

**Study of magnetic shaping effects on plasma flows
and micro-instabilities in tokamak plasmas
using a full-f gyrokinetic code
based on a real space field solver**

Kevin Obrejan

Abstract

The successful magnetic confinement of a fusion plasma requires a deep understanding of numerous types perturbations which span from the macroscopic scale of MagneticHydroDynamics (MHD) down to the microscopic scale of electrons and ions GyroKinetics (GK). Although both extremes have been thoroughly studied, the multi-scale interaction between them is a complex difficult problem both theoretically and numerically. The work presented in this PhD thesis introduces of a new simulation code making few reductive hypotheses both at the local level and at the global one. This electrostatic full-f 5D-Vlasov simulation code, with an innovative field solver in real space accurately treating Full Larmor Radius (FLR) effects, uses Cartesian coordinates in the poloidal section, allowing for arbitrary magnetic configurations – D-shaped or X-points – to be handled. The code is then exploited to study shaping effects on Landau damping of Geodesic Acoustic Modes (GAMs) and Ion Temperature Gradient (ITG) through 2 essential parameters: elongation and triangularity.

GAM damping is characterised by the residual Zonal Flow (ZF) level, the oscillation frequency and the damping rate, whose variation in non circular domains had already been partially studied using simple local GK theory and simulation codes. Our study included the development of an innovative method to solve the ZF equation using a diagonalisation of the differential operator to better take into account its intrinsic geometrical properties despite an unfitting numerical grid. After confirming already existing results around elongation, new results were introduced about the influence of triangularity, showing a strong influence on the damping rate, comparable to the one of elongation, which was furthermore quantified by an empirical law.

The code was later used to conduct a similar study on linear ITG, which is thought to play an important role in the anomalous heat transport observed after linear saturation has been reached. While shaping effects on heat diffusivities in the nonlinear phase had been studied numerically, no such study had been performed for ballooning structures using a full-f code. For this reason, the difference between δf and full-f was also studied, the latter exhibiting an interaction between toroidal modes even in the so-called linear phase, leading to significantly lower growth rates and V-shaped ballooning structures. Our simulations in non-circular domains showed that the width of ballooning structures and heat diffusivities hint at similar influences of shaping parameters: elongation stabilises confinement – reducing ballooning structures' width and heat diffusivities – while triangularities transitions from a slightly destabilising influence for near-circular configurations to no detectable influence for more elongated domains.

Acknowledgements

Firstly, I would like to express all my gratitude to my supervisor Prof. Yasuaki Kishimoto, Prof. Jiquan Li and Doc. Kenji Imadera for their continuous support over the years of my PhD. The path towards the completion of this PhD was long and full of uncertainties but their advice and support helped me find my way. Besides my advisors, I also thank the thesis committee Prof. Nakamura and Prof. Tanaka for their helpful remarks.

My sincere thanks also go to Paul Hilscher and Henrik Skibbe for their support and guidance both with life in Japan and in the world of research. I would also like to thank Ms. Nakamura and Ms. Takahashi as well as R.Matsui for helping me a lot making sense of Japan and understanding everyday life in this country.

More generally, I would like to thank all the members of Kishimoto laboratory who I had the opportunity to meet over the years. Although I was not the most social member of the lab, I really enjoyed the atmosphere and had many great conversations there.

Scientific contributions during this PhD

Published peer-reviewed papers

(author underlined)

- Obrejan, K., Imadera, K., Li, J., and Kishimoto, Y. (2017). Development of a new zonal flow equation solver by diagonalisation and its application in non-circular cross-section tokamak plasmas. *Computer Physics Communications*, 216:8–17.
- Obrejan, K., Imadera, K., Li, J.-Q., and Kishimoto, Y. (2015). Development of a global toroidal gyrokinetic vlasov code with new real space field solver. *Pasma Fusion Res.*, 10:42.

