] are showed all the checks on various criteria resulting in distinguishing a specific type of trajectory.

Listing 3.13: Portion of `gc2d_modules.py` containing `define_type` function.

```python
def define_type(case, sol, axis=1, output=[0, 1, 2]):
    vec = xp.repeat(output[1], sol[0][:, 0].size)
    delta = xp.asarray([el.ptp(axis=axis) for el in sol[:2]])
    vec[xp.sqrt(xp.sum(delta**2, axis=0)) <= case.threshold] = output[0]
    untrapped = xp.sqrt(xp.sum(delta, axis=0)) > case.threshold
    vec[xp.all((delta[0] / delta[1] > case.threshold * 2, untrapped), axis=0)] = output[2]
    vec[xp.all((delta[1] / delta[0] > case.threshold * 2, untrapped), axis=0)] = output[2]
    return vec
```

Computation of $\mu(t)$ and $H(t)$

Once a certain number of particles have been classified as a specific type some previously introduced quantities are computed for that specific typology. In case full-orbit mode is executed it is necessary to obtain the correspondent guiding centre trajectories through `fo2gc`, just after that, also the magnetic moment of each particle over time is calculated through `compute_mu` function. In case the user is interested in plotting the trend of the total energy over time, in the form of relative error with respect to the initial value of the energy, the concerning quantity is obtained through the call of `compute_energy`, nevertheless, the code store only the trend of the relative errors over time. Immediately after this, the colour for the specific type of particle is set. As shown in Listing. [3.14] all the above-mentioned operations are performed in rapid succession, no integration of the equations is carried out so the procedure is quite fast. Naturally, this portion of the code is executed only if the class for the specific type is not empty, in case there are no particles of a certain type, this section of the code must not be executed, the first row is designed for this.

Listing 3.14: Portion of the definition of class `Trajectory` contained inside `gc2d_modules.py`.

```
if self.x.size:
    if case.Method.endswith('_fo'):
        self.vx, self.vy = sol_[2], sol_[3]
        self.x_gc, self.y_gc = case.fo2gc(t, *sol_)
        self.mu = case.compute_mu(t, *sol_)
    if case.check_energy:
        self.k = sol_[-1]
        h = case.compute_energy(t, *sol_)
        self.h = ((h.T - h[:, 0]) / h[:, 0]).T
    if case.darkmode:
        cs = ['k', 'w', 'c', 'm', 'r']
    else:
        cs = ['w', 'k', 'b', 'm', 'r']
    self.color = {type in ['trap', 'trapped']: cs[2], type in ['diff', 'untrapped']:
    ↪  cs[3], type == 'ball': cs[4]}.get(True, cs[1])
```

Mean square displacement computation

As anticipated, if the category of the simulation is `diffusion` the code is gonna compute some diffusion coefficients describing the behaviour of a particular category of particles. This is done only for "diffusive" and "ballistic" since the coefficients associated with "trapped" particles would result in being close to 0, these particles remain confined to a very small region of space. One useful quantity calculated in order to better understand the behaviour of particles is mean square displacement. It is indicated as r^2, (or *MSD*), and it measures the portion of space being reached by particles (or entities in general) as well as a group of them. The computation is performed through the comparison of the position of the considered entity at current time t, with respect to a reference position (previously occupied by the same).

r^2 for a single particle

In case the position of a particle is repeatedly registered after a fixed amount of time Δt, the computation of *MSD* for a specific particle is performed through an average over multiple time intervals, the expression is the following:

$$r^2(\tau\Delta t) = \frac{1}{T-\tau}\sum_{k=0}^{T-\tau}\left|\boldsymbol{r}_{k+\tau} - \boldsymbol{r}_k\right|^2, \quad \tau = 0, ..., T-1, . \tag{3.7}$$

It is necessary to introduce some definitions in order to clarify some terms of Eq. [3.7]:

- The total number of time intervals available at time t is T, this means that $t = T\Delta t$.
- The total $r^2(t)$ is always evaluated on $T-1$ time intervals, the last one is never considered.
- τ is the τ-th instant of time at which the position of the entity is registered.
- r_i indicates the position of the particle at the i-th instant of time.

r^2 for N particles

Assuming that multiple particles will be part of a specific class, but also that *MSD* is supposed to describe the behaviour of the whole class, it is necessary to compute a single value for each category of particles. This is simply done through a simple average over all the values of $r^2(t)$ coming from all particles being part of the considered system, the resulting formula is the following:

$$r^2(\tau\Delta t) = \frac{1}{N}\sum_{n=0}^{N-1}\frac{1}{(T-\tau)}\sum_{k=0}^{T-\tau}|\boldsymbol{r}_{k+\tau}^{(n)} - \boldsymbol{r}_{k}^{(n)}|^2, \quad \tau = 0, ..., T-1. \tag{3.8}$$

The resulting trend of *MSD* is quite oscillating for the first instants of time as well as for the last ones, since some fittings need to be performed on the resulting trend it is necessary to cut the tails of the latter in order to obtain high-quality fittings. As shown in Listing. [3.15] this is done considering only the central values of the vector `r2` resulting `r2_win`.

Listing 3.15: Portion of class `Trajectory` containing $r^2(t)$ computation.

```
if case.Method.startswith('diffusion') and type in ['diff', 'ball']:
nt = self.x[0, :].size
xd, yd = (self.x, self.y) if case.Method.endswith('_gc') else (self.x_gc,
↪  self.y_gc)
self.r2 = xp.zeros(nt)
for _ in range(nt):
    self.r2[_] = ((xd[:, _:] - xd[:, :-_ if _ else None])**2 + (yd[:, _:] - yd[:,
    ↪  :-_ if _ else None])**2).mean()
self.t_win, self.r2_win = self.t[nt//8:7*nt//8], self.r2[nt//8:7*nt//8]
```

Curves fitting

Once *MSD* trend is obtained for the specific category of particles, two fittings of the curve are performed. This is done to better characterise the behaviour of the present group of particles. It has been chosen to perform two fittings because of the usual trend resulting from ballistic particles. In addition to a standard linear plot, from which we obtain the usual diffusion coefficient, a power law is carried out, the generic expressions of the two curves are:

$$f_1(t) = D \cdot t. \tag{3.9}$$

$$f_2(t) = (a \cdot t)^b. \tag{3.10}$$

Both the regressions are carried out whether is the category of the particles under analysis, obviously, since the usual trend of $r^2(t)$ for *diffusive* is linear in time, it makes more sense to pay greater attention to the linear regression instead of the power law. In fact, the fitting from the power law can actually be misleading in case of *diffusive* particles, *a* value can be very low in case b is not close to 1. In this case one can't interpret a as a diffusion coefficient, it can't be defined, we recall the definition of *diffusion coefficient*:

$$\mathcal{D} = \lim_{t\to\infty} \frac{r^2(t)}{t}. \tag{3.11}$$

On the other hand, for ballistics the trend is quite different, when this type of behaviour is present (for certain combinations of parameters we don't have this type of particles) the trend is quadratic, and it is in this case that the power law fitting is better performing, the typical regression presents a value of b close to 2. The criteria for distinguishing the two types are not fully resolute, given that it is based on a fixed threshold and that `Tf` can be increased indefinitely, a lot of diffusive can be accounted for as ballistic, in this specific case one can check the value of b in order to better understand if there is a good percentage of hiding diffusive particles inside the other category. In general, we can classify the behaviour of the system of particles by looking only at b:

- $b \simeq 0 \Rightarrow$ the system doesn't diffuse.
- $0 < b < 1 \Rightarrow$ the system is sub-diffusive.
- $b > 1 \Rightarrow$ the system is super-diffusive.

The fittings are exploited through two different methods, in the case of linear regression it is used `linregress`, imported from `scipy.stats`. In the case of power law fitting it is used `curve_fit`, imported from `scipy.optimize`. The latter is very powerful, the user can specify the model function in order to use whatever function he wants, resulting in a fully customized fit of a specific set of data.

Coefficient of determination R^2

The coefficient of determination is widely used in literature to measure the quality of a curve fitting. It is obvious that for a linear regression of a periodic (high amplitude) oscillating function, the resulting fit is not well approximating the initial trend, the computation of this quantity is giving us a quantitative measure of the quality of the regression, this is very useful to screen in a fast way the results, we may choose to not account for results with a value of R^2 lower than a specific threshold. The best score for the latter is 1, however, there is not a fixed threshold for considering bad a fitting. This is due to the fact that fittings can be very different from one to the other, multiple variables are involved and the systems can be very different. What has been done is looking at multiple plots in order to look for not accurate fitting and their correspondent R^2 values, naturally this value can be different in the case of $f_1(t)$ and $f_2(t)$. It is possible to introduce the generic expression of this quantity in order to better understand what represents. First, we introduce some definitions, to better contextualize, there are some references to quantity for power law fitting:

- $\boldsymbol{y} = (y_0, ..., y_T)$: numerical data vector (`r2_win` in our case).
- $\hat{\boldsymbol{y}} = (y_0, ..., y_T)$: associated curve fitting data (`r2_fit` in our case).

- $\bar{\boldsymbol{y}} = \frac{1}{T+1}\sum_{j=0}^{T} y_j$: mean value of $\boldsymbol{y}$ (mean of `r2_win` over time in our case).

The expression of the coefficient of determination is the following:

$$R^2(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 1 - \frac{\sum_{j=0}^{T}(y_j - \hat{y}_j)^2}{\sum_{j=0}^{T}(y_j - \bar{\boldsymbol{y}})^2}. \tag{3.12}$$

The computation of this quantity, for both fittings, is left to built-in functions, `r2_score`, imported from `sklearn.metrics`, is used for the power law fitting, while for the linear regression, it is used the squared value of a specific attribute associated with an output of `lineregress`.

In Listing. [3.16] one can find all the implemented versions of fittings and R^2 computation.

Listing 3.16: Portion of class `Trajectory` containing fittings computation.

```python
from scipy.optimize import curve_fit
from scipy.stats import linregress
from sklearn.metrics import r2_score

res = linregress(self.t_win, self.r2_win)
self.diff_data = [res.slope, res.intercept, res.rvalue**2]
func_fit = lambda t, a, b: (a * t)**b
popt, pcov = curve_fit(func_fit, self.t_win, self.r2_win, bounds=((0, 0.25), (xp.inf,
↪  3)))
self.r2_fit = func_fit(self.t_win, *popt)
R2 = r2_score(self.r2_win, self.r2_fit)
self.interp_data = [*popt, R2]
```

3.4.3 Example of standard execution

Now that we know how the class works, it is possible to analyse the instructions the code has to follow, given that we defined only one equation to be integrated (depending on the mode the right governing equations will be integrated) and that all the computation of useful quantities for the trajectories are executed inside the specific class (on the basis of which parameters have been set for the simulation different instructions will be exploited) the resulting implementation is quite simple and concise.

Listing 3.17: Portion of `gc2d_modules.py` that actually executes the most important parts of the code.

```python
    if not case.TwoStepIntegration:
        sol = solve_ivp(case.eqn, (0, t_eval.max()), y0, t_eval=t_eval,
        ↪  max_step=case.TimeStep, atol=1, rtol=1)
        sol_ = xp.split(sol.y, case.dim)
        Trapped = Trajectory(case, t_eval, sol_, 'trap')
        Diffusive = Trajectory(case, t_eval, sol_, 'diff')
        Ballistic = Trajectory(case, t_eval, sol_, 'ball')
```

Once that all concerning data have been obtained, it is possible to store information from the three classes. The variable `data` is a data structure, once it will be saved inside the `.mat` file it will be easy to access all the information contained. For example, to access a specific class, using a `matlab` script, it will be sufficient to assign to a variable the value of `data{i-th class}`. They are stored in the order shown in Listing. [3.17] so that if the user wants to access class `Ballistic` has to index in position 3 of the `data` structure. Once that information are stored in a second variable, (but this can be done also without storing all quantities from a class in a variable), one can access specific values as attributes of the data. For instance, if one wants to access the trajectory of all particles from `Diffusive` class it will be sufficient to exploit the following lines:

Listing 3.18: Generic MATLAB script that can be used to access to `Diffusive` trajectories.

```
load(dir('*.mat').name);
Diffusive = data{2};

x_diff = Diffusive.x;
y_diff = Diffusive.y;
```

This can be done for all the quantities computed inside the class `Trajectory`, it is very efficient and quite intuitive to work with this type of data structure oriented to object-oriented programming. Naturally, the code is quite articulated and not so easy to read but once everything is at the right place and everything is perfectly understood every passage and instruction turns out to be very reasonable and with a logical sense other than being very synthetic.

3.4.4 Two-step integration

Two-step integration is a method that can be used in order to reduce the computational time of a single execution. The user can set this option through `gc2d_dict.py` as always. This functionality has been implemented because some particles, the trapped ones, follow the same close path despite the fact the number of cycles is very high. If after $1000 \div 1500$ cycles a particle is still considered trapped (while still following the same trajectory) then it is convenient to not continue to integrate its trajectory, this means reducing the number of equations integrated with an obvious reduction of the computational time. It is clear that if there aren't a lot of trapped particles the influence of this functionality on the computational time is very low. The parameter `Tmid` is equal to the number of cycles for which the trajectory of all `Ntraj` particles will be integrated. After this time cycle, the total number of trajectories integrated will be `Ntraj` − `xp.size(Trapped.x,0)` and it will continue without any stop until the end of the simulation. In order to do that, as shown in Listing. [3.19], it is necessary to call the integrator two times, the first from 0 to `t_eval[Tmid]` while the second time from `t_eval[Tmid]` to `t_eval[Tf]`. Between the two, the class `Trajectory` is called in order to establish which particles are considered trapped. After that, the vector `y0` is re-defined in order to furnish the integrator with the new initial conditions. At the end of the second integration, the essential vectors for untrapped particles are concatenated with the ones coming from the first integration. At the end of the procedure class `Trajectory` is called on `Untrapped` in order to distinguish diffusive particles from ballistic ones.

Listing 3.19: Portion of `gc2d_modules.py` containing the two-step integration procedure implemented in the code.

```
if not case.TwoStepIntegration:
    (...)
else:
    sol = solve_ivp(case.eqn, (0, t_eval[case.Tmid]), y0,
    ↪   t_eval=t_eval[:case.Tmid+1], max_step=case.TimeStep, atol=1, rtol=1)
    sol_ = xp.split(sol.y, case.dim)
    Trapped = Trajectory(case, t_eval[:case.Tmid+1], sol_, 'trapped')
    Untrapped = Trajectory(case, t_eval[:case.Tmid+1], sol_, 'untrapped')

    ↪   print("\033[90m        Continuing with the integration of {} untrapped particles... \033[00m".form
    ↪   .x[:,
    ↪   0].size))
    y0 = xp.concatenate((Untrapped.x[:, -1], Untrapped.y[:, -1]), axis=None)
    if case.Method.endswith('_fo'):
        y0 = xp.concatenate((y0, Untrapped.vx[:, -1], Untrapped.vy[:, -1]),
        ↪   axis=None)
    if case.check_energy:
        y0 = xp.concatenate((y0, Untrapped.k[:, -1]), axis=None)
    sol = solve_ivp(case.eqn, (t_eval[case.Tmid], t_eval.max()), y0,
    ↪   t_eval=t_eval[case.Tmid:], max_step=case.TimeStep, atol=1, rtol=1)
    sol_ = xp.split(sol.y, case.dim)
    Untrapped.x = xp.concatenate((Untrapped.x, sol_[0][:, 1:]), axis=1)
    Untrapped.y = xp.concatenate((Untrapped.y, sol_[1][:, 1:]), axis=1)
    vec_un = (Untrapped.x, Untrapped.y)
    if case.Method.endswith('_fo'):
        Untrapped.vx = xp.concatenate((Untrapped.vx, sol_[2][:, 1:]), axis=1)
        Untrapped.vy = xp.concatenate((Untrapped.vy, sol_[3][:, 1:]), axis=1)
        vec_un += (Untrapped.vx, Untrapped.vy)
    if case.check_energy:
        Untrapped.k = xp.concatenate((Untrapped.k, sol_[-1][:, 1:]), axis=1)
        vec_un += (Untrapped.k,)
    Diffusive = Trajectory(case, t_eval, vec_un, 'diff')
    Ballistic = Trajectory(case, t_eval, vec_un, 'ball')
data = [Trapped, Diffusive, Ballistic]
info = 'Trapped / Diffusive / Ballistic'
```

3.4.5 Standard and modular Poincaré sections

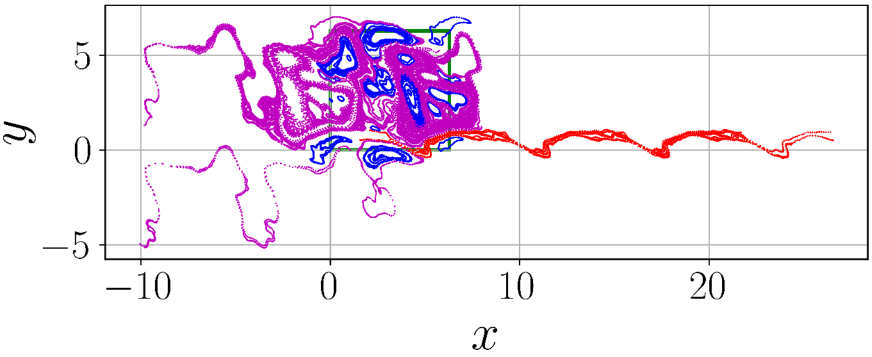
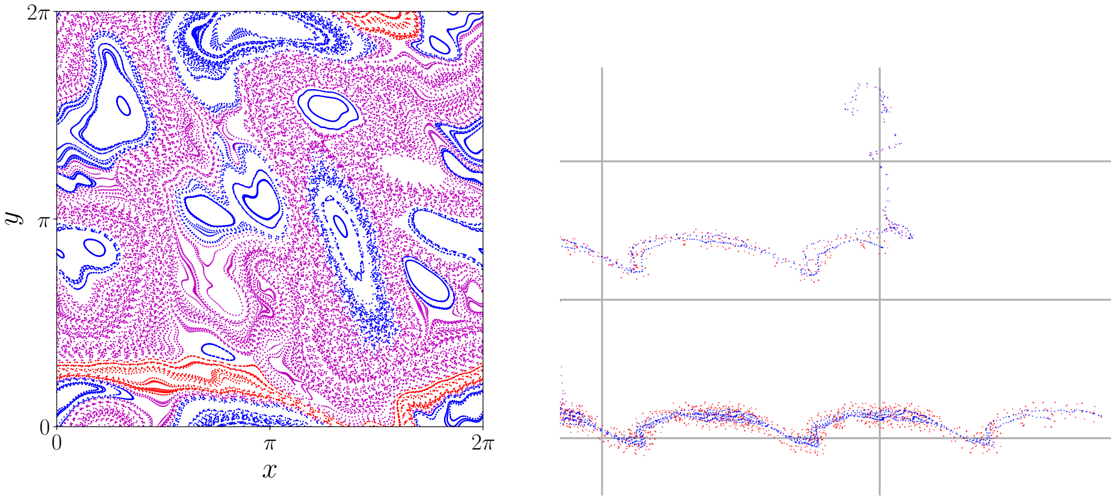


Figure 3.4: Standard Poincaré section from Guiding Center mode.