Presentations at academic conferences

(speaker underlined)

- Obrejan, K., Imadera, K., and Kishimoto, Y., "*Effects of plasma shaping on ITG instability in a full-f gyrokinetic simulation*", 日本物理学会 (JPS 2017), March, 2017. (*Oral*)
- Obrejan, K., Imadera, K., and Kishimoto, Y., "*Effect of plasma shaping on ITG/TEM instability in a full-f gyrokinetic simulation*", 22nd NEXT workshop, March, 2017. (*Oral*)
- Kishimoto, Y., Imadera, K., Liu, H., Wang, W., Obrejan, K. and Li, J. Q., "*Characteristics of turbulent transport in flux-driven toroidal plasmas*", 26th IAEA Fusion Energy Conference, October, 2016. (*Poster*)
- Obrejan, K., Imadera, K., Li, J. Q., and Kishimoto, Y., "*The effect of magnetic shaping on zonal flow in a toroidal global gyrokinetic simulation*", 21st NEXT workshop, March, 2016. (*Poster*)
- Obrejan, K., Imadera, K., Li, J. Q., and Kishimoto, Y., "*The effect of magnetic shaping on zonal flow and geodesic acoustic mode in a toroidal global gyrokinetic simulation*", 日本物理学会 (JPS 2015), September, 2015. (*Oral*)
- Imadera, K., Kishimoto, Y., Obrejan, K., Kobiki, T. and Li, J. Q., "*Non-local profile relaxation and barrier formation in toroidal flux-driven turbulence*", The 7th IAEA Technical Meeting on "Theory of Plasma Instabilities", March, 2015. (*Oral*)

- Obrejan, K., Imadera, K., Li, J. Q., and Kishimoto, Y., "*Development of a global gyrokinetic Vlasov code with new real space field solver*", 24-th International Toki Conference (ITC-24), November, 2014. (*Poster*)
- Imadera, K., Kishimoto, Y., Obrejan, K., Kobiki, T. and Li, J. Q., "*Global profile relaxation coupled with ExB staircase in toroidal flux-driven ITG turbulence*", 25th IAEA Fusion Energy Conference, October, 2014. (*Poster*)
- Obrejan, K., Imadera, K., Li, J. Q., and Kishimoto, Y., "*Development of a new real space Poisson solver for global toroidal gyrokinetic simulations*", Sokendai Asian Winter School (AWS2013), December, 2013. (*Poster*)
- Obrejan, K., Imadera, K., Hilscher, P., and Kishimoto, Y., "*Study of multi-scale turbulent transport in fusion plasmas using gyrokinetic simulations*", Mines ParisTech-Kyoto-Ajou University Joint Symposium on Sustainable Energy, March, 2013. (*Oral*)

Contents

Abstract	i
Acknowledgements	ii
Scientific contributions	iii
Contents	v
1 Introduction	1
1.1 Nuclear fusion as a source of energy	2
1.1.1 Humanity’s always increasing needs for energy	2
1.1.2 The energy of the Sun	3
1.2 Achieving nuclear fusion on Earth	4
1.2.1 Artificial fusion devices for magnetic confinement	4
1.2.2 The road ahead for tokamaks	7
1.3 Modelising and simulating magnetically confined fusion plasmas	9
1.3.1 Magnetohydrodynamics	9
1.3.2 Gyrokinetics	10
1.4 About this thesis	12
1.4.1 Objectives	12
1.4.2 Organisation of the chapters	13
2 The gyrokinetic formalism	15
2.1 Gyro-averaging of particle trajectories in a tokamak	17
2.1.1 Plasma confinement	17
2.1.2 The gyro-averaging procedure	20
2.2 Gyro-averaged dynamics	21
2.2.1 Hamiltonian and one-forms	22
2.2.2 Gyro-averaged dynamics of the guiding-centre	23
2.2.3 Gyro-centre coordinates	25
2.3 The gyrokinetic Vlasov equation	26

2.3.1	Full-f gyrokinetics	26
2.3.2	Splitting between equilibrium distribution and perturbation	29
2.4	Field equation	33
2.4.1	Gyrokinetic quasi-neutrality condition	33
2.4.2	Gyro-averaged terms in Fourier space	35
2.4.3	Adiabatic electrons	37
2.5	Simulating gyrokinetics	38
2.5.1	Simulation geometry and representation	38
2.5.2	Physics included in the simulation model	40
2.5.3	Overview of the existing simulation codes	41
2.6	Summary	42
3	Design of GKNET	43
3.1	The slab geometry code GKMUST	45
3.1.1	General design	45
3.1.2	Higher order finite differences	46
3.1.3	Time integration	48
3.2	From slab to toroidal geometry	49
3.2.1	GKMUST and the several GKNETs	49
3.2.2	Numerical grid and boundary conditions	51
3.2.3	Variable time step	52
3.3	Real space averaging method	55
3.3.1	Simple gyro-averaging	55
3.3.2	Double gyro-averaging	57
3.4	Field solver	60
3.4.1	Implementation of the gyro-averagings	60
3.4.2	Fixed-point method	62
3.5	Parallelisation	66
3.5.1	Cross-section shape in Cartesian coordinates	66
3.5.2	Sub-parallelisation μ and $v_{ }$ of the gyro-averagings	68
3.5.3	Allocation of available processes	68
3.6	Summary	69
4	Zonal Flow dynamics in D-shaped configurations	71
4.1	Simulating zonal flow dynamics in non circular domains	72
4.1.1	Constructing a flux function	73
4.1.2	Parametrisation of the flux surfaces	73
4.1.3	Zonal Flow solver by projection	75
4.2	Diagonalisation of the Zonal Flow equation	76
4.2.1	Eigenfunction equation	76
4.2.2	Eigenfunctions and eigenvalues	78

4.2.3	Correspondence to the Fourier description	79
4.3	Zonal Flow solver by diagonalisation	80
4.3.1	Diagonalisation based solver	80
4.3.2	Comparative performances	82
4.4	Shaping effects	82
4.4.1	Influence of elongation	83
4.4.2	Influence of triangularity	85
4.5	Summary	87
5	Full-f study of magnetic shaping effects on the Ion Temperature Gradient instability	89
5.1	The Ion Temperature Gradient Instability	90
5.1.1	Creation of ballooning structures	90
5.1.2	From linear ITG modes growth to the saturated phase	91
5.2	Difference between δf and full-f in circular configuration	92
5.3	Influence of elongation	93
5.3.1	Linear ITG growth rates	93
5.3.2	Ballooning structures	95
5.3.3	Combining results	95
5.4	Influence of triangularity	97
5.5	Summary	98
6	Conclusion	99
6.1	Summary of the results	100
6.2	The meaning of this all	101
6.3	To go beyond	102
	Bibliography	103

Chapter 1

Introduction

Contents

1.1	Nuclear fusion as a source of energy	2
1.1.1	Humanity's always increasing needs for energy	2
1.1.2	The energy of the Sun	3
1.2	Achieving nuclear fusion on Earth	4
1.2.1	Artificial fusion devices for magnetic confinement	4
	Magnetic confinement fusion	5
	Inertial confinement fusion	7
1.2.2	The road ahead for tokamaks	7
	Measuring a tokamak's capacities	7
	The projects ITER and DEMO	8
1.3	Modelising and simulating magnetically confined fusion plasmas	9
1.3.1	Magnetohydrodynamics	9
1.3.2	Gyrokinetics	10
1.4	About this thesis	12
1.4.1	Objectives	12
1.4.2	Organisation of the chapters	13

1.1 Nuclear fusion as a source of energy

1.1.1 Humanity's always increasing needs for energy

In December 2015, representatives of 196 countries gathered near Paris for the 2015 United Nations Climate Change Conference (also referred to as COP 21 or CMP 11) in hope of reaching an agreement on the reduction of their greenhouse gases emissions in the hope of curbing global warming and more generally, climate change. This agreement came as a natural next step to the Kyoto Protocol, established in 1997, one crucial difference: while most countries soon gave up on their Kyoto Protocol engagement or simply refused to even sign it, the Paris Agreement was signed, to date, by 148 parties, including major greenhouse gases emitters such as China and India, and most of the signatories seem genuinely committed to the goals agreed upon. Despite the late decision of the USA – world's largest emitter per inhabitant and second overall – to leave the agreement, several of its States and largest cities have expressed a strong desire to enforce the terms of the Paris Agreement at their local level.