If `Method` starts with ’poincare’ and `PlotResults` = `True` then Poincaré section of correspondent trajectories are plotted. Particles are moving in the transverse plane under the influence of the turbulent potential, one has to remember that they are also moving along $\boldsymbol{z}$ direction with a constant velocity that is equal to $v_{||}$. Since we are only interested in the movement in the transverse plane, it is possible to store the data at the end of each time cycle (corresponding to points with a different value of z in the $6D$ phase space) but plotting all of them in the same plane, this can be done for each trajectory. The resulting plot is called Poincaré section (or stroboscopic plot) and it is widely used in literature to study the behaviour of particles in simple geometries like ours. The modular version of stroboscopic plots is made by applying the modulo operator (referred to 2π) to the trajectories, the resulting sections range only over a limited region of space, the fundamental square indeed. Through this kind of operation particles going out of the square from the right side will re-enter from the left side and so on. This tool is very useful in order to study trajectories that seem to follow the exact repetitive same path along the cycles moving away from the fundamental square, the modular plot actually superimposes all the paths being followed during each step from side to side. In the plots, different kinds of particles are coloured differently, as we remember the class `Trajectory` set the colour as an attribute `.color` for each typology. The plots are generated at the end of the execution but the figures can be stored (with a format customized by the user through `fig_extension`) when `SaveData` = `True`. In case `Method` ends with ’_fo’ particles’ trajectory and correspondent guiding centre trajectories are both plotted. Looking at Fig. [3.5b], the red points are points belonging to the full orbit trajectories while the blue ones are the correspondent guiding centre positions.

3.4.6 Curves fitting plots

If `Method` starts with ’diffusion’ and `PlotResults` = `True` then the plot of various fitting is being exploited. For both `Diffusive` and `Ballistic` particles, the code plots the trends of $r^2(t)$, $r^2_{win}(t)$ and $r^2_{fit}(t)$. The latter comes from the `func_fit` in case of `Ballistic`



(a) Modular Poincaré section from guiding center mode.

(b) Zoom of Poincaré section from full orbit mode.

Figure 3.5: Modular Poincaré section and zoom of a Poincaré section coming from full orbit mode.

while from the `linregress` in case of `Diffusive` class. In the plot the dotted lines are the two $r^2(t)$, the solid lines are $r^2_{win}(t)$ trend while the dashed line represents $r^2_{fit}(t)$ (if the fitting is well done the last two are not distinguishable). As we anticipated in Sec. 3.4.2 , for `Diffusive` linear fitting is considered while the power law is considered for `Ballistic`. While looking at Fig. it is clear that this is the most reasonable choice, the trend of these curves is quite different. Obviously, there are some conditions (combination of parameters `A`, `rho` and `eta`) that don't produce a plot like the present one, in that case, linear regression may be helpful also when `Ballistic` are under analysis, their behaviour may be quite similar to `Diffusive`, that's why both fittings are performed for both types of particles. The present one is the graph in output from the code, naturally one can easily recover data from `.mat` file and re-plot other than customize the entire plot, this is a graph produced simply to have a first quick anticipation of the results.

3.4.7 Generating animations of the potentials

If `Method` is equal to 'diffusion' and `PlotResults` = `True`, the code produces a plot of the concerning potentials. Both the standard potential and the guiding centre potential but in addition to that, the first and second order of the latter are separately plotted in order to give more information to the user. All the plots are referred to the same instant of time, $t = 0$ to be precise. In case `SaveData` = `True` data with which these figures have been produced are stored inside the `.mat` file, generated with this specific category of simulation. As it is possible to see, the potentials are defined right inside the fundamental square, the domain is exactly the same in which modular Poincaré sections

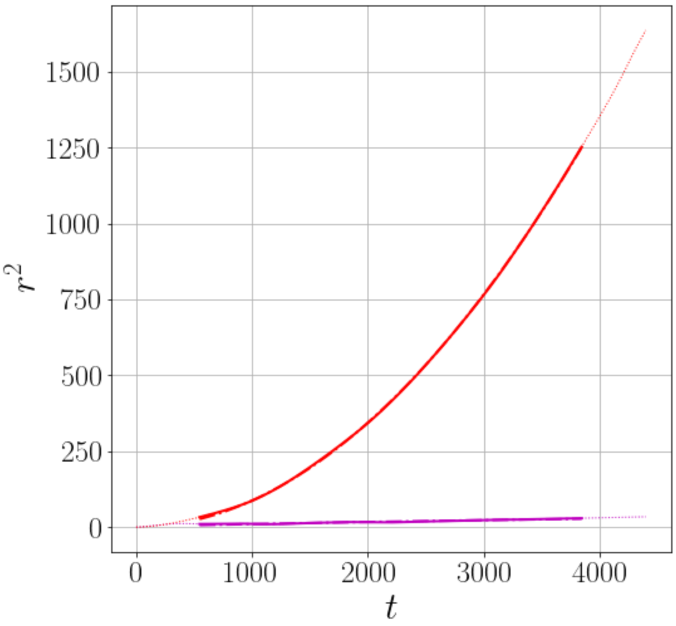


Figure 3.6: Curves fitting plot, guiding center mode.

are plotted. As we know the potential is periodic in time as well, since it can be interesting to have a little animation of how the various potentials behave through the temporal period, it has been inserted a tool to produce `.gif` animations inside the code. Given that the temporal dependence of the potentials is embodied inside the exponentials, it is just necessary to generate a vector time with a certain number of values going from 0 to 2π (each period does last 2π) and multiplying the respective complex potential with the respective exponential (we know that, for instance, second-order guiding centre potential has a different time dependence), after having taken the imaginary (or the real part, it depends on the specific case) we have the trend over time for each point (of the potential grid). The code stores all the data and produces a `.gif` file showing the evolution of all the potentials overtime during an entire period. The figures produced by the code can be really useful whenever the user wants to change the combination of parameters, the effect of varying `A` for example, can be clearly seen in the resulting figures as well as the change in the sign of `eta` or the increase in the value of the latter. Another case where this feature is very useful is the changing in the random phases seed, one can actually see if a specific seed produced the result sought by the user.

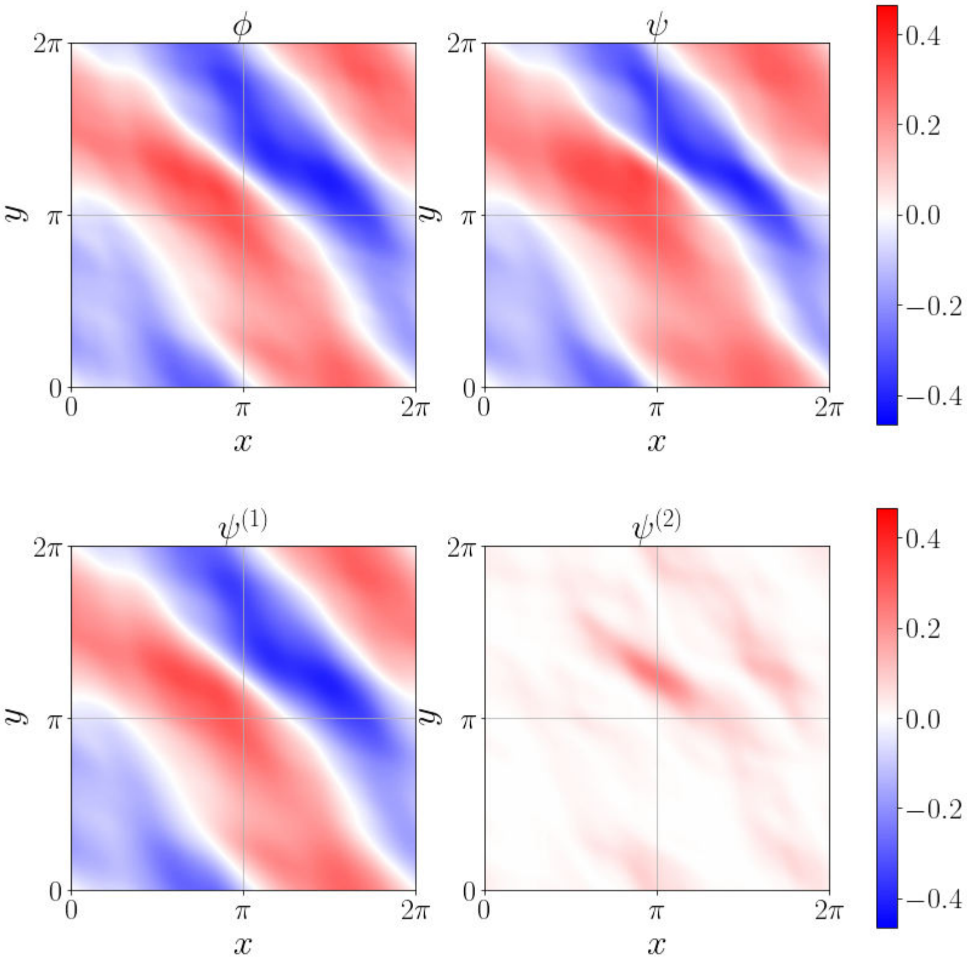


Figure 3.7: Potentials plots at fixed time $t = 0$. Standard potential at the top-left figure, full guiding centre at the top-right, first-order g.c. at the bottom-left and second-order g.c. at the bottom-right figure.

Chapter 4

Numerical Analysis

4.1 Full-orbit to Guiding-Center

One of the main tasks after the model was completely derived and fully implemented has been the check on the operation of `fo2gc` function. The proper functioning of the latter also gave a lot of information about the good functioning of full-orbit mode, if the dynamics of the guiding centres coming from the full-orbit mode were the correct one then also the dynamics of the particle and the implemented model were correct. The main aim of the work was to make comparable the results in output from the two different modalities. As anticipated in the Subsec. [3.3.5], the function works simply through the use of the equations for the change of coordinates one has to use when he passes from the particle framework to the guiding centre one. First, some Poincaré sections have been analysed in order to see if they could have been considered comparable. First, we analyse the resulting dynamics of single particles in order to understand if the method works correctly. The dynamics of different types of particles should be analysed, first we perform the analysis on a super-diffusive one, the typical rotation of the particle around the trajectory of the guiding center should be highlighted. The trajectories have been extracted from a simulation performed with $\mathtt{A} = 0.7$,

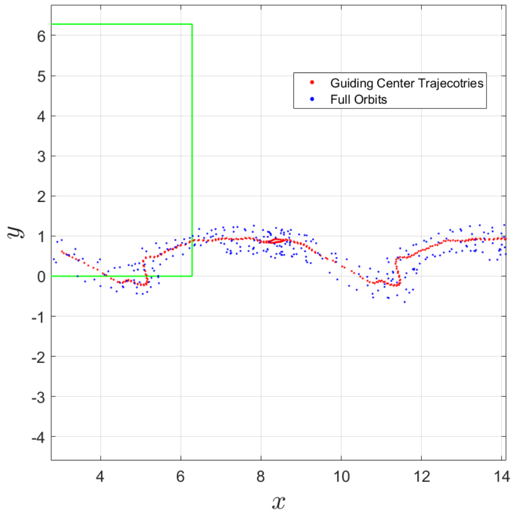


Figure 4.1: Full-orbit trajectories comparison with guiding-centre corresponding ones through the use of `fo2gc` method for two super-diffusive particles.

`eta` = 0.14 and `rho` = 0.3. This set of parameters provides a very high number of super-diffusive particles so we extracted two of them. As shown in Fig. [4.1] the method works perfectly, the expected dynamics is very well reproduced, particles are rotating around the guiding centres and this behaviour is maintained until the end, here only a part of the entire trajectory is shown to better highlight the dynamics.

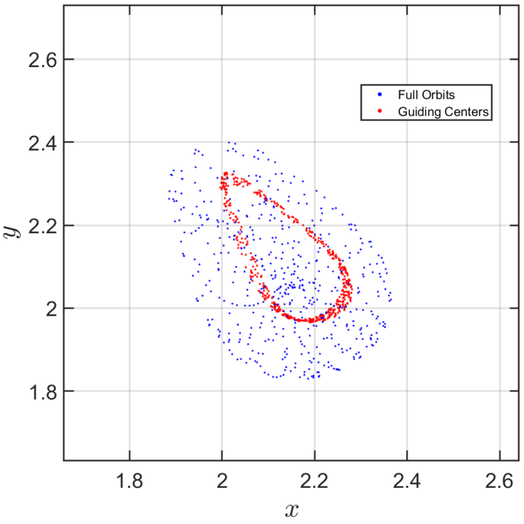


Figure 4.2: Full-orbit trajectories comparison with guiding-centre corresponding ones through the use of `fo2gc` method for a trapped particle.

In order to better understand if the method works correctly in all circumstances it has been decided to investigate also the behaviour of trapped particles, some trajectories have been extracted, from a simulation performed with `A` = 0.7, `eta` = 0.10 and `rho` = 0.10. Also in this case, the code seems to work correctly, as shown in Fig. [4.2] the dynamics in evidence reflects the real dynamics of a trapped particle, the particle rotates correctly around the guiding centre. It is possible to conclude that `fo2g` function works correctly and that it is well implemented in the code, following analysis coming from full-orbit mode will show always guiding centres, this is done to better compare the output results from the two modes, now we are also sure that this particular passage is reliable.

4.2 Time-step analysis

In the following pages the influence of `TimeStep` parameter, fixed in file `gc2d_dict.py` by the user is under analysis. This parameter is very important for the quality of the results, if the time discretization is too large then the calculated trajectory is far from being similar to the real one. That's why, like in every code in which the evolution in time of quantities is calculated, this choice is fundamental. Firstly we have to remember that choosing the GC mode or full orbit mode changes the dynamics of the system. While for the Guiding Center approach the trajectory calculated and the real trajectory of the latter is quite regular and smooth if analysed from an external observer the trajectory of the corresponding particle is very chaotic. The particle rotates around the Guiding Center position at a very high speed, the dynamics is faster and as we can imagine this affects the choice of temporal discretization. In addition to the reduction of the order of the system, this is another great strength of the gyro-kinetic approach to this kind of system. The integration of one less variable reduces the computational time but being able to integrate those remaining variables which a time step of a different order of magnitude compared to the previous grants a huge reduction in the time required for the calculations. This concept is at the base of the gyro-kinetic approach and has to be clear in order to better understand and reach the final goal. It is possible to remember the equations we are integrating within *GC_mode* and *Ions_mode* respectively: 3.152 Tesi and:

$$\frac{\partial \hat{V}_x}{\partial \hat{t}} = -sign(\hat{\eta})\frac{1}{\hat{\rho}}\frac{\partial \hat{\psi}}{\partial \hat{x}} + \frac{1}{2|\hat{\eta}|}\hat{V}_y. \tag{4.1}$$

$$\frac{\partial \hat{V}_y}{\partial \hat{t}} = -sign(\hat{\eta})\frac{1}{\hat{\rho}}\frac{\partial \hat{\psi}}{\partial \hat{y}} - \frac{1}{2|\hat{\eta}|}\hat{V}_x. \tag{4.2}$$

In order to choose the value of `TimeStep` for both modalities it is necessary to evaluate the quality of the results while varying that specific parameter. One can be misled into thinking he can directly compare trajectories coming from different input parameters. In this case, only one parameter is varying, nevertheless, it is not possible to confront the trajectory of the same particle despite the fact that it starts from the same position and although it is driven by the same potential. That's because the system is chaotic, the potential is turbulent and also an insignificant difference in the position (at a certain instant of time) can lead to a completely different trajectory of the same particle. Having said that, it is not possible to compare through the definition of a simple error the solution of a certain particle taking as a reference the solution obtained with the smallest time interval. It makes more sense to analyse the trend of the Hamiltonian observable for some particles and evaluate how these trends vary consequently to a variation in the temporal discretization. As it is well known the Hamiltonians of the two modes have a different expressions, In the gyro-kinetic approach, the first two contributions are constant, given that $\boldsymbol{B} = B\hat{\mathbf{z}}$ and that is valid $B = Cst$, likewise, $v_{\parallel}$ is constant as well. We are interested in the evolution of H and H', these should oscillate if compared to the initial value but should not move too much. The energy enters the system only through the potential as we have seen in Eq. [2.36] but the latter is periodical in time and space and it is not supposed to increase the energy of a particle indefinitely. In conclusion, the final result of the trends should be a continuous oscillation close to the initial value for all particles. While looking at Eq. [2.36] and [2.296] we notice that after the integration of the equations we defined all data necessary for the computation of the variable terms to H and H' are available. From the integration of energy k is available for both modes, the values of ϕ, $\langle\phi\rangle_\theta$ and $\langle\tilde{\phi}^2\rangle_\theta$ are available from the computation of the potential at each instant of time and in full orbit mode we know the values of v_x and v_y for each particle at the end of each time cycle despite the fact that we are calculating the scaled version. It is possible to introduce an equation that computes the values of H and H' at the end of each time cycle taking advantage of the availability of data already available, naturally, this happens if the parameter `check_energy` = `True` (in `gc2d_dict.py`), otherwise the computation of k and H as well is not expected. In order to implement the equations it is necessary the non-dimensionalisation of both H and H'.

4.2.1 Computing of absolute relative error

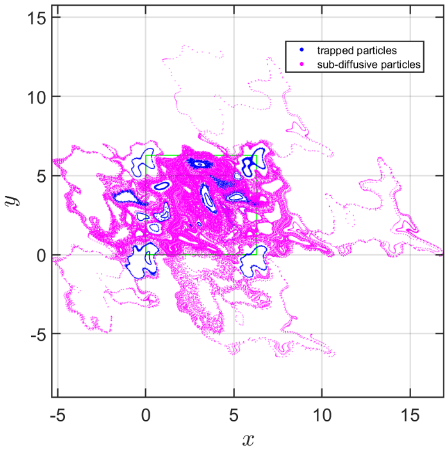


Figure 4.3: Poincaré section from GC mode, diffusive regime.