For most countries, the effort to curb green house gases emissions in the next few years will consist in replacing their fossil fuel powered electrical plants by renewable energy sources – for the most part solar panels and wind turbines – and/or nuclear fission reactors. However, renewable energies suffer from their irregular electricity production whose compensation is still an issue, and the not so negligible emissions generated by the production, installations and maintenance of the very large amount of small units needed to meet a large country's demand; as an example, after 20 years of diminution, Germany's CO₂ emissions have reached a plateau since 2010, making their goals for 2020 seem less and less reachable [Wettengel, 2017]. The problems related to nuclear fission are even more well-known and include long-term radioactive waste management, risks of reactor meltdown or more generally public acceptance and the increasing exploitation cost as safety standards become more and more strict, in particular since the tragic 2010 Fukushima incident. For those reasons, neither options could, on their own or combined together, be a definitive long term solution to Earth's problems.

Nuclear fusion promises a green, safe and virtually unlimited source of energy fuelled by light and abundant elements such as hydrogen without releasing greenhouse gases nor generating long-term radioactive waste. In addition to solving the current greenhouse gases emission issues our modern society are faced with, it could open a new era of energy technologies with applications to other areas such as space exploration. Nuclear fusion is the very physical phenomenon which makes the stars shine but unlike burning fossil fuels or our current nuclear fission powered plants, fusion cannot be achieved simply in easy test conditions and later scaled up. Each experimental step is complex and requires a lot of money, time and effort. Harnessing

nuclear fusion represents one of the greatest scientific and technological challenges of humankind.

1.1.2 The energy of the Sun

Our sun, like most small stars of similar mass, consists of a large majority of hydrogen (about 73.5% of its mass) and helium (25%), kept together by gravity. Each layer of gas supports the weight of the layer above it, making pressure and temperature increase rapidly and atoms are soon stripped from their electrons, turning the gas into a plasma where the positively charged ions and negatively charged electrons flow separately as 2 electromagnetically interacting fluids. At the core, the crushing pressure and extreme temperatures are so strong that atom nuclei, in their fast erratic movements, can overcome the electric repulsion of their positive charges and come close enough for their components to interact: nuclear fusion happens. Hydrogen nuclei collide and fuse into the heavier helium nuclei, releasing in the process humongous amounts of energy in the form of photons, which, after thousands of years of travel through the Sun's various layers, eventually reach the outermost layers of the Sun and escape it.

The main reaction taking place in the 14 million Kelvin of our Sun's core is the so-called **proton-proton chain** in which 4 protons are consumed and a helium-4 nucleus is produced (see Fig. 1.1). The overall reaction can be summarised in 3 steps.

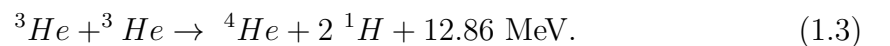
The first step is the combination of the collision of 2 protons 1H , forming a diproton followed by the β^+ decay of one of the protons, yielding deuterium 2D , a positron and an electronic neutrino:



The quick annihilation of the positron with an electron also releases two 511 keV γ -photons although this part of the reaction is not taken into account above. The β^+ decay of the proton plays an important role in the frequency at which this first step can successfully occur, the diproton is much more likely to split back into 2 protons given that the half-life of a proton in solar-core conditions is close to a billion years. However, once this initial step has occurred, nuclear fusion is quickly obtained when it collides with a proton, forming a helium-3 nucleus:



From this point, 4 reactions are possible, the most likely one in our Sun being:



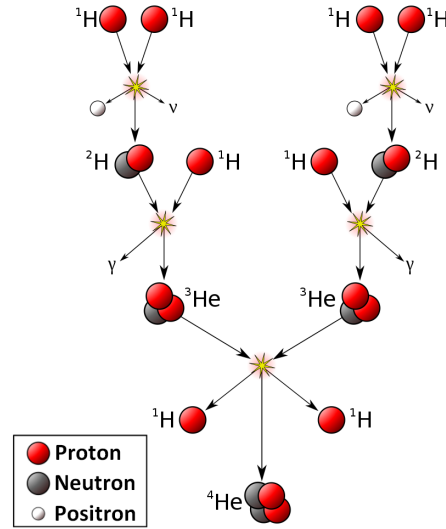


Figure 1.1: The proton-proton chain reaction P-P I, which dominates in our Sun, turning 4 protons into a helium-4 nucleus and releasing energy. [Image credit: Adapted from [Wikimedia Commons](#).]

While the nuclear fusion reaction in any star starts with hydrogen, turning it into helium and producing energy, when this primary fuel runs out, the stars either contract into even hotter and denser states where the fusion of heavier elements is possible, or they die out in various manners. Although the nuclear fusion reaction only releases more energy than it consumes when the by-products are lighter than iron-56 (${}^{56}_{26}\text{Fe}$) or nickel-62 (${}^{62}_{28}\text{Ni}$), heavier elements have been produced through nuclear fusion during extreme astrophysical events such as supernovae. Indeed, all the elements found on Earth today were one day produced that way in the dying moments of long-gone stars.

1.2 Achieving nuclear fusion on Earth

1.2.1 Artificial fusion devices for magnetic confinement

The first man-made nuclear fusion reactions were achieved in the early 1930s, as a way to understand stars. Soon, the shadows of World War II lead to military tests as part of the Manhattan project, until the first hydrogen bomb was successfully detonated in the early 1950s. It is also from that period that research into controlled nuclear fusion as a source of energy really started. Since then, scientific and engineering research have achieved tremendous progress.

Several approaches involving various isotopes of hydrogen and other light ele-

ments have been considered with in mind the objective of producing as much energy as possible within attainable pressure and temperature conditions. Two main general ideas are currently under intense research: **magnetic confinement** attempts to create and maintain of a self-sustaining reaction in conditions similar to the ones at the core of the Sun, while **inertial confinement** aims at repeatedly initiating small fusion reactions by hitting pellets of fuel with high intensity lasers to create a brief explosive reaction. In both cases, the current main candidate reaction would be the fusion of deuterium and tritium into helium-4, which yields the highest energy gain in the most accessible conditions.

Magnetic confinement fusion

Magnetic confinement fusion is currently the most advanced approach, with over 200 experimental devices build to date. The fuel of the fusion reaction is confined within a toroidal chamber using powerful supra-conductive magnets, taking advantage of the electromagnetic properties of plasmas. The stability of such a confinement is capital since no solid material could possibly enter in direct contact with plasmas heated to temperatures of the order of 150 million Celsius degrees. Early in the development of such devices, it was understood that the curvature of the magnetic fields creates forces in opposite directions on the ions and electrons, causing their accumulation respectively at the bottom and top of the chamber, destroying the neutrality of the plasma and therefore its confinement. To address this issue, 2 main solutions were found, which allows to split fusion reactor designs into 2 categories: **tokamaks** and **stellarators**.

In tokamaks, an electrical current is driven through the plasma, in toroidal direction, therefore generating an additional poloidal magnetic field which winds the particles' trajectories around the concentric magnetic surfaces (for more detailed explanation about the magnetic configuration of tokamaks, see Sec. 2.1.1). They are the most well-developed design and were the first to reach **break-even**, a situation in which the energy produced by the fusion reaction equals the one provided to maintain the plasma in a steady state. Figure 1.2 shows a picture and design drawing of the tokamak JET (Joint European Torus) which is currently the largest device in operation, until the completion of its successor: ITER (International Thermonuclear Experimental Reactor).