In order to better look for the desired trend we can choose to not plot H and H' directly. Given that we are interested in the deviation from the initial value, we are always interested in $H - H_0$ with H_0 defined as the initial value of H. This means we are not interested in computing fixed terms. It is possible also to normalize the trend on the initial value, plotting directly a relative error in absolute value. We won't be able to see the oscillations from the positive to the negative value of H but the trend can be plotted in a semi-logarithmic graph, this will lead to a more accurate analysis. Inside the code, some lines are responsible for saving the desired variable in the *mat* file. It is stored directly:

$$rel\_error = \frac{|\hat{H} - \hat{H}_0|}{|\hat{H}_0|}. \tag{4.3}$$

This is very practical also for Guiding Center mode, it is necessary to remember that the equation for $\hat{H}$ is re-scaled. Given that rel_error is normalized on $2\hat{\eta}\hat{H}_0$ then we don't care about the presence of $2\hat{\eta}$. This trend is required to check the solidity of the model and of the equation implemented, anomalous results even with a very little time step would mean that there is an error in the model. Once we check that the model is correct it will be necessary to choose a value for `TimeStep` in both modes. Given that in previous simulations it has been highlighted the presence of two main regimes, diffusive and super-diffusive it is necessary to compare the plot of energy done with the same time-step but different regimes in order to avoid the choosing of a certain time discretization that gives you acceptable results in one regime while not acceptable in the other. It is necessary to assure that the chosen value works fine whether is the regime simulated. The two designed regimes are just a couple of combinations of parameters, naturally, they are not representative of all the possible conditions. The following combinations have been chosen to determine a diffusive and super-diffusive regime respectively:

$$\mathtt{A} = 0.70\ ;\ \mathtt{rho} = 0.115\ ;\ \mathtt{eta} = -0.100. \tag{4.4}$$

$$\mathtt{A} = 0.70\ ;\ \mathtt{rho} = 0.250\ ;\ \mathtt{eta} = 0.05. \tag{4.5}$$

The following pictures show the resulting Poincaré sections calculated from GC mode for both regimes, it has been checked that the Poincaré sections coming from GC and Full Orbit mode are in good accordance in both cases. The procedure consists in simulate this couple combinations of parameters with both modes using different values of the time step. In order to have a better comparison between full orbit and guiding centre mode the series of values of `TimeStep` is fixed for all the simulations. It will be also analysed

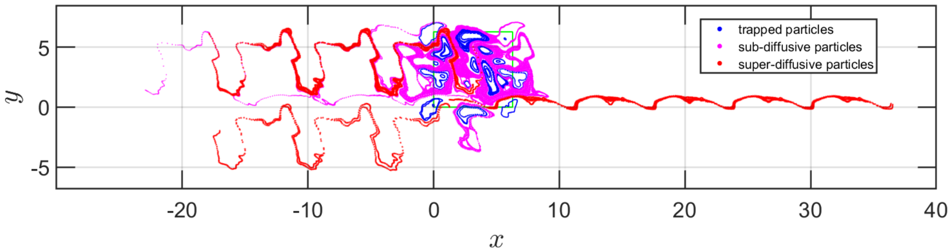


Figure 4.4: Poincaré section from GC mode, super-diffusive regime

the computational time required for each simulation, this aspect is very important given that the choice of the value has to take into account also the time that will be required for the calculations, it has to be a balance between accuracy (and quality) of results and computational time. The other important parameters fixed for all the simulations are:

$$\begin{gathered}\texttt{Ntraj} = 100 \ ; \ \texttt{Tf = 700} \ ; \ \texttt{FLR} = \texttt{(True, True)} \ ; \ \texttt{Parallelization} = \texttt{(False, 1)} \ ; \\ \texttt{TwoStepIntegration} = \texttt{False} \ ; \ \texttt{check\_energy} = \texttt{True} \ ; \ \texttt{modulo} = \texttt{False}.\end{gathered} \tag{4.6}$$

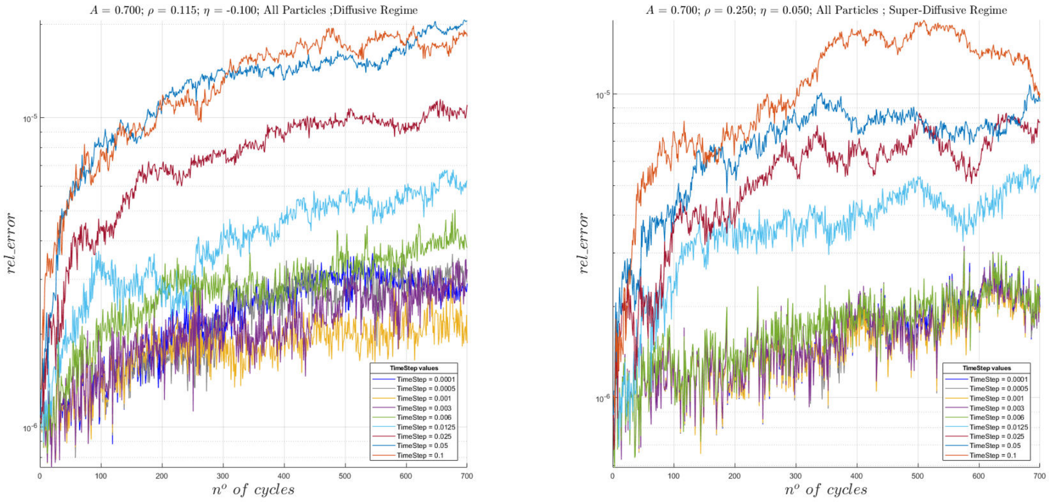
It has been chosen to use a small number of particles given that integrating also the equation of energy is more expensive computationally speaking and that we don't need a large number of them to check what is required. The Tf has to be not that little in order to allow the particles to stabilize and settle to their final error in oscillation. The list of values of `TimeStep` has been chosen as:

$$\texttt{TimeStep} = \Big[0.1, 0.05, 0.025, 0.0125, 0.006, 0.003, 0.001, 0.0005, 0.0001\Big]. \tag{4.7}$$

As anticipated each regime will be simulated with both modalities and all the possible values of `TimeStep` included in the list. The analysis of results has been done in Matlab, I have coded a script able of loading all data from the same type of simulation (same physical parameters) but with different temporal discretization. The plot of the various trend for rel_error has been done also for trapped and untrapped particles separately other than for all particles. All this is to analyse if trapped and untrapped particles have different behaviours. From the code, it is obtained the value of rel_error for each particle at the end of each time cycle. It has been decided to plot the evolution in time of the average error (average made on all the particles of the considered group) and compare the different trends generated by different TS. It has also been decided to plot the variance of the error considering the group of particles, it is possible to have an indication of how many and how many particles have deviated from the average value. It is important to remember that it is possible to have a good average value of the error while having some particle diverging in energy, choosing a TS that determines the deviation (in energy) of a lot of particles is not optimal, it means that the discretization is not sufficiently dense. The variance gives us an idea of what is happening but it is possible to do more. It is possible to fix a value of error at which a particle is considered diverging. Once this has been done it is possible to count how many particles are reaching that threshold and compare that number whit the one coming from different simulations. It is also possible to fix the second threshold on the value of the maximum average error for that specific `TimeStep`, it is like fixing an error bar and counting how many particles are going out of the desired band. After this little introduction to the various types of plots used to complete the analysis, it is time to show the various plot. Results from the same mode are first shown together in order to understand if there is any difference between the two regimes and then results from different modes are compared to check that the initial purposes and the initial forecasts were correct.

4.2.2 Time-step analysis for guiding centres dynamics

Starting from GC mode it is possible to plot the trends of *rel_error* from the average made on all the particles simulated for different `TimeStep` values and both regimes. From Fig. [4.5] it is clear that there is not a major difference between the error generated from the same `TimeStep` value in the two regimes. This means that it is not required to vary the temporal discretization when we want to simulate a regime rather than another. This is the major result of this comparison. In addition to that it is clear that the error is very low in the majority of cases. The average error is under 10^{-5} if `TimeStep` ≤ 0.0125. As it can be clearly seen in Fig. [4.5b], for $TS \leq 0.006$ a sort of convergence is reached, the utilisation of too low values is not worth it given that as we will see the time required for the calculation is inversely proportional to the above-mentioned value. These were the plot of the average error, as we anticipated this doesn't tell us how many particles are diverging in energy. In order to do that we plot the number of particles from each

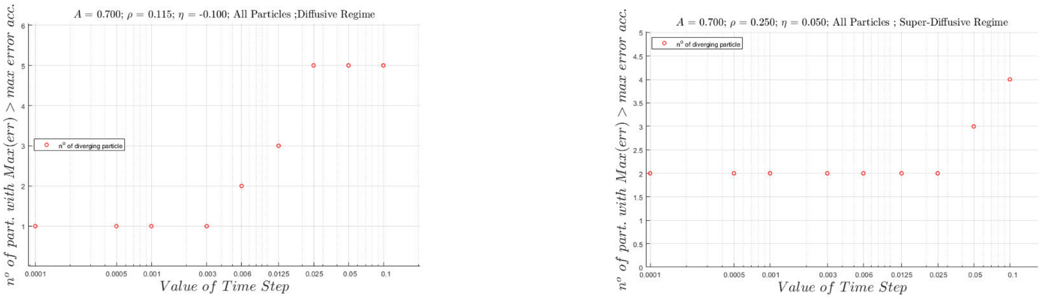


(a) *rel_error* trends from different `TimeStep` value, GC mode in diffusive regime

(b) *rel_error* trends from different `TimeStep` value, GC mode in super-diffusive regime

Figure 4.5: *rel_error* comparison between diffusive and super-diffusive regimes in GC mode

TS value that has a maximum *rel_error* above 10^{-4}, the latter is considered as a good threshold value in this case. The above-mentioned plot has been done for both regimes in order to look for substantial differences. It is shown simply the number of those particles, one has to remember that in this case, the total number of particles is 100. From Fig. [4.6] it is clear that the results are in good accordance between the two modes, the number of particles considered diverging is very low, $\leq 5\%$ for all the TS used. It hasn't reached the non-divergence of all particles but it can be said that the behaviour of the system is good. In a diffusive regime, after the `TimeStep` goes above 0.003 there is a little transition, the number slightly increases but the value reached is considered small and of the same order of magnitude as the precedent. To be cautious the choice of `TimeStep` will account for that, it is necessary to avoid any anomalous behaviour. No remarkable differences between trapped and untrapped particles have been noted in various plots comparison for GC mode, it has been noted that trapped particles have a slightly higher error if we compare results from the same TS but this problem will be faced later. Given all that has been said in this subsection the proposed value of `TimeStep` for whatever combination of parameters chosen by the user for a generic simulation is `TimeStep` $= 0.005$. The latter is giving us very good values of *rel_error* for both regimes, it seems it can reach the convergent value of *rel_error* aligning its result with the one coming from much more expensive simulations. It would be not necessary to choose a smaller value, at the end of the section some useful considerations on the residual error will be carried out.



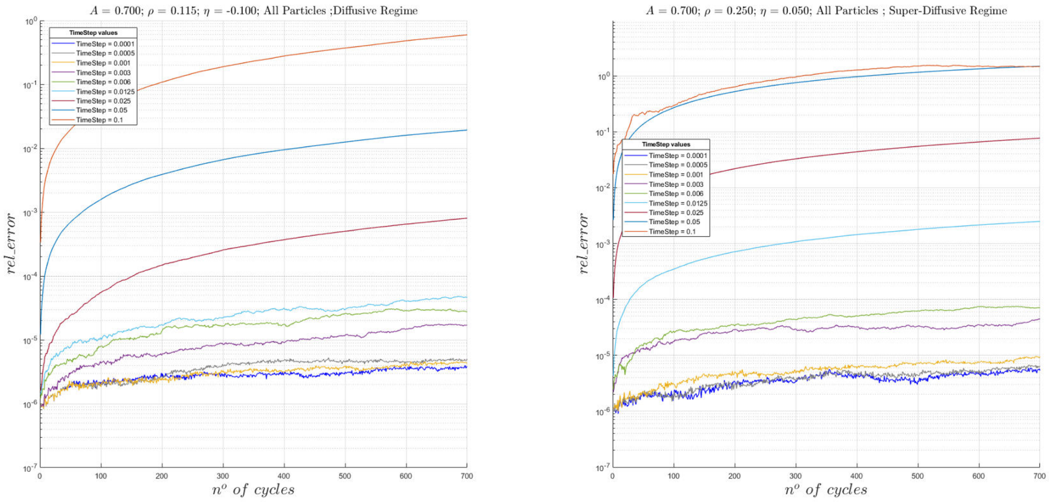
(a) Number of particles out of the error threshold for different `TimeStep` value, GC mode in diffusive regime. Total number of particle is 100

(b) Number of particles out of the error threshold for different `TimeStep` value, GC mode in super-diffusive regime. The total number of particles is 100

Figure 4.6: Number of particles out of the error threshold comparison between diffusive and super-diffusive regimes in GC mode.

4.2.3 Time step analysis for full orbits dynamics

The following plots show the correspondent trends of Fig. [4.5], they have been done with the same script in MATLAB. The trends from GC are completely different from the one of Fig. [4.7]. In this case in fact the errors are significantly higher if we compare them with the correspondent trends made with the same `TimeStep` it cannot be denied that there is a difference of 2 or 3 orders of magnitude in certain cases. The resulting errors

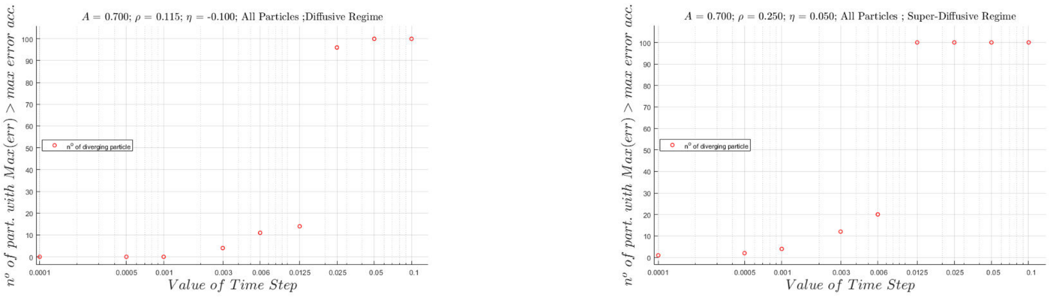


(a) rel_error trends from different `TimeStep` value, Full Orbit mode in diffusive regime

(b) rel_error trends from different `TimeStep` value, Full Orbit mode in super-diffusive regime

Figure 4.7: rel_error comparison between diffusive and super-diffusive regimes in Full Orbit mode

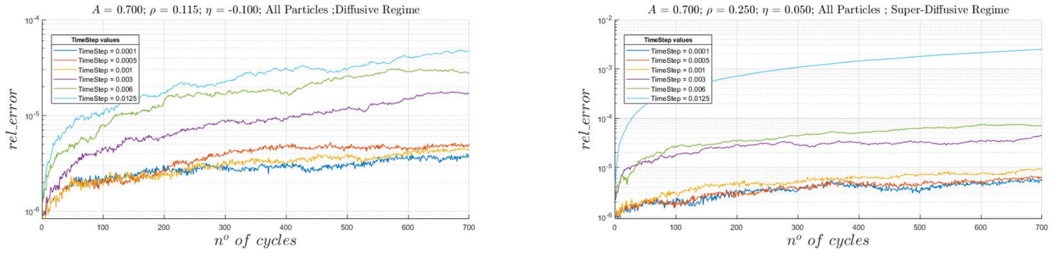
from a lot of the chosen values of TS are not acceptable. In order to reach the same error order of GC mode it is required to use a `TimeStep` value of a smaller order of magnitude, this was expected from the early analysis of the system we have done at the beginning of this section. It is very important that it is verified otherwise it would have meant that something was wrong with the model, the trajectories that the two modes are studying are completely different in the dynamics at it is correct that they require a different temporal discretization in order to generate results of the same quality. It is possible to say that the chosen value of `TimeStep` needs to stay under 0.006 in order to obtain some decent results, we will see how this threshold will be further reduced. Looking at Fig. [4.7], it is possible to say that also in this case a sort of convergence is reached. In fact from Fig. [4.7b] it's obvious that there is no point in going under `TimeStep` = 0.001. What has to be stressed while using Full Orbit mode is that not all the correct values of TS for a regime are suitable for another set of physical parameters given that the dynamics of the system can be completely different. The proposed approach won't be varying the temporal discretization in dependence of the above-mentioned parameters but on the contrary in choosing a TS giving us good quality results whichever are the conditions taken into account. In this case, the most unfavourable condition is the super-diffusive regime, it is necessary to choose temporal discretization taking as a reference the results from the super-diffusive regime in order to assure that we don't have any problem in any conditions. From Fig. 4.7 we should choose a value of TS that assure us of having reached the convergence in error, like we did with GC mode, then at least 0.001. It is required to analyse if this value matches the value suggested from the analysis on diverging particles as well. As it can be clearly seen from Fig. 4.8 it is necessary to use a low value of TS, also from these plots it can be seen that the super-diffusive regime is the most penalized one. In addition to that, with respect to GC mode, the percentage of diverging particles for a correspondent average *rel_error* is higher. The average relative errors are higher in general, particles are oscillating more and in order to reach a value of diverging particles $\leq 10\%$ it is required to use a value of $TS \leq 0.001$. Can be noticed a strong transition in both Fig. 4.8a and Fig. 4.8b while increasing the value of TS, it is recommended to work out of this region, the value previously proposed is borderline. Also in this comparison, it is possible to see the difference between the two regimes, given that for 0.0125 in the first case, we have a limited amount of diverging particles while not in the second case. As anticipated it is possible to notice that for $TS = 0.0125$, in the diffusive regime we reach an acceptable error (under 10^{-4} at least) while in the super-diffusive regime, it goes over 10^{-3}. This last consideration is clearly noticeable in Fig. 4.9, it is clear evidence that the two regimes under analysis require two different temporal discretizations in order to obtain comparable trends of *rel_error*. From the comparison of Fig. 4.9 it is also noticeable that in general, for almost every value of *TS* the *rel_error* is greater. One might wonder if



(a) Number of particles out of the error threshold for different `TimeStep` value, Full Orbit mode in diffusive regime.

(b) Number of particles out of the error threshold for different `TimeStep` value, Full Orbit mode in diffusive regime.