Stellarators instead prevent the charge accumulation problem by twisting the charged particles' path along the toroidal direction. Although this design is one of the oldest, the complexity of the modelisation of such magnetic fields lead to difficulties and the abandon of the concept for a few decades until the 1990s. The more complicated magnetic field which confines the plasma in such a manner can be generated using various approaches, which we will not list here, but one of which

1. INTRODUCTION

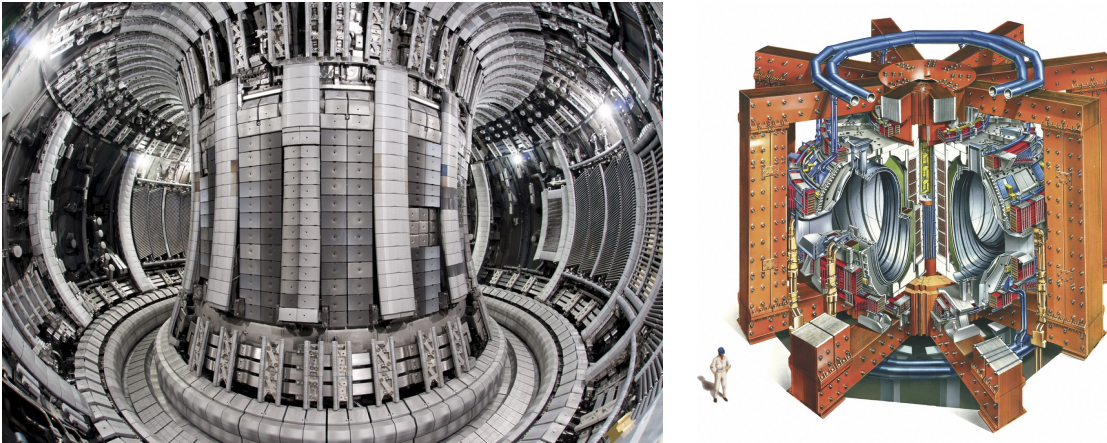


Figure 1.2: Picture (left) and original design drawings (right) of the tokamak JET (Joint European Torus), operated by the Culham Centre for Fusion Energy near Oxford (UK) since 1984. [Image credit: [EUROfusion](#) (Right: *Oldie but goldie. 1980's drawing of the JET machine*) under [Creative Commons 4.0](#)]

can be seen on Fig. 1.3 on the schema of the coil systems used in the stellarator Wendelstein 7-X.

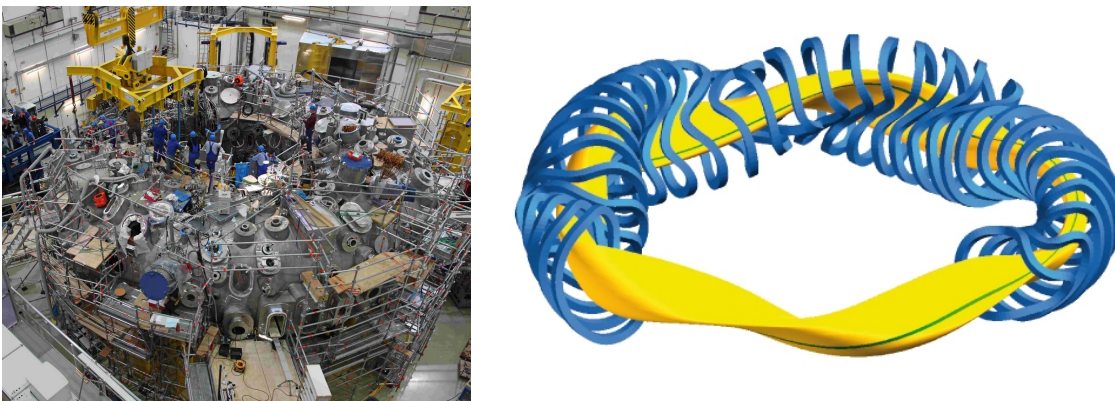


Figure 1.3: Picture (left) and schema of the coil system (right) of the stellarator device Wendelstein 7-X built in the Max Planck Institute of Plasma Physics (IPP) of Greifswald, Germany. With a major radius of 5.5m, this device is, to date, the largest stellarator device in operation. [Image credit: Max-Planck Institute für Plasmaphysik through [Wikimedia Commons](#).]