Figure 4.8: *rel_error* comparison between diffusive and super-diffusive regimes in Full Orbit mode



(a) Selected *rel_error* trends from different `TimeStep` value, Full Orbit mode in diffusive regime.

(b) Selected *rel_error* trends from different `TimeStep` value, Full Orbit mode in super-diffusive regime.

Figure 4.9: Selected *rel_error* trends comparison between diffusive and super-diffusive regimes in Full Orbit mode.

the additional contribution to the error is given from particles characterised by a ballistic behaviour. In order to investigate this last point, it has been set a threshold value for the distance the particle must travel for being considered ballistic. Naturally, the latter needs to vary with the number of cycles that it is necessary to simulate given that otherwise some diffusive may be wrongly considered ballistic. To better clarify and get a better picture of what we are considering it is possible to highlight those interesting particles in a *Poincaré* section. This test case has been simulated for $Tf = 700$ and 50 has been chosen as the threshold value. From Fig. [4.10] it is clear that only a small number of particles

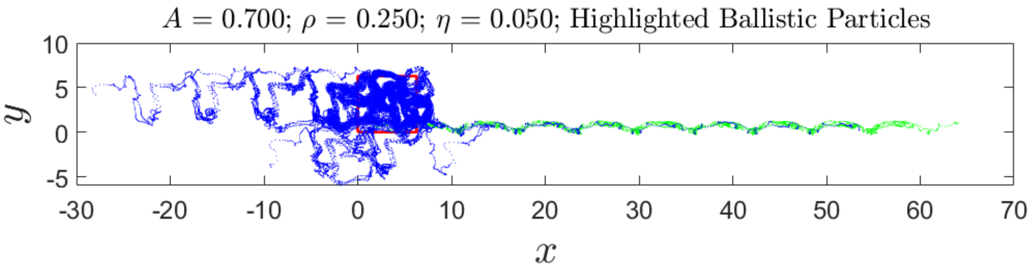
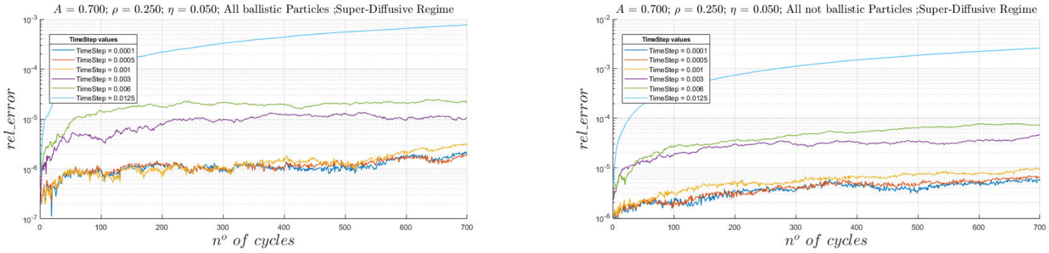


Figure 4.10: *Poincaré* section with highlighted ballistic particles from Full Orbit mode, super-diffusive regime

have the above-mentioned behaviour. What has been done is trying to plot the trends of *rel_error* for the various *TS* values for only ballistic or non-ballistic particles. This is meant to emphasize if the major contribution to the error is given by ballistic particles that are present in greater numbers when a super-diffusive regime is simulated. From



(a) Selected *rel_error* trends of only ballistic particles from different `TimeStep` value, Full Orbit mode in super-diffusive regime.

(b) Selected *rel_error* trends of only non-ballistic particles from different `TimeStep` value, Full Orbit mode in super-diffusive regime.

Figure 4.11: Selected *rel_error* trends comparison between ballistic and non-ballistic particles, super-diffusive regime in Full Orbit mode.

Fig. [4.11] it is noticeable that on contrary to what one would have thought the ballistic particles are characterised by a smaller error than the non-ballistic ones, this could be counter-intuitive but diffusive particles are associated with chaotic transport, whereas the ballistic ones are not, this cause a greater oscillation in the error of the diffusive ones. One further plot has been introduced in order to highlight this concept, from Fig. [4.12] it is clear the above-mentioned difference, it is really interesting to have a confirmation

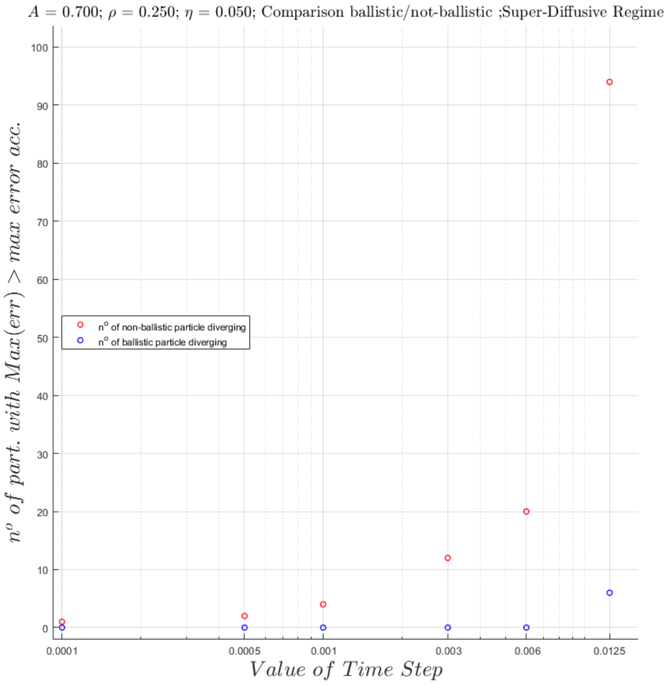


Figure 4.12: n^o of ballistic and non-ballistic diverging particles from different `TimeStep` values

of the fact that those particles are not oscillating at all if the right `TimeStep` value has been chosen. Given that the value of 0.001 is too close to the transition region it has been chosen 0.0005 as the final value of `TimeStep`.

From the previous analysis, the following values are the chosen ones

$$\texttt{TimeStep}_{\ gc\_mode} = 0.005\ ;\ \texttt{TimeStep}_{\ ions\_mode} = 0.0005\ ;$$

4.2.4 Spatial siscretization influence on energy error

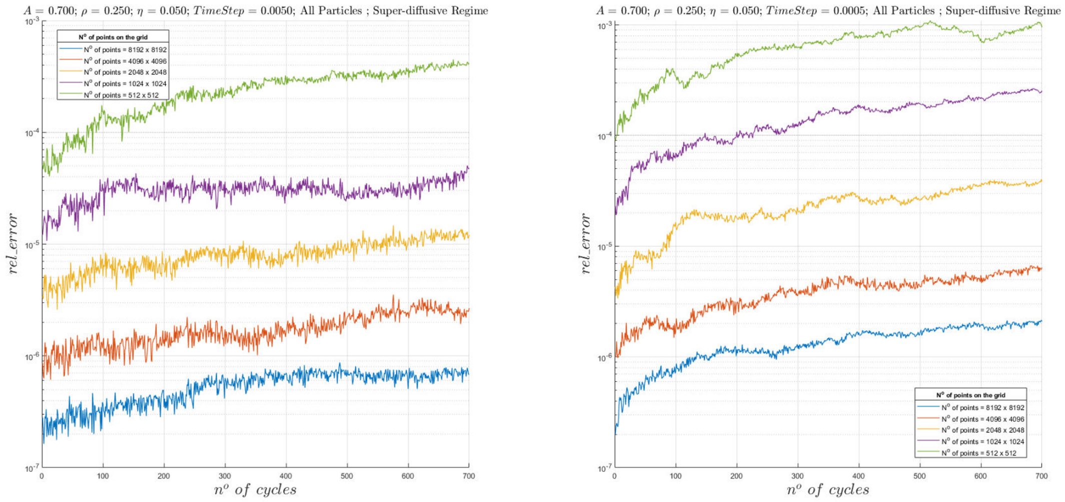
It has been said that those value guarantee acceptable average values of error and a very low number of diverging particles. It is possible to wonder if there are any other parameters that influence the quality of the solution and in particular the integration of energy. As we remember we are comparing results from energy in order to assure that also our resulting trajectories will be of good quality, in other words, more reliable data. It is a good practice to fully understand what influences the error in the solution, this error in fact can come from numerical schemes and operations done in the code that may not directly come from the mathematical model utilised. In our specific case while running the code there are a lot of specific operated functions from python libraries, given that those functions are fully optimized by python developers the complete operation has not been investigated, it would have taken a lot of time. In order to understand if some of those functions are generating numerical errors some tests have been done. One of the most investigated operations has been the interpolation function, this function is contained inside the equations of motion of the guiding centre and ions as well but also in the equation of energy. The interpolation function is called when it is required to know the value of the potential (or of its derivatives) on the particle position. The potential is known only on the points of the grid that composes the fundamental square. At each time step, during the integration of the equations, the interpolation is called in order to determine how the particle will move and which is its value of energy given its position in space. Like all interpolation, it produces an error while doing its main function, naturally less dense the grid greater will be the error introduced. Having said that it has been investigated if varying the N^o of points on the fundamental square grid can influence consistently the error on energy and if it is possible to obtain results of higher quality simply by increasing the above-mentioned number. This fact would be very valuable, it would mean that the model is solid and that it doesn't contain a lot of approximations. Naturally, a higher number of points on the grid means that the memory required for the calculations will be larger and it is necessary to make some compromises in order to leave the code of practical use. Some tests have been done in order to verify this, for both GC and Full Orbit mode the code has been executed while varying only the N parameter, the latter is the N^o of points on each side of the fundamental square, naturally, it is clear that:

$$Total\ N^o\ Points = N * N. \tag{4.8}$$

The code has been executed for the couple `TimeStep` values chosen in the previous subsection. It is possible to plot the results in graphs like the one of Fig. 4.5 and Fig. 4.7 in order to analyse the influence of the grid on the solution. From both the plot of Fig. 4.13 it is possible to notice a considerable influence on the value of the error for all time cycles, the trend and behaviour is almost the same for all the different set of data. At each doubling of the value of N the error becomes smaller, obviously, the plot in a semi-logarithmic graph helps in seeing the differences. Given that the quality of results improves it has to be decided which value of N one should take, as anticipated increasing the latter demands a great amount of memory while the simulation is running. While running from a standard laptop one should stop at $N = 4096$ to avoid problems with memory. Given that for these kinds of calculations usually, one uses a cluster, the availability of memory is not a big deal, it has been tried to analyse how the spatial discretization influences the time required for the calculations in order to see if we should make a compromise or if it is possible to use the higher value of N. The code has been executed for both GC and Full Orbit mode in a super-diffusive regime for various values of N, given that the code has been executed on the servers there isn't any problem with memory and the computational time is not so affected by increasing the spatial discretization.

$$N = 4096 = 2^{12}. \tag{4.9}$$

All the graphs previously shown have been simulated with this value of N to remain consistent with the rest of the manuscript. This value guarantees a good quality of results without interfering too much with the performance of the code and without requiring a large memory for the simulation. In previous simulations, it has been mostly used the



(a) *rel_error* trends from different total number of points on the grid, GC mode in super-diffusive regime

(b) *rel_error* trends from different total number of points on the grid, Full Orbit mode in super-diffusive regime

Figure 4.13: Influence of spatial discretization on energy error in both GC and Full Orbit mode, super-diffusive regime

value of $N = 1024$, the integration of energy allowed us to understand better what value was more correct and it was very important to make the project successful.

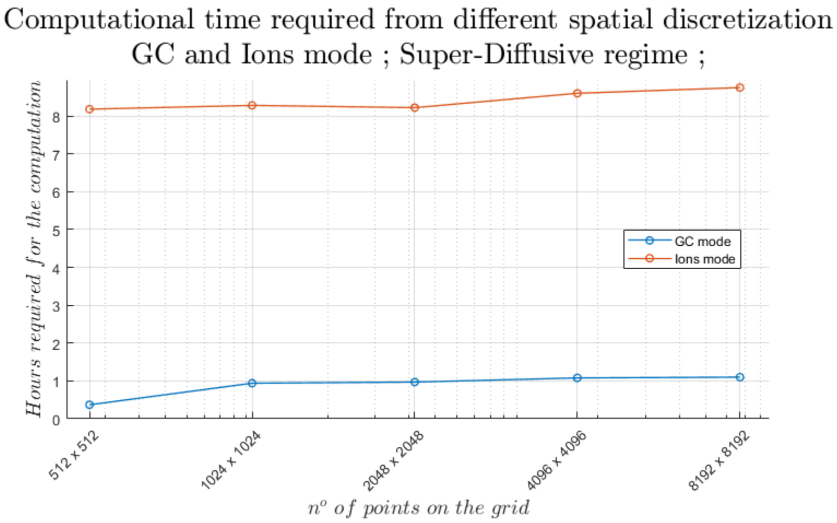
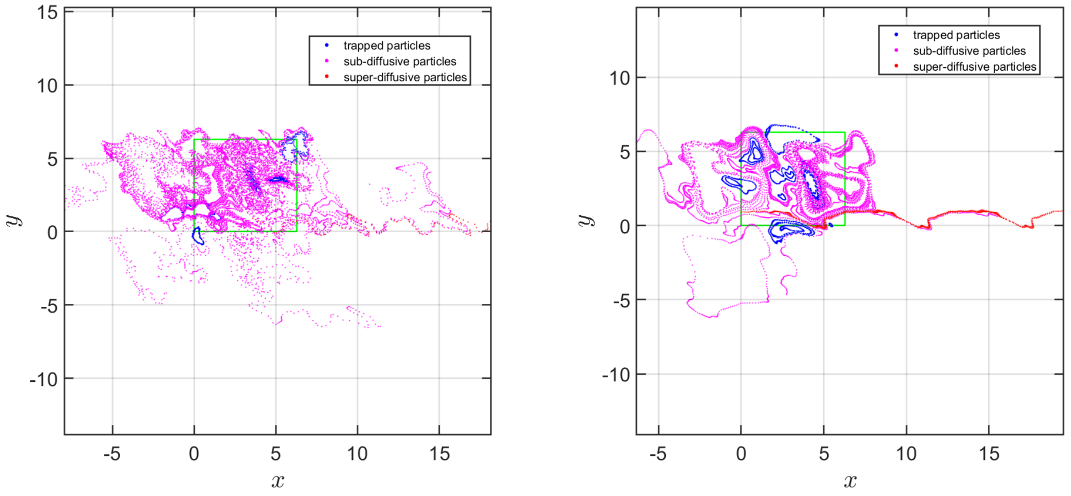


Figure 4.14: Time required for calculations for various value of N, GC and Full Orbit mode, Super-Diffusive regime.

4.3 Influence of physical parameters

Now that the results from the simulations can be considered reliable it is time to start investigating the dynamics of the guiding centre in the turbulent potential. It is important to remember that the potential will be the same for all the simulations, it can be changed by simply changing the seed of the random phases generator, therefore the dynamics is strongly influenced and it is not the aim of this work, to assess the influence of a different potential on the dynamics. As anticipated the code works through the initialization of the particles inside the fundamental square and it studies the evolution of the global system through the use of the governing equations implemented and derived in the previous chapters. By looking at multiple Poincaré sections, generated from multiple sets of parameters, it has been noticed that the combination of parameters can strongly influence the dynamics and the behaviour of the system. While looking at Fig. [4.15] it is clear that the combination of different parameters leads to different behaviour of the system. In addition to that, it is possible to evidence particles that have different behaviours, this is why, within the code, it has been implemented a method to distinguish these various types. One can clearly see the particles that are classified as trapped, they are easily recognisable given that they orbit forming islands. One can also distinguish some particles that are characterized by an elongated trajectory along a preferential direction, when increasing the number of time cycles `Tf` this fact is even more evident. The remaining particles (in magenta) have a more isotropic behaviour, they appear to be characterized by more chaotic behaviour with respect to the others. Once these three hypothetical types of particles have been identified it has been tried to



(a) Poincaré section generated from `A` $= 0.7$, `eta` $= -0.20$ and `rho` $= 0.10$.

(b) Poincaré section generated from `A` $= 0.7$, `eta` $= 0.14$ and `rho` $= 0.30$.

Figure 4.15: Poincaré sections generated for a bunch of particles for a different set of physical parameters in order to highlight the differences in the dynamics.

distinguish them directly within the code through the use of some geometrical criteria as described in Subsec. 3.4.2. After the computation of the trajectories the trend of mean square displacement over time is obtained, this is done only for particles that diffuse, not for trapped ones. Once the trends are obtained, as anticipated in Subsec. 3.4.6, two different fittings are performed on the two different trends associated with the two types of particles. We briefly recall the equations associated with the fittings:

$$\textbf{Linear regression:} \;\; f_{sub} = D \cdot t + Cst$$

$$\textbf{Power law:} \;\; f_{super} = (a \cdot t)^{b}$$

In Fig. [4.16] are shown the trends of MSD and the associated fittings for the same results of Fig. [4.15b] (same set of parameters), it is clear that the two types of particles have completely different behaviour, the trend for the particles associated with an elongated trajectory is almost quadratic while in the other case, the linear regression is well approximating the trend. Given these results, it has been decided to classify the particles

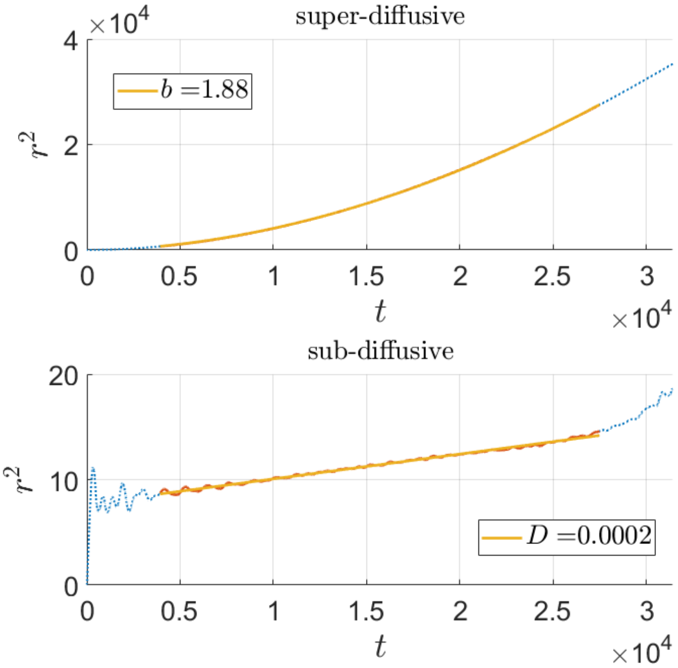


Figure 4.16: *MSD* trends and associated fittings, power law for super-diffusive and linear-regression for sub-diffusive.

associated with an elongated trajectory as super-diffusive particles while from now on we will refer to the others as sub-diffusive. This, despite the fact they are not really satisfying some known criteria officially recognized and used to define and identify these two types of particles by the scientific community. To briefly recap, from some early results, it has been noticed that the dynamics of the system is strongly influenced by the physical parameters and that whatever the combination of input parameters, different types of particles, associated with different behaviours are present. While looking at Fig.[4.15], it is clear that a different percentage of super-diffusive is present in the two cases, this is valid for trapped particles as well. One might wonder which is the influence of physical parameters on this specific aspect, which is the influence on the percentages of the various types of particles. In order to analyse this aspect a lot of simulations have been performed. Two of the three physical parameters (`A` has been fixed) have been varied simultaneously, this has been done simply through the insertion of the following inputs in `gc2d_dict.py`:

$$\texttt{rho} = \texttt{xp.linspace(}0,\ 0.3,\ 128\texttt{)}$$

$$\texttt{eta} = \texttt{xp.linspace(}-0.2,\ 0.3,\ 128\texttt{)}$$

The insertion of these inputs results in a high number of sub-simulations, all the possible combinations of parameters are tested, the total number is equal to the total number of permutations between these values, 16384 to be exact, a sort of grid is created, each point of the grid corresponds to a different combination of `rho` and `eta` and it is associated with a specific simulation. For each simulation (point of the grid) some interesting quantities are computed, we briefly list them:

- Trapped particles percentage.
- Sub-diffusive particles percentage.
- Super-diffusive particles percentage.
- Diffusion coefficient D associated with the linear regression performed on MSD trend of sub-diffusive particles.
- Intercept value Cst associated with the linear regression performed on MSD trend of sub-diffusive particles.
- Coefficient of determination R^2 associated with the linear regression performed on MSD trend of sub-diffusive particles.
- a coefficient associated with the power law fitting performed on MSD trend of super-diffusive particles.
- b coefficient associated with the power law fitting performed on MSD trend of super-diffusive particles.
- Coefficient of determination R^2, associated with the power law fitting performed on MSD trend of super-diffusive particles.