Inertial confinement fusion

Rather than maintaining a fusion reaction for a long period of time, inertial confinement relies on the small fusion reactions performed at a very high rate. The general idea is to blast high-energy lasers at a fuel target to trigger the explosion of its outer layer, thus compressing the core into the extreme pressure and temperatures required to ignite a fusion reaction strong enough to propagate back to the exploded outer layer. Despite the apparent roughness of the method, it requires extreme accuracies on the shape of the pellet to preserve the symmetry of the implosion. Figure 1.4 shows a representation of the design of the hohlraum (reaction chamber) of the NIF (National Ignition Facility).

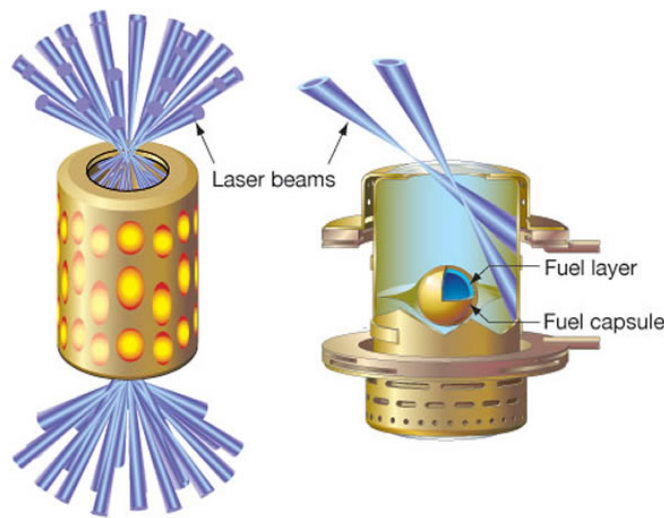


Figure 1.4: Schematic representation of the reaction chamber used in the NIF located at the Lawrence Livermore National Laboratory, in the USA. [Image credit: [Lawrence Livermore National Laboratory](#).]

1.2.2 The road ahead for tokamaks

Measuring a tokamak's capacities

The improvement of magnetic fusion devices can be measured using the fusion energy gain factor Q , which is the ratio of the power produced by the fusion reaction over the power required to maintain the plasma in a steady state. The case $Q = 1$ is called break-even (and was already mentioned above) and represents an important milestone for any fusion device, although much higher values are required for a self-sustaining reaction. Indeed, the energy produce essentially takes 2 forms: α

particles (helium-4 nuclei) which remain in the core plasma after being produced, and the light and fast neutrons, which escape confinement and are absorbed in various manners by the wall (or used to breed tritium from lithium blanket in the case of the fast neutrons). While the latter form of energy is not lost but converted into heat which in turn is used to generate electricity, it represents for the plasma a loss of heat which needs to be compensated for. However, the conversion of thermal energy into electricity is only about 40% efficient and running a reactor requires electricity beyond the simple heating of the plasma. When taking into account all those factors, it is usually estimated that $Q > 20$ would be required for a fusion reactor to be exploitable for energy generation.

Using theoretical estimates for the fusion power produced and the proportion of energy kept in confinement or evacuated, another useful figure of merit can be derived. The **Lawson criterion** or its later extension into the **fusion triple product** $n_i \tau_E T$, loosely connects the ion density n_i , energy confinement time τ_E and plasma temperature T required for a given gain factor to be reached. The 3 terms in the product are often used to compare operation conditions for fusion reactors using different parameters, as shown on Fig. 1.5, along with some major achievements.