Results from different simulations are stored inside a `.txt` file that can be analysed after all the sub-simulations have been performed. In this way, it is possible to obtain multiple information on how the dynamics is influenced by the variation of `rho` and `eta` in a very efficient way. The simulations were characterised by `Ntraj` $= 1024$, `Tf` $= 5000$, `N` $= 2^{10}$ and `TimeStep` $= 0.05$. The use of a larger time-step (ten times larger than the one chosen from the Time-step analysis in the previous section) and of a lower number of points on the potential grid is due to the fact that following some early tests, performed to estimate the average time required for each sub-simulation, and given the total number of CPU cores available (50 in our case), the estimated time to complete the entire simulation using the nominal values would have been greater than 200 days, it wouldn't have been acceptable. As one can check on Sec. 4.2.2, the quality of the results is however acceptable, they can be considered reliable. Also, the influence of N doesn't generate excessive numerical errors as one can check in Sec. 4.2.4. Given that choice of input parameters, the simulation took 20 days, the time required for the computation is exactly inversely proportional to the value of `TimeStep`, and the occupied RAM memory was compatible with the smooth operation of the machine over the long period on which the simulation took place. The output `.txt` file has been post-processed through a MATLAB script, thanks to the latter it is possible to produce some sort of heat maps for the interesting quantities. The plot of some maps is not of great interest to us, the most interesting ones are the percentages maps and the one of b, these ones are giving us a lot of information about the dynamics of the system. Before analysing the results it is necessary also to point out that the various maps are filtered on a minimum value of the percentage of the specific type of particles. For instance, in case the number of super-diffusive particles is under 5% in a certain region of the map, then the corresponding area is coloured in white, and the values in the grid are substituted by *nan*. The same is done with the corresponding point in the b map, the filter is applied on the percentage map but the effect is also applied to maps made for that specific type of particle.

4.3.1 Effect of ρ and η on the particles percentages

As anticipated, the influence of ρ and η on the dynamics strongly influences the percentage of different particles. Fig. [4.17] shows the maps for every type of particle, different palettes have been used to identify the various types, in order to distinguish between trapped, sub-diffusive and super-diffusive cyan, magenta and red palettes have been respectively used. As it is possible to notice from the colour bars, the darker the colour, the higher the percentage. The results are quite interesting, one of the most important aspects is the behaviour of trapped particles, since we are interested in studying the transport of particles in this mock potential then trying to understand which regime guarantees higher confinement is very important. While looking at the top-left plot (the one with a cyan palette) it is clear that an increase in the value of ρ regularises the dynamics, this means that a higher fraction of particles will be confined inside the fundamental square or in the nearby. Since a gyro-average is performed, if the Larmor radius increases then the average will be exploited on a wider region, given that the potential is turbulent and that it is characterised by islands structure also at a very little scale, exploiting the average on a wider region results in a completely smooth guiding centre potential, naturally this affects the dynamics of particles. A smoother (guiding centre) potential will lead to a higher percentage of particles that will be trapped, they are trapped in the sense that they orbit on closed trajectories without the possibility of escaping. This result is expected, it is a sign that the model works properly. It is also possible to conclude that large values of $|\eta|$ lead to a total absence of trapped, the behaviour is not totally symmetric but this is the general trend. We now switch to the investigation of the other two plots, as it

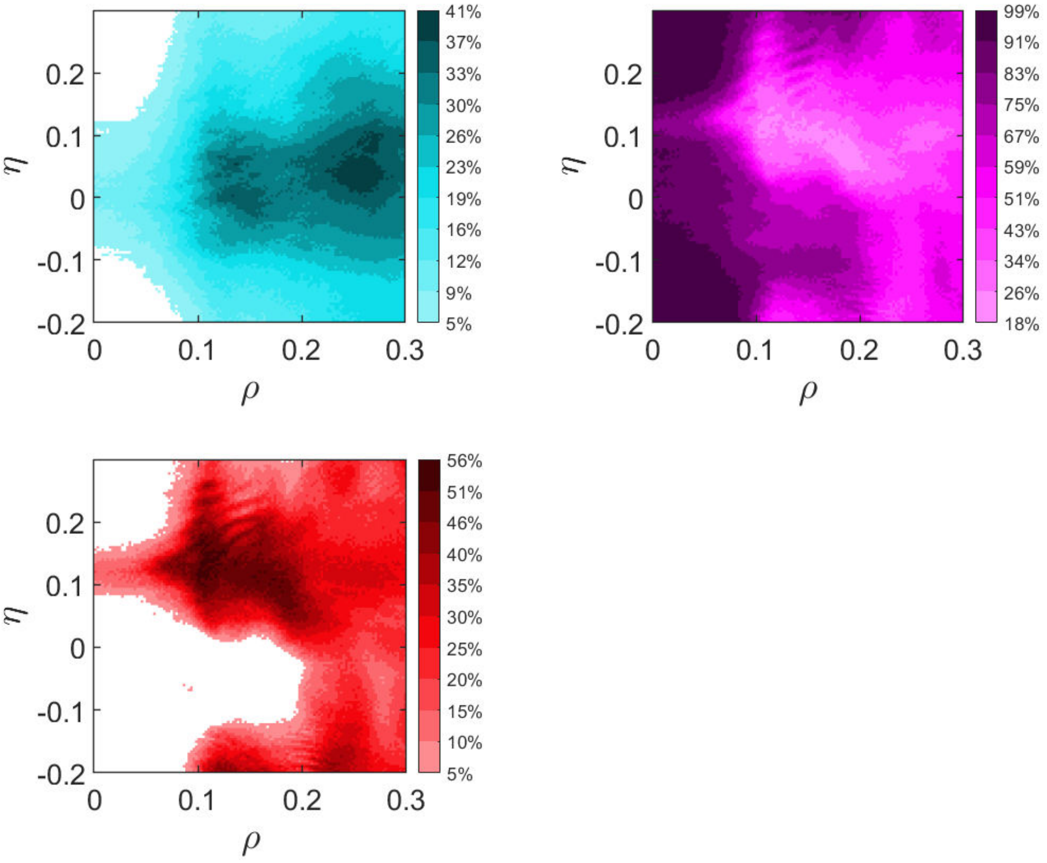


Figure 4.17: Percentages maps for all types of particles, specific palettes identify specific types of particles, cyan for trapped, magenta for sub-diffusive and red for super-diffusive.

is possible to see, there is a sort of connection between the two maps. It is possible to evidence a sort of complementary behaviour between super and sub-diffusive particles, for ρ values higher than 0.1 regions characterised by a high percentage of the super-diffusive ones presents also the lowest percentage of sub-diffusive and vice-versa. This behaviour is dominant until $\rho \geq 0.25$, at that point the behaviour is kinda homogeneous, this is due also to the regularisation of the dynamics. What is sure is that the value of η strongly influences the behaviour of the global system with ρ in the range between 0.1 and 0.25. For values of ρ lower than 0.1 the chaotic behaviour associated with sub-diffusive particles is dominant, since the average is performed on a very small region the forces to which the guiding centre is subjected are almost equal to the ones the particle is feeling, the G.C. potential the potential has not been smoothed and the presence of the turbulence does not allow particles to orbit on closed or regular trajectories, as anticipated chaotic behaviour is dominant in this region. It is also possible to notice how the percentage of super-diffusive is really low for very large areas, it indicates that the mechanism leading to this kind of behaviour is quite fragile and that it should be investigated better.

4.3.2 Effect of ρ and η on b value

Since the chaotic transport of particles is very well investigated in the literature, as well as the mechanism that leads to the entrapment of particles it has been decided to better investigate the comportment of particles associated with the almost (and in some cases proper) quadratic trend of *MSD* over time. As has been anticipated previously in this section, these particles are not properly satisfying the criteria that are usually used to classify these entities as such. We classified these entities as super-diffusive just because they are characterised by having much higher diffusion coefficients. While looking at Fig.[4.18] some interesting aspects can be listed, the plot describes the comportment of the system for various combinations of ρ and η, it is clear that it can be totally different depending on the input parameters. The most important and evident fact is that starting from $\rho = 0.2$, for a wide range of η, an increase in `rho` corresponds to a high increase in the quadratic behaviour of the trend. This, in practice, results in particles reaching higher r^2(`Tf`) (for an equal chosen value of `Tf`), this means that some particles travel more and more in space consequently to an increase of ρ. This fact is very interesting and apparently, it has no specific motivation. The value of b massively increases in a very thin region passing by an average value of 1.3 to an average of 1.78 while ρ has passed from 0.22 to 0.24, as anticipated this is valid for a wide range of η, from -0.1 to 0.2, it is not only present for values of η where (for lower values of ρ) the amount of particles is considerably high. It is also evident that lowering ρ leads to the destruction of this

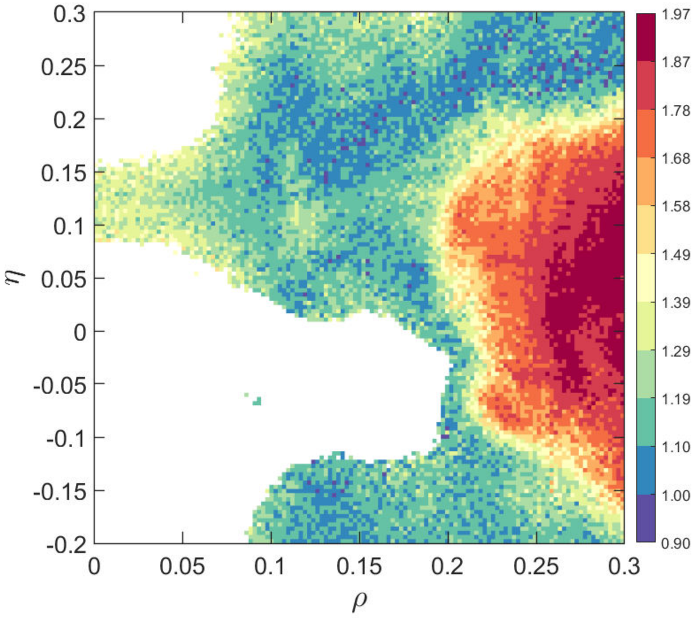
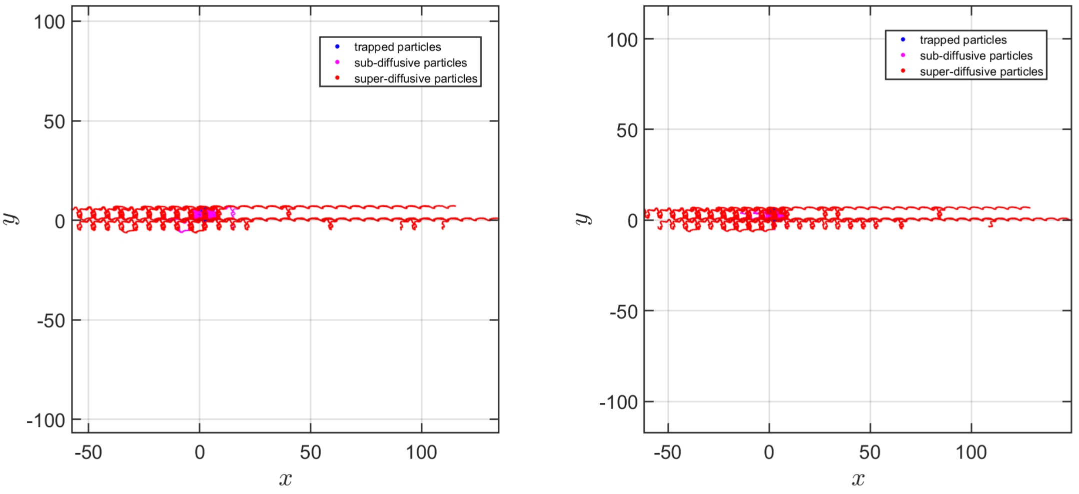


Figure 4.18: Map of b for different combinations of ρ and η, we recall that this value is associated with the exponent in the power law fitting $(a \cdot t)^b$ performed on the trend of *MSD* over time for particles classified as super-diffusive.

mechanism, by analysing regions characterised by a lower value of b and by investigating transition regions this should be visible. Another noticeable fact that one can observe in Fig.[4.18] is that in the region identified by $\rho \in [0, 0.05]$ and $\eta \in [0.1, 0.15]$, the system present an unusual average value of b. In fact, if we compare the values from this region with the ones from the region just to the right, for $\rho \geq 0.05$, we can clearly see that the values are higher, this is unusual given that the other mechanism, leading to high transport of particles, works in the opposite way. This region should be further analysed in order to understand if the high values of the exponent of the power law fitting are due to the same mechanism or not and, to understand which is the influence of η in this scheme, we clearly see that only for $\eta \in [0.1, 0.15]$ a considerable amount of super-diffusive particles is guaranteed. In summary, it has been evidenced an anomalous comportment of the system, the investigation of this specific behaviour is based on Fig.[4.18], it clearly shows which are the areas of concern to study this phenomenon, the analysis started by investigating concerning region characterised by high ρ values in order to understand what is the dominant mechanism in that region.

4.4 Investigation of anomalous super-diffusive behaviour

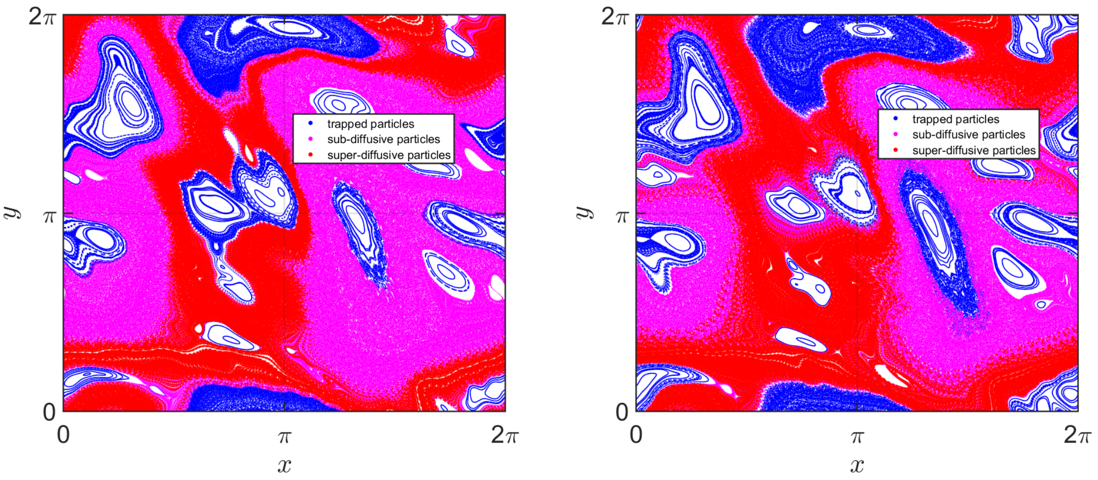
The analysis of the region characterised by unusually high values of b started by plotting a lot of Poincaré sections from simulations made by using high values of ρ. Fig. [4.19] shows a couple of full Poincaré sections generated by a discrete number of particles (`Ntraj` $= 500$) over a considerable number of time cycles (`Tf` $= 1500$) in order to highlight the characteristic comportment of super-diffusive particles. From these figures it is pretty clear why the trend of *MSD* has that particular shape, some particles (not all of them) are moving along a preferential direction (x axe in this case, but depending on the potential it may be the y axe) with a speed well above the average. It can be said that they have a sort of ballistic behaviour, especially the ones going toward the right. It is true that the input parameters are different in the two cases but by looking again at Fig. [4.18] it is clear that they belong to the same dynamic region, this is why the final Poincaré section is almost identical. In both cases it is possible to notice that the particles going to the right and to the left are following a path with a different shape, despite this in both cases is a sort of periodical motion, a repetitive path. Since the potential is periodic in time and space this should not be considered such an unusual fact but it seems some of them seem to not deviate at all from the initial path, the resulting one after only a few cycles have passed. This is very interesting and it may be the key to the concerning phenomena. In order to better analyse and understand this phenomenon it has been decided to get advantage of Modular Poincaré sections, if the path is that repetitive then it should be clearly evidenced in that kind of plot. Fig. [4.20] shows the modular Poincaré sections corresponding to the ones of Fig. [4.19]. By looking at Fig. [4.20a] and Fig. [4.20b] we can easily recognise the two preferential paths that the particles are following. The one at the bottom is the one leading the particles to the right while the one starting at the top-right of the figures, and that intricately pass through the figure, is leading the particles to the left. Through the modular version of this kind of plot, it is easier to compare and look for differences between the results of the two simulations. In



(a) Full Poincaré section generated from `A` = 0.7, `eta` = 0.14 and `rho` = 0.30.

(b) Full Poincaré section generated from `A` = 0.7, `eta` = 0.12 and `rho` = 0.27.

Figure 4.19: Full Poincaré sections generated with `Ntraj` = 500 and `Tf` = 1500 for different sets of physical parameters to investigate the region where anomalous super-diffusive behaviour has been highlighted.



(a) Modular Poincaré section generated from `A` = 0.7, `eta` = 0.14 and `rho` = 0.30.

(b) Modular Poincaré section generated from `A` = 0.7, `eta` = 0.12 and `rho` = 0.27.

Figure 4.20: Modular Poincaré sections generated with `Ntraj` = 500 and `Tf` = 1500 for different sets of physical parameters to investigate the region where anomalous super-diffusive behaviour has been highlighted.

particular, the red paths associated with the case with $\rho = 0.3$ are tighter with respect to the other case, this suggests that, despite the fact that the percentage of super-diffusive is almost the same for the two cases (21.5% for $\rho = 0.3$ and 22.5% for $\rho = 0.27$) these entities are following more accurately the same periodic path, this is why the simulation characterised by having `A` = 0.7, `eta` = 0.14 and `rho` = 0.30 is more interesting for our goal. Anyway, from the analysis of these two last figures, it is even more clear that the key point is a more accurate analysis of these preferential paths that super-diffusive particles are strictly following. Our efforts are therefore concentrated towards these ones, therefore, it is necessary to look for a way of efficiently analysing these regions without computing the trajectories that are not of our interest.

4.5 Local analysis

In order to study the above-mentioned phenomena it has been decided to perform some local analysis starting with all the particles in a small region of space inside the preferential path at the bottom of the modular Poincaré sections. This is very useful in order to obtain a lot of information about the particles moving along the preferential path without the necessity of increasing enormously the initial number of particles to obtain a good density in the interesting region. Thanks to this strategy a lot of computational time has been saved, and feedback from simulations has been obtained much faster. Later in this chapter, this strategy will be extremized to perform a certain type of analysis. As anticipated we are now interested only in the dynamics of super-diffusive particles following a very repetitive path along their trajectory. The first noticeable result from the local analysis is that, contrary to what was thought, a lot of the particles are not always following the same curve at each passage in the modular Poincaré section, it is not that different but this fact can't be neglected. Nevertheless, there are also a lot of particles that follow very accurately the same path at each passage, these are the particles in which we are really interested. They have been extracted from the modular Poincaré section of Fig. [4.20a], we recall that from now on, only results generated with `A` = 0.7, `eta` = 0.14 and `rho` = 0.30 will be shown. The trajectories of interest are shown in Fig. [4.21], different colours have been used for different hypothetical types of particles, they will be explained in the following rows of this section. While looking at Fig. [4.21] it is

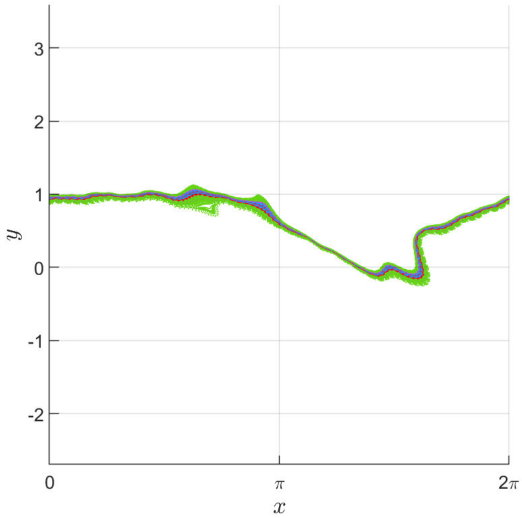


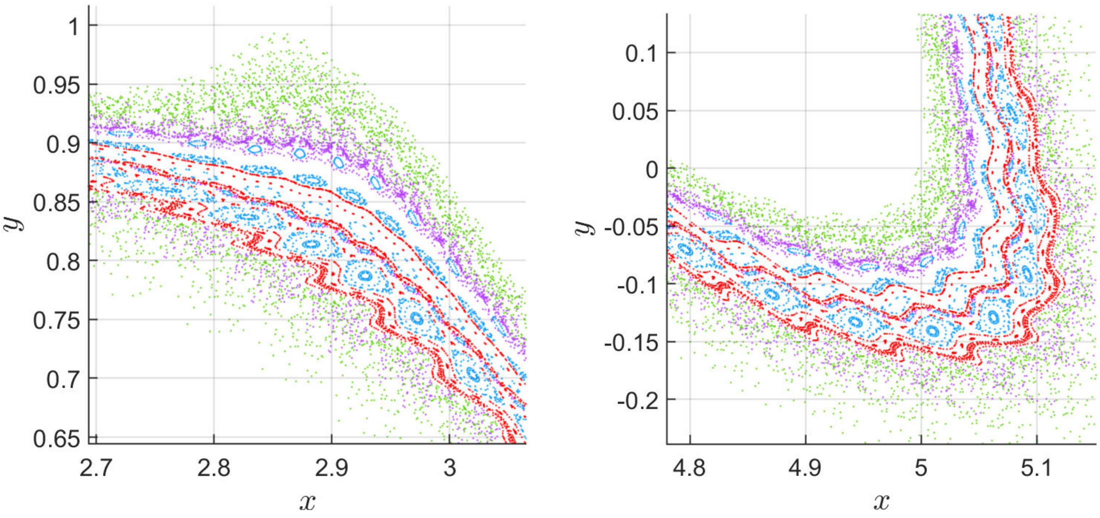
Figure 4.21: Isolated super-diffusive particles trajectories in the modular Poincaré section produced through the usual set of parameters: `A` = 0.7, `eta` = 0.14 and `rho` = 0.30 .

clear that except for the particles coloured in green, the behaviour we were looking for is present even for a relatively high number of time cycles (`Tf` $= 2000$ in this case). In order to better investigate the behaviour of the particles it is necessary to zoom in on the figure in some interesting areas, where the trajectories seem to be more distant from each other and consequently can be better analyzed, I chose to zoom in the following x ranges:

$$x_1 \in [2.7, 3.1]$$

$$x_2 \in [4.7, 5.2]$$

In these particular ranges of x, the trajectories are kinda curving, their behaviour is better highlighted. Fig. [4.22] shows both the investigated ranges, at this level one can better understand the use of different colours. As anticipated, it is easy to recognise different types of particles inside the zoomed versions of the Poincaré section, we can list them while specifying also the colour thanks to which they can be recognized:



(a) Zoomed modular Poincaré section showing isolated trajectories, focus on x_1 range, generated from `A` $= 0.7$, `eta` $= 0.14$ and `rho` $= 0.30$.

(b) Zoomed modular Poincaré section showing isolated trajectories, focus on x_2 range, generated from `A` $= 0.7$, `eta` $= 0.14$ and `rho` $= 0.30$.

Figure 4.22: Modular Poincaré sections generated with `Ntraj` $= 500$ and `Tf` $= 1500$ for different sets of physical parameters to investigate the region where anomalous superdiffusive behaviour has been highlighted.

Invariant: Highlighted in red, these particles always follow the exact same path at each passage.

Invariant forming islands: coloured in cyan, they don't follow the exact same curve but still, they can be considered invariant particles. Their main characteristic is that, at each passage, they pass in a slightly displaced point with respect to the previous passage, the resulting effect is that some chains of islands are produced, note that each island has a characteristic size and that it is produced by a specific particle.

Not invariant: coloured in violet, these particles are kinda jumping on a different curve at each passage, they cannot be considered as invariant.

Almost chaotic: Highlighted in green, their behaviour is similar to one of the violet particles but, as it is clearly visible from the non-zoomed Poincaré section, they are producing very different curves (relatively speaking).

The most noticeable fact coming from Fig. [4.22] is that some invariant particles are present. In our case, invariance is the property that characterises particles that follow exactly the same periodic path, it is easy to recognise their trajectories since they appear as if the particle has passed only once. Another very interesting fact is the particular chains of islands produced by the particles coloured in cyan. In literature one can find out that this behaviour is known and deeply analysed [OC95], their presence means also that the computation is done correctly. As anticipated these can still be considered invariant particles. Away from the centre of the channel particles are starting to jump more from one path to another, it seems that only particles at the centre are characterised by being invariant. In addition to that, by looking better at Fig. [4.22], it is very interesting to notice that green and violet particles are never crossing the border formed by the invariant curves, this behaviour is known as well in literature [Bal98], it is really important to notice this fact, the presence of invariant tori (trajectories) leads to the presence of a transport barrier that doesn't allow to be crossed perpendicularly. In summary, the main result from the local analysis is that some invariant particles following an opened path are present, one has also to remember that also trapped particles are actually forming some invariant curves, the main difference is that they form a closed path. We can imagine invariant particles as the frame of the Poincaré sections, they are kinda guiding the dynamics of the system. Fig. [4.23] shows a filtered version of Fig. [4.20a], only invariant particle trajectories are shown. It is very interesting to compare the latter with the non-filtered version of the plot, it is clear that these invariant curves are also determining the dynamics of chaotic particles.

4.6 Meandering tori

Following all the interesting results we obtained, a little comparison with what was already present in literature [MW09], [dCNGM96], has been made, this has highlighted the presence of many points in common with a particular phenomenon, the so-called *meandering tori* behaviour. The latter match with the behaviour of the particles in our system, some invariant tori contained within a channel are separating the phase-space, the name comes from the shape of these invariant curves, they are quite similar to the meanders formed by rivers over the years. The meander is separating the phase space in the sense that it forms this sort of transport barrier blocking particles by crossing the border of the meander perpendicularly. Naturally, it doesn't block particles going along a trajectory that is parallel to the meander border itself. As anticipated the transport barrier is present in our result, this is the main reason why we are trying to refer our results to *meandering tori* theory. it is also interesting that we noticed the presence of this phenomenon thanks to an increase in particles transport while this phenomenon itself

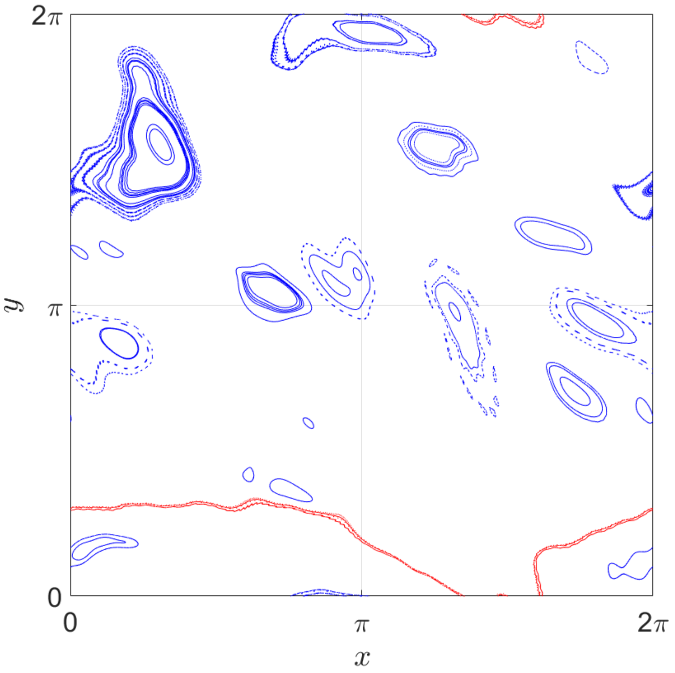


Figure 4.23: Filtered modular Poincaré section showing trajectories of invariant particles only, various types of particles are distinguished by the usual colours, blue for trapped, red for super-diffusive.

is decreasing in certain sense, it is a bit counterintuitive. In reality, we still don't have the confirmation that the phenomenon is the same at 100%, some additional results are required, *meandering tori* is always associated with the presence of a so-called twist-less invariant torus inside the meander. This torus is also called the shear-less curve, it is characterised by having the maximum value of rotational number (also-called winding number) when compared to the values associated with the other tori within the meander. In literature the presence of the transport barrier is always associated with a non-twist invariant curve [MW09], [OC95], [Bal98], [dCNGM96]. In order to confirm its presence it is necessary to compute and plot the values of winding numbers for particles starting in different positions inside the meander, the trend should present a local maximum and consequently a non-monotonous trend. In order to perform this kind of analysis, it is necessary to change again the initialisation of the particles' position. Now the particles need to be started from a vertical segment that cuts the meander itself, by starting a lot of particles on this little segment we obtain a sort of continuous trend of winding numbers for particles starting from different positions.

4.6.1 Rotational number computation

The rotational number is usually computed for exploiting invariance analysis on systems like the one under analysis, in literature these analyses are widely used to obtain quantitative information about the chaotic behaviour of particles, they are used especially to distinguish between chaotic and quasi-periodic dynamics [SM20]. In our case, the winding number is computed to check for the presence of the above-described shear-less curve. First, it is necessary to introduce its expression:

$$\rho = \lim_{n \Rightarrow \infty} \frac{F^n(\theta) - \theta}{n}. \tag{4.10}$$

For our system, θ is the initial x coordinate of the trajectory and $F^n(\theta)$ is the value of x coordinate at the n_{th} time cycle. It is giving us information on how much is the average space that particles travel during a generic temporal period. Usually, it is computed through the use of a weighted Birkhoff average so that a better convergence is obtained:

$$WB_S(\rho(\theta)) = \frac{1}{C_S} \sum_{n=1}^{S-1} \omega\left(\frac{n}{S}\right) \rho(F^n(\theta)). \tag{4.11}$$

With:

$$C_S = \sum_{n=1}^{S-1} \omega\left(\frac{n}{S}\right) \ ; \ \omega(t) = \exp\left(-\frac{1}{t(1-t)}\right).$$

Where ω is the so-called bump function. As anticipated, this is computed for each trajectory that starts on the segment that is crossing the meander, the segment is vertical and positioned in $x = \pi$, while the range of y is $[0.575, 0.62]$. Trajectories need to be very long in order to obtain a good convergence of winding number computed with weighted Birkhoff average, this is why it has been decided to set `Tf` $= 7500$. In addition to that, it is also necessary to have a good discretization along y to obtain clear results, this is why, despite the fact the segment is very short, a very high number of trajectories has been computed, `Ntraj` $= 1024$ in fact. Since the quality of results is very important in this case, it has been chosen to use a smaller time step with respect to the one that is usually set, in this case, `TimeStep` $= 0.001$, it is still quite greater than the one used for full orbit mode but it strongly affects the time required for the computation.

4.6.2 Rotational number analysis

Fig. [4.24] shows the trend of winding number for the different particles starting positions over the segment. First, it is important to remember that the convergence is assured only for invariant particles, this means that here we have further verification of the fact that the trajectories under analysis are invariant curves. In case one tries to compute the same quantity for particles starting in a region leading to chaotic behaviour, the result is completely different, you don't have any convergence. While looking at Fig. [4.24] one can recognise regions characterised by different behaviours, starting from the left we have a bell-shape curve, it is easy to recognise the local maximum that we were looking for, and then we have a region with a constant ρ value while on the right in the figure, we have a sort of devil staircase. Since the bell shape and the latter need a more detailed explanation we start explaining the region at constant winding number. It is known that this particular trend is associated with particles that are forming chains of islands, as it is possible to see in Fig. [4.22], some islands are containing other islands characterised by having a smaller size, each one is produced by different particles but it is known from

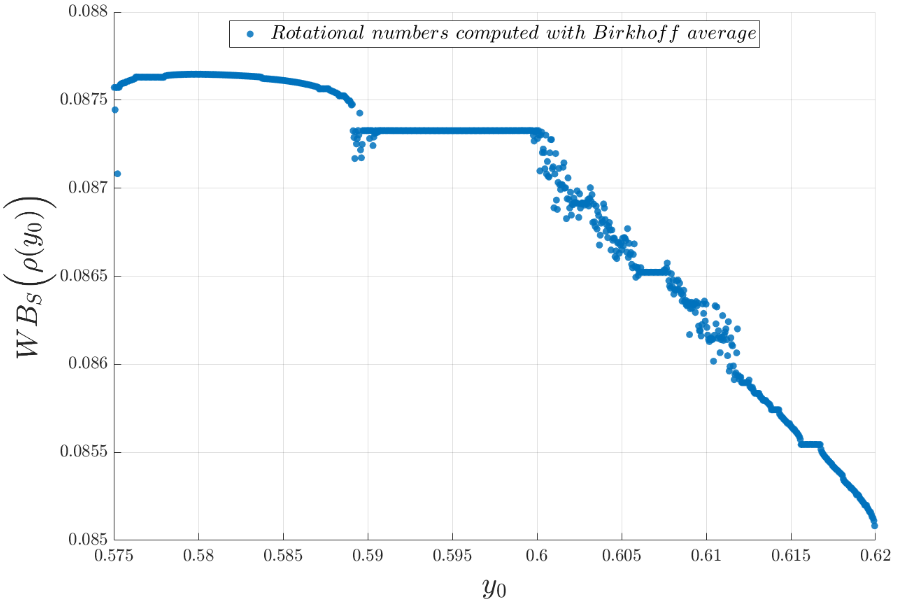


Figure 4.24: Rotational number plot from the local analysis made through the initialization of particles position on a segment, generated with `A` $= 0.7$, `eta` $= 0.14$ and `rho` $= 0.30$.

the literature [OC95] [MW09] that particles within the same island are characterised by the same value of the rotational number. This means that particles initialised for $y \in [0.59, 0.60]$ are producing a single chain of islands. It is now time to analyse the bell-shape curve, Fig. [4.25] is a zoomed version of Fig. [4.24], it clearly emphasizes the presence of the local maximum together with the non-monotony of the trend. This is the final confirmation of the presence of the twist-less invariant torus. This particular trend is known as the bell-shape curve, it is the typical rotational number trend where a twist-less invariant torus is found, we can consider it as a further confirmation of the validity of *meandering tori* theory. Another noticeable fact is the presence of a devil staircase, particles initialised for $y \in [0.61, 0.62]$ are producing this kind of effect. A devil staircase is a particular type of staircase where the steps are not equally spaced other than the fact that their height can be really different. Fig. [4.26] is a zoomed version of Fig. [4.24], the above-mentioned shape is clearly present, as anticipated previously, we know that regions at constant ρ are part of the same islands, this means that the trend behaviour is determined by the presence of some islands of different size, it has no bearing on the presence of the shear-less curve but it is quite interesting to be able to highlight and confirm the presence of particular phenomena (chains of islands) directly from this particular trend. In summary, in this section the presence of a twist-less invariant torus has been confirmed, this is the phenomenon associated with the anomalous increase of transport along a preferential direction detected in the b map of Fig. [4.18]. We didn't understand why this is present in our case, the potential is definitely one of the causes

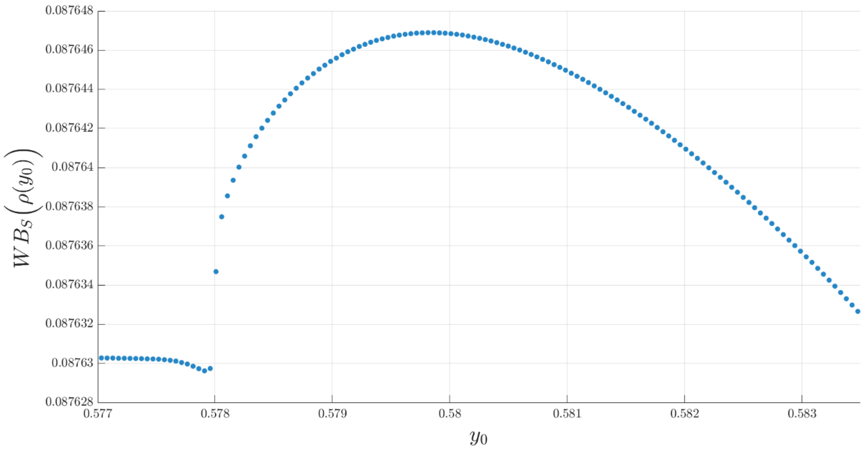


Figure 4.25: Focus on the bell-shape curve in the rotational number plot, the local maximum corresponds to the twist-less invariant torus.

outside the model but we fully characterised and analysed its presence in our results.

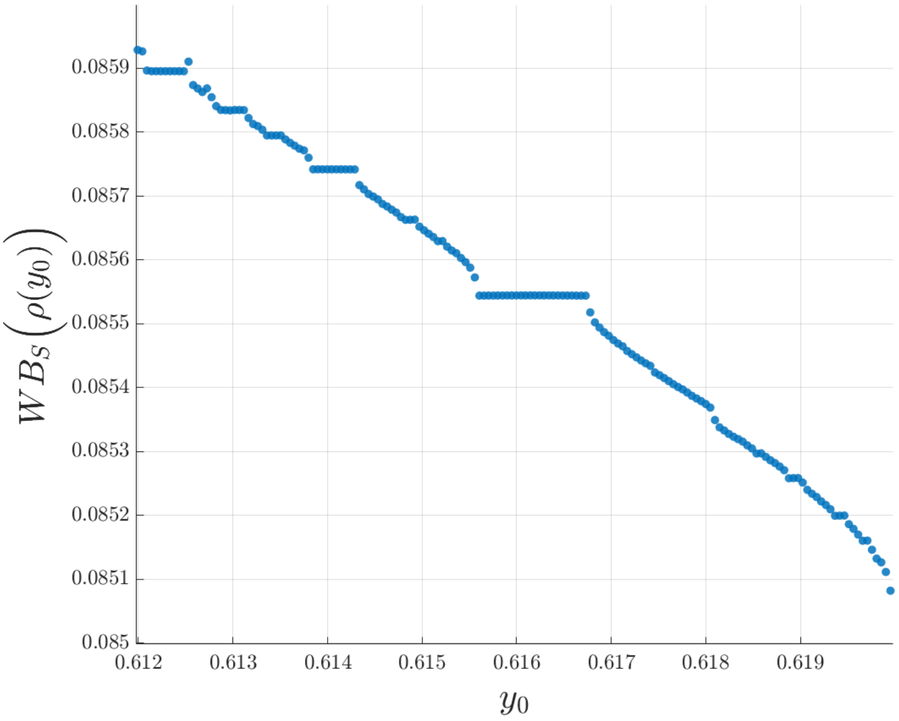
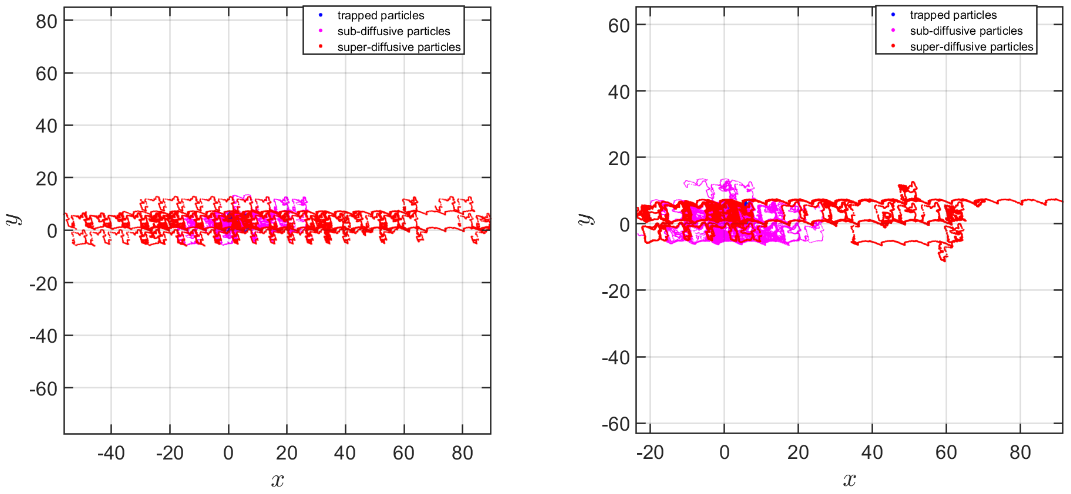


Figure 4.26: Focus on devil staircase in rotational number plot, regions with constant ρ corresponds to zone within islands showed in Fig. [4.22].

4.7 Comparison with full-orbit dynamics

The main aim of the implementation of full-orbit mode is to check on the fact that the anomalous super-diffusive behaviour highlighted in the previous sections is not the result of errors or bad approximations in the considered gyro-kinetic model. Since the system is chaotic, one cannot expect a full accordance between the Poincaré sections in output from the two different modalities but, it is important that the global behaviour is the same. This is why, in addition to the possibility of computing the trajectories of the particles themselves, `fo2gc` has been implemented. The trajectories of ions and guiding centre are simply shifted by the Larmor radius but this can be quite evident, especially in the modular Poincaré sections. The analysis simply consists in plotting full and modular Poincaré sections coming from the two different modalities, naturally by using the same set of input parameters. This has been done for different combinations but here we show only the results coming from a single set. Fig. [4.27] shows the comparison of full Poincaré sections between the two modes. It can be said that the global behaviour is the same for the two dynamics, it can be clearly seen that the global behaviour is quite similar. The super-diffusive particles are following paths with almost the same shape, the diffusion coefficient is similar, the particles are doing the same amount of space, and the extremes on the axes are almost the same. The amount of particles characterised by being super-diffusive is lower in the case of full orbit, now that we know which is the mechanism leading to this kind of behaviour this result is quite expected, the dynamics is not regularised at

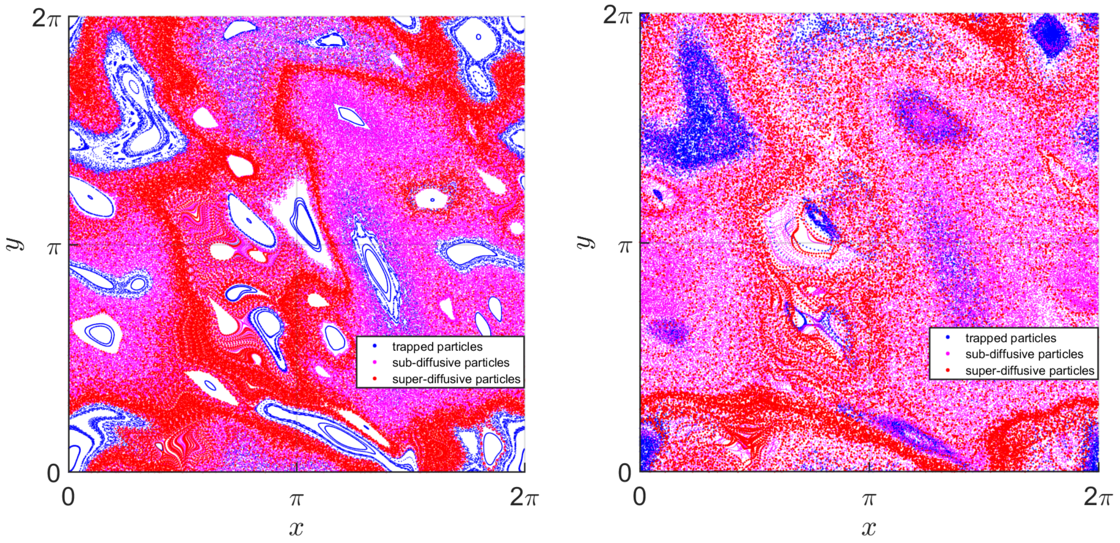


(a) Full Poincaré section showing guiding centre dynamics from guiding centre mode, generated with `A` = 0.7, `eta` = 0.10 and `rho` = 0.10.

(b) Full Poincaré section showing guiding center dynamics from full orbit mode and `fo2gc`, generated with `A` = 0.7, `eta` = 0.10 and `rho` = 0.10.

Figure 4.27: Comparison of guiding centre dynamics coming from different modalities in full Poincaré sections, on the left results from guiding centre mode while on the right results from full orbit mode.

all, ions are following a much more intricate trajectory and it is easier for them to go out of the meander leading to the ballistic behaviour. In this way, it is easier for a particle to become a chaotic one, with a non-periodic path. The most important thing we can conclude from Fig. [4.27] is that, in full orbit mode, there are some particles characterised by having an average speed that is in accordance with one of the ballistic particles in guiding centre mode, this means that the observed anomalous behaviour does not come from an error in the model, it is the most important fact coming from this analysis, it means the results can be considered quite reliable from the numerical point of view. It

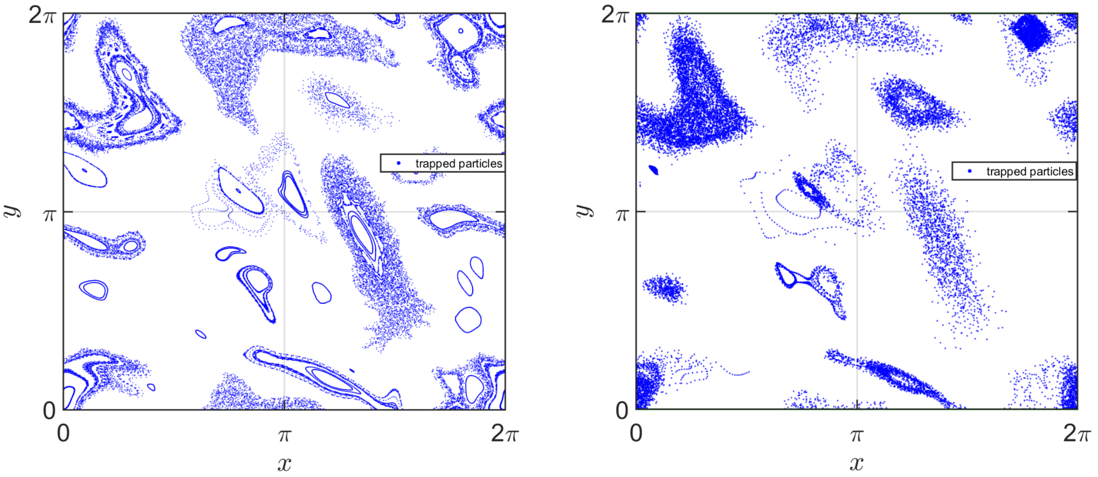


(a) Modular Poincaré section showing guiding centre dynamics from guiding centre mode, generated with `A` $= 0.7$, `eta` $= 0.10$ and `rho` $= 0.10$.

(b) Modular Poincaré section showing guiding centre dynamics from full orbit mode, generated with `A` $= 0.7$, `eta` $= 0.10$ and `rho` $= 0.10$.

Figure 4.28: Comparison of guiding centre dynamics coming from different modalities in modular Poincaré sections, on the left results from guiding centre mode while on the right results from full orbit mode.

is now time to switch to the modular version of the Poincaré sections, Fig. [4.28] shows the comparison between the results. Before analysing the results it is necessary to clarify a little thing, in the case of the results from the gyrokinetic framework, the solution is in a $2D$ phase space, and the latter is fully represented in the Poincaré section. On the contrary, the solution from full orbit mode is in a $4D$ phase space, by plotting the Poincaré section in that case, we are only showing a projection of the $4D$ phase space on a $2D$ one. The plots are comparable but actually, they are not the same thing, one has to remember it while looking at the figures. By looking at Fig. [4.28] we can find a good accordance between the two figures, the regions populated by chaotic particles are the same, and this is valid for trapped particles as well. It is also possible to notice that the super-diffusive are following the same path, as anticipated while analysing Fig. [4.27], the behaviour is a more chaotic path but it is easy to recognise the main path where *meandering tori* is leading the dynamics. This is a further confirmation of the fact that the increase in the transport of particles along the meander is not artificially generated by the considered gyrokinetic model. Since trapped particle trajectories are almost covered by the diffusive ones, it has been decided to plot these types of trajectories in the modular Poincaré sections. By looking at the comparison between the two plots of Fig. [4.29], the general behaviour is quite in accordance, as anticipated, the results from full orbit mode are more chaotic but this is expected knowing how the two models work. The islands have the same shape in a lot of cases, the invariant behaviour is much more present in the first case but again this is due to the regularisation of the dynamics. The fact of having the same sort of islands in the same points of the fundamental square is assuring us of obtaining the same kind of dynamics while considering all the types of particles, the trajectories of the chaotic ones are kinda squeezed between these islands but we notice also that the ballistic particle paths are kinda moving around the latter. It is possible to conclude that also the dynamics of trapped particles is in good accordance between the two modes, this, coupled with the fact that in both cases we obtain the anomalous super-diffusive dynamics reassures us a lot about the quality of the results, most importantly we are able to investigate the same type of phenomena using a reduced model. We have to remember that the simulations for full-orbit are much more expensive with respect to the one for the guiding centre, on average they take 10 times more time than a simulation exploited with the gyrokinetic model. This is the real power of gyrokinetics, being able to investigate the same dynamics in a much faster way. For instance, the three maps of



(a) Modular Poincaré section showing trapped guiding centre dynamics from guiding centre mode, generated with `A` $= 0.7$, `eta` $= 0.10$ and `rho` $= 0.10$.

(b) Modular Poincaré section showing trapped guiding centre dynamics from full orbit mode, generated with `A` $= 0.7$, `eta` $= 0.10$ and `rho` $= 0.10$.

Figure 4.29: Comparison of trapped guiding centre dynamics coming from different modalities in modular Poincaré sections, on the left results from guiding centre mode while on the right results from full orbit mode.

Fig. [4.17], made by the union of data coming from 16384 different simulations, required 20 days to be generated, this while running on 50 CPU cores. If one wants to obtain the same three figures, with the same discretization, within the same ρ and η ranges but through the use of full-orbit mode, the total simulation would take 200 days more or less, this would not be acceptable for our kind of code. A standard machine running non-stop for that amount of time start to commit errors other than the fact that the results probably are not worth the money needed to pay the energy required for the calculation. The target of every gyrokinetic code is to reproduce the results of a full dynamics code in the most accurate way while minimising the computational cost of the calculation.

4.8 Future perspectives

This work opens up further analysis of the concerning phenomena:

- Better study the influence of using Padé approximation instead of full Bessel functions when accounting for *FLR* effects.

- The effect of varying random phases of the potential can be tested, this can even lead to a massive change in the dynamics of the system but through the production of multiple percentage maps one can look for the global behaviour of the system, this should be done in order to generalize the results. The variation in the random phases can be performed simply by varying `numpy.random.seed`.

- It should be better investigated if the range on which input parameters have been varied makes sense if compared to the real ones, the ones associated with a plasma generated in a magnetic fusion device. What can be already said for the *Larmor radius* is that in our non-dimensional model, the macrostructures of the potential have a characteristic dimension in the order of the unit while the value of `rho` has always stayed in the order of 1/10, ten times lower. There are some regions in a Tokamak where the plasma has these characteristics but it should be better analysed. The same investigation should be performed for `A` and `eta`.

- It should be tried to use different turbulent potential maps, in particular, it should be interesting to adapt some maps generated from other gyrokinetics codes in order to investigate the dynamics of test particles in a more realistic case.

- The present model could be applied to a more realistic geometry, a curved toroidal as well as a non-uniform magnetic field can be considered. This would lead to different governing equations, with a greater number of independent variables (you don't have the reduction of order 2 as in the present case). The particles would no longer only be subject to $\mathbf{E} \times \mathbf{B}$ drift but also to the $\nabla\mathbf{B}$ or the curvature drift, as in more realistic codes. The complexity of the code would massively increase but, at the same time, the results would be better compared with the ones of other gyrokinetics codes. The steps of the procedure to obtain the equations would be the same, additional dependencies would appear in case of a more complex $\mathbf{B}$ naturally, leading to more difficult and complex expressions.

Chapter 5

Conclusions

The present work exploits guiding centre theory to study particle dynamics inside turbulent plasmas. The pre-existing 2D simplified gyrokinetic model has been checked and has been further developed, and the expression of the gyrokinetic Hamiltonian has been expanded up to the third order as reported in Sec. 2.7.3. A comparable model for the study of particles' full orbits has been corrected, and the fixed non-dimensional version of the equations has been obtained and implemented, as shown in Sec. 2.9 while implementing also a method to obtain guiding centre trajectories from the ones of the corresponding ions (Sec. 4.1). Now it is possible to compare the dynamics coming from a full-kinetic approach with the one resulting from the gyrokinetic model up to the second order. The Hamiltonian is an invariant of the system, it is important to study the trend of the latter over time for each particle, in order to check that the computation is done in the correct way. The expression of the equation for the time evolution of k (additional independent variable introduced with the autonomization of the system) has been derived for both the approaches (guiding centre in Sec. 2.13.1 and full kinetic in Sec. 2.13.2) and its non-dimensional version has been implemented in order to be integrated together with the governing equations for particles dynamics. The expression of Hamiltonian (total energy of a particle) has been obtained for both the reduced (in Sec. 2.14.1) and the full system (in Sec. 2.14.2) and its non-dimensional expression has been implemented in order to be able to plot its trend over time and check if it remains constant. Since it has been verified that the magnetic moment (independent variable of the system) becomes a parameter (and it remains constant, as reported in Sec. 2.7.2) in the gyrokinetic framework, its expression has been obtained through the application of Lie transform, as described in Sec. 2.15, so that it is possible to compute its trend up to the second order from full orbits data and check that it is well conserved. The numerical Python code is available at `github.com/cchandre/Guiding-Center/` and it has been heavily modified during the internship, all the above-mentioned equations other than the new full kinetic approach have been implemented together with the method to obtain guiding centres dynamics from the ions one, this required a massive restructuring of the existing code. One of the main features of the new version of the code is the use of class `Trajectory`, described in Sec. 3.4.2, it has been inserted to distinguish various types of particles (trapped, sub-diffusive and super-diffusive) using geometrical criteria as well as to compute interesting quantities for these three categories right after having obtained the full trajectories in a very efficient way. As regards the numerical analysis of results, five main areas have been investigated:

- **Check on the correct operation of `fo2gc` method:** As shown in Sec. 4.1, the implemented `fo2gc` method (described in Sec. 3.3.5) works properly and it seems to accurately estimate the guiding centre orbits since the dynamics showed in Fig. [4.1] is pretty similar to the expected one, it can be concluded that the function is implemented correctly and that it can be used to compare the dynamics from the two different modes.

- **Time-step analysis for both full kinetic and gyrokinetic approach:** The time-step analysis has been performed thanks to the exploiting of total energy computation, the same simulation executed with the use of different values of `TimeStep` parameter revealed different trends of the Hamiltonian as shown in Fig. [4.5] and Fig. [4.7] respectively from Sec. 4.2.2 and Sec. 4.2.3. In case of an excessively high value of the time-step then the Hamiltonian is not well conserved, two different optimal values have been chosen for the two modalities:

  - $5 \cdot 10^{-3}$ for the gyrokinetic model.
  - $5 \cdot 10^{-4}$ for full kinetic model.

  It is also very important to have verified that, since the guiding centres' dynamics is much smoother with respect to particles' one, the required time-step for the integration can be much larger (1 order of magnitude) emphasizing the great advantage of a gyrokinetic approach with respect to a full kinetic one. Within this analysis, as reported in Sec. 4.2.4, the influence of the number of points on the grid, where the potential is evaluated, on the conservation of Hamiltonian has been also investigated, since the computational time is not affected by this parameter (Fig. [4.14]), and following the results showed in Fig. [4.13] when a large amount of memory is available the use of $\mathtt{N} = 2^{12}$ must be preferred, this lead to a total number of points on the grid equal to 2^{24}.

- **Influence of physical parameters ρ and η on guiding centres dynamics:** The influence of `rho` (*Larmor radius*) and `eta` (*real parameter*) is principally investigated in Sec. 4.3.1 and Sec. 4.3.2, it has not been investigated the influence of `A` (Amplitude of electrostatic field oscillations) since it was already been done in previous work on this model [Sta22]. As shown in Fig. [4.17], the choice of input parameters strongly influences the behaviour of the system. The expected regularisation of the dynamics associated with an increase of ρ is clearly present while looking at the percentage map for trapped particles, this is fundamental since it clearly reflects the real physics of a system like this. On the other hand, Fig. [4.18] highlights the presence of a strong super-diffusive regime for $\mathtt{rho} \geq 0.2$. What is more, is that this particular phenomenon occurs in the same region where the percentage of trapped is at its maximum, the novelty here is that together with a well-known effect we have also the occurrence of an unidentified behaviour with exact opposite dynamics with respect to trapped particles one.

- **Investigation of anomalous ballistic behaviour in certain regimes:** Sec. 4.4, 4.5 and 4.6 have been dedicated to the analysis of the phenomenon discovered during the analysis at the previous point. Through the use of local analysis as well as local rotational number analysis, it has been confirmed that the anomalous ballistic behaviour is associated with *meandering tori* mechanism. The presence of a transport barrier, which prevents transport along the perpendicular direction with respect to the trajectory of invariant particles, together with the presence of a twist-less invariant torus, a trajectory for which the computed winding number presents a local maximum, are the final confirmation of the presence of the well known *meandering tori* mechanism. The reason that leads to the appearance of this phenomenon has not been investigated, probably lies in the turbulent potential, the latter has only been clearly identified.

- **Comparison of dynamics of the two different kinetic approaches:** The comparison of results from the different modalities of the code is faced in Sec. 4.7. Despite the obvious fact that the behaviour of guiding centres from full orbits is more chaotic since the potential is turbulent and the particles follow large orbits. The dynamics and the Poincaré sections for the two modalities are in good accordance, it is possible to recognise the two preferential paths that invariant guiding centres follow and it is noteworthy that some particles have a comparable speed with the ones from the gyrokinetic model. This means that the revealed and examined ballistic behaviour is not artificially generated by the considered gyrokinetic model but it is present also in a more reliable approach as the one that simply exploits the integration of the equations of motion. It is noticeable also the fact that also the dynamics of trapped particles is well comparable, this is very important as well.

In conclusion, the present work has ranged over various aspects, from the further development of the theoretical gyrokinetic model to the derivation and implementation of new methods and tools in the numerical code to the numerical analysis of results through the use of non-conventional local analysis. A good level of novelty comes from my work on this topic, the investigation of an anomalous super-diffusive behaviour has led to the discovery of the presence of a particular mechanism that hadn't been previously observed in models like the considered one, this while also verifying that the results from the gyrokinetic framework were reliable thanks to the comparison with a full kinetic approach. This work surely opens up future investigations and research as anticipated in Sec. 4.8, it can be exploited to better understand transport associated with the turbulence in magnetic fusion devices.

Appendix A

Hamiltonian systems

This appendix is based on [Cha18].

A.1 Hamilton's equations

The variational principle (also known as Hamilton's principle) applies to the system considered, this means that the system evolves in time minimizing the energy of the system, if functional energy is defined, through the minimization of this functional in time we would be able to know how the system evolves. This system is Hamiltonian indeed, this property is inherited by the Vlasov-Maxwell equation system, of which our system is a simplification. Despite the easing, the system maintains this fundamental property. If the system wasn't Hamiltonian the particles would have lost or increased their energy, the dynamical system wouldn't have been physically coherent. An example that can be mentioned is the validity of the Ljuville theorem, if this wasn't the case one of the most important theorems for the study of dynamical systems composed of particles would have lost its validity. Maintaining the characteristic of being a Hamiltonian system is crucial for us, our manipulation and operations on the model shouldn't drop this condition. A Hamiltonian system is characterised by the scalar function $H(\boldsymbol{x}, \boldsymbol{p}, t)$ also known as Hamiltonian. $\boldsymbol{q}$ and $\boldsymbol{p}$ describe the state of the system $\boldsymbol{r}$ with $\boldsymbol{r} = (\boldsymbol{q}, \boldsymbol{p})$. Of great interest are the evolution equation known as Hamilton's equations:

$$\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} = -\frac{\partial H}{\partial \boldsymbol{q}}. \tag{A.1}$$

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = +\frac{\partial H}{\partial \boldsymbol{p}}. \tag{A.2}$$

Hamilton's equations govern the time evolution of a dynamical system, an important binary operation in Hamilton mechanics is the Poisson bracket, the latter is characterised by useful multiple properties such as bilinearity, antisymmetry, Leibniz's rule and Jacoby identity. They are respectively summoned up in the following equations:

$$\Big\{F, G + \alpha H\Big\} = \Big\{F, G\Big\} + \alpha\Big\{F, H\Big\}. \tag{A.3}$$

$$\Big\{F, G\Big\} = -\Big\{G, F\Big\}. \tag{A.4}$$

$$\Big\{FG, H\Big\} = F\Big\{G, H\Big\} + \Big\{F, H\Big\}G. \tag{A.5}$$

$$\Big\{F, \Big\{G, H\Big\}\Big\} + \Big\{H, \Big\{F, G\Big\}\Big\} + \Big\{G, \Big\{H, F\Big\}\Big\} = 0. \tag{A.6}$$

In a more general sense, Poisson Bracket is used to define Poisson algebra.

A.2 Lie transform

Lie transform is a canonical transformation, a change of coordinates, very useful whenever you need to reduce the order of the system. Also in this case we do not lose the main characteristic of our system, being a Hamiltonian system. This coordinate change applies as follows:

$$Z' = e^{\varepsilon L_s} Z \iff Z = e^{-\varepsilon L_s} Z'. \tag{A.7}$$

With Z' and Z being respectively the new and old variables. The operator L_s is defined as follows:

$$L_s = \Big\{S, \cdot\Big\}$$

With S being the generating function associated with the contextual change of coordinates. Being L_s a Poisson bracket itself, it is a linear operator. If we take his exponential it will still be a linear operator, therefore by applying it we don't lose linearity. It is possible to write $e^{\varepsilon L_s}$ in a more interesting way:

$$e^{\varepsilon L_s} = \sum_{n=0}^{\infty} \frac{\varepsilon^n L_s^n}{n!} = 1 + \varepsilon L_s + \frac{\varepsilon^2 L_s^2}{2} + ...$$

With meaning of L_s^n being:

$$L_s^n = \Big\{S, \Big\{S, \Big\{S, ...\Big\{S, \cdot\Big\}\Big\}\Big\}\Big\}$$

What is more, is that for a generic observable F the following relation is valid:

$$F'(Z') = F(Z). \tag{A.8}$$

This means that the observable expressed with the new form through the new coordinates maintains the same value of the F expressed in the old shape with the old variables. This is really important, if this wasn't the case why would our Hamiltonian change his value consequently to a change of coordinates, it wouldn't be correct physically speaking. It is possible to show that Eq. [A.8] comes from the validity of Leibniz Rule:

$$F'(z') = F'(e^{\varepsilon L_s} z) = e^{\varepsilon L_s} F'(z)$$

In fact, through a few passages, it is easy to show that:

$$F'(z) = e^{-\varepsilon L_s} F(z) \quad \Rightarrow \quad F'(z') = e^{-\varepsilon L_s} F(z') = e^{-\varepsilon L_s} e^{\varepsilon L_s} F(z) = F(z)$$

Developing $e^{\varepsilon L_s}$ at the first-order means that:

$$F'(e^{\varepsilon L_s}z) = F'(z + \varepsilon\Big\{S, z\Big\}). \tag{A.9}$$

Using the property of bilinearity and the Jacoby identity it is possible to demonstrate the following equation:

$$\Big\{e^{\varepsilon L_s}F, e^{\varepsilon L_s}G\Big\} = e^{\varepsilon L_s}\Big\{F, G\Big\}. \tag{A.10}$$

Eq. [A.10 implies that the Poisson bracket are not being modified by the application of the operator $e^{\varepsilon L_s}$, developing the operator at the first-order, as it is done in Eq. [A.9]], it is possible to prove it:

$$\Big\{e^{\varepsilon L_s}F, e^{\varepsilon L_s}G\Big\} = \Big\{F + \varepsilon\Big\{S, F\Big\}, G + \varepsilon\Big\{S, G\Big\}\Big\} = \Big\{F, G\Big\} + \Big\{\varepsilon\Big\{S, F\Big\}, G\Big\} + \Big\{F, \varepsilon\Big\{S, G\Big\}\Big\} + O(\varepsilon^2)$$

It can be taken advantage of Jacoby's identity:

$$\Big\{F, \Big\{S, G\Big\}\Big\}(-)(-)\Big\{G, \Big\{F, S\Big\}\Big\} = -\Big\{S, \Big\{G, F\Big\}\Big\} = \Big\{S, \Big\{F, G\Big\}\Big\}$$

So that we obtain:

$$\Big\{e^{\varepsilon L_s}F, e^{\varepsilon L_s}G\Big\} = \Big\{F, G\Big\} + \varepsilon\Big\{S, \Big\{F, G\Big\} + O(\varepsilon^2)\Big\}. \tag{A.11}$$

It can be concluded that the Poisson brackets are not being modified from this operator, during the change of coordinates this is very important for example, in fact, the following expression is valid:

$$\Big\{F', G'\Big\}(z') = e^{\varepsilon L_s}\Big\{F, G\Big\}(z')$$

It should be noted that both brackets are calculated in the new variables, this is very useful, it is possible to write:

$$\Big\{F, G\Big\} = \frac{\partial F}{\partial z_i}\Big\{z_i, z_j\Big\}\frac{\partial G}{\partial z_j} = \frac{\partial F}{\partial z_i}J_{ij}(z)\frac{\partial G}{\partial z_j}$$

$$\Big\{F', G'\Big\} = \frac{\partial F'}{\partial z'_i}\Big\{z'_i, z'_j\Big\}\frac{\partial G'}{\partial z'_j} = \frac{\partial F'}{\partial z'_i}J_{ij}(z')\frac{\partial G'}{\partial z'_j}$$

As it is known the new expression of the variables can be calculated as:

$$z'_i = e^{\varepsilon L_s}z_i \ ; \quad z'_j = e^{\varepsilon L_s}z_j \ ;$$

If we substitute those two last definitions we have:

$$\Big\{z'_i, z'_j\Big\} = \Big\{e^{\varepsilon L_s}z_i, e^{\varepsilon L_s}z_j\Big\} = e^{\varepsilon L_s}\Big\{z_i, z_j\Big\} = e^{\varepsilon L_s}J_{ij}(z)$$

This last equation is fundamental, it is telling us that the expressions of the elements of the matrix for the change of coordinates do not change, of course, the value of the element will change, it is calculated in a new different variable, but the expression of the elements won't change consequently to the application of the Lie Transform. This is why one wants to use a canonical transformation, that's one of the most important advantages of doing that.

Bibliography

[Bal98] R. Balescu. Hamiltonian nontwist map for magnetic field lines with locally reversed shear in toroidal geometry. *Phys. Rev. E*, 58:3781–3792, Sep 1998.

[BH07] A. J. Brizard and T. S. Hahm. Foundations of nonlinear gyrokinetic theory. *Rev. Mod. Phys.*, 79:421–468, Apr 2007.

[Bis22] Breanna Bishop. National ignition facility achieves fusion ignition, 2022. [Online; Accessed: 13-Feb-2023.

[CB09] John R. Cary and Alain J. Brizard. Hamiltonian theory of guiding-center motion. *Rev. Mod. Phys.*, 81:693–738, May 2009.

[Cha18] Cristel Chandre. Hamiltonian systems of charged particles in electromagnetic fields: towards gyrokinetic theory. Technical report, Institut de Mathématiques de Marseille, 2018.

[Cha21] Cristel Chandre. Report on electrostatic guiding-center theory. Technical report, CNRS, Aix Marseille Univ, I2M, 13009 Marseille, France, 2021.

[com23] Scipy community. Scipy documentation, 2023. [Online; Accessed: 21-Feb-2023.

[dCNGM96] Diego del Castillo-Negrete, John Morton Greene, and Philip J. Morrison. Area preserving nontwist maps: periodic orbits and transition to chaos. *Physica D: Nonlinear Phenomena*, 91:1–23, 1996.

[Den90] Richard O Dendy. *Plasma dynamics*. Oxford University Press, 1990.

[dev23] Scipy developers. Numpy documentation, 2023. [Online; Accessed: 21-Feb-2023.

[EF23] 2023 EURO-FUSION. Euro-fusion. https://euro-fusion.org/, 2023.

[Fou23] Python Software Foundation. Python 3.11.2 documentation, 2023. [Online; Accessed: 21-Feb-2023.

[Fre08] Jeffrey P Freidberg. *Plasma physics and fusion energy*. Cambridge university press, 2008.

[GL21] Xavier Garbet and Maxime Lesur. Gyrokinetics. Technical report, CEA Cadarache: Saint Paul lez Durance, Provence-Alpes-Côte d'Azur, FR, 2021.

[Hel02] D.Sigmar Helander. *Collisional transport in Magnetized Plasmas*. Cambridge university press, 2002.

[HSK00] Archie A Harms, Klaus F Schoepf, and David Ross Kingdon. *Principles of fusion energy: an introduction to fusion energy for students of science and engineering*. World Scientific, 2000.

[IPP23] IPP. Max plank institute for plasma physics. https://www.ipp.mpg.de/en, 2023.

[Lit79] Robert G. Littlejohn. A guiding center hamiltonian: A new approach. *Journal of Mathematical Physics*, 20:2445–2458, 1979.

[Lit80] Robert Grayson Littlejohn. Hamiltonian theory of guiding center motion. Technical report, California University, Berkeley, 1980.

[Lit81] Robert G. Littlejohn. Hamiltonian formulation of guiding center motion. *Physics of Fluids*, 24:1730–1749, 1981.

[Lit83] Robert G. Littlejohn. Variational principles of guiding centre motion. *Journal of Plasma Physics*, 29:111 – 125, 1983.

[LPP22] LPP. Turbulence, 2022. [Online; Accessed: 16-Feb-2023.

[Mer16] Gabriele Merlo. *Flux-tube and global grid-based gyrokinetic simulations of plasma microturbulence and comparisons with experimental TCV measurements.* PhD thesis, EPFL, Lausanne, CH, June 2016.

[MW09] P. J Morrison and A. Wurm. Nontwist maps. *Scholarpedia*, 4(9):3551, 2009. revision #91590.

[OC95] Gisele A. Oda and IberêL. Caldas. Dimerized island chains in tokamaks. *Chaos, Solitons & Fractals*, 5(1):15–23, 1995.

[PVM$^+$88] Marco Pettini, Angelo Vulpiani, Jacques H Misguich, Michel De Leener, John Orban, and Radu Balescu. Chaotic diffusion across a magnetic field in a model of electrostatic turbulent plasma. *Physical Review A*, 38(1):344, 1988.

[SM20] E. Sander and J.D. Meiss. Birkhoff averages and rotational invariant circles for area-preserving maps. *Physica D: Nonlinear Phenomena*, 411:132569, 2020.

[Sta07] Weston M Stacey. *Nuclear Reactor Physics, Second Edition, Completely Revised and Enlarged.* John Wiley & Sons, 2007.

[Sta22] Matteo Stanzani. Gyrokinetic modelling of plasma transport and investigation on test particle dynamics. Master's thesis, University of Bologna, Bologna, BO, May 2022.

[TC18] Natalia Tronko and Cristel Chandre. Second-order nonlinear gyrokinetic theory: from the particle to the gyrocentre. *Journal of Plasma Physics*, 84(3), 2018.

[Wes04] Wesson. *Tokamaks, Third Edition.* Clarendon Press - Oxford, 2004.