The projects ITER and DEMO

The next great milestone in the history of nuclear fusion is the completion of the next largest fusion device, ITER (International Thermonuclear Experimental Reactor), currently under construction in the South of France. The project is the result of the funding and intense research effort (both in fusion science and in essential components of the reactor) of more than 35 countries: the European Union, India, Japan, China, Russia, South Korea and the United States. This huge reactor would host a vacuum chamber more than 12 meters wide (twice the current record) and confine plasmas at temperatures beyond 150 million degrees for more than 16 minutes.

The project aims at setting several new records by generating 500 MW of fusion power (compared to the current record of 16 MW held by JET) with a gain factor of 10. It is also expected to demonstrate the feasibility of an industrial fusion power plant as well as test various promising ideas such as the interception of fast neutrons by blankets of lithium, thus producing tritium directly inside the vacuum chamber.

Despite the gigantism of the ITER project, it is not the end of the road and its direct successor, DEMO, has already been planned thoroughly. More than just a "simple" experimental reactor, DEMO will be the first prototype of a commercially exploitable nuclear fusion power plant. It is intended to be superior to ITER in all regards and expected to produce at least 2 GW of fusion power on a continuous basis, and with a gain factor of at least $Q = 25$.

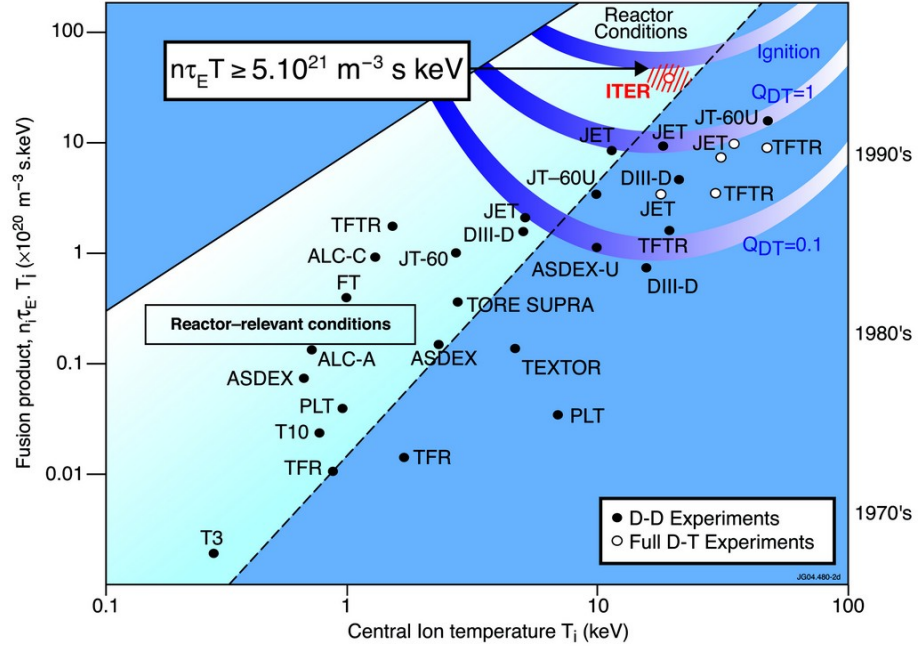


Figure 1.5: Fusion triple product achieved with various magnetic confinement fusion devices. The progress towards the top right part of the graph, corresponding to higher fusion gain, is also matched by the year when the shown result was obtained, as indicated on the right axis. [Image credit: [EUROfusion](#), under [Creative Commons 4.0](#)]

1.3 Modelising and simulating magnetically confined fusion plasmas

1.3.1 Magnetohydrodynamics

Invented in the early 1940s by Hannes Alfvén, Magnetohydrodynamics (MHD) is, in a way, the most intuitive modelisation of a plasma. As its name suggests, it combines derivatives of the Navier-Stokes equations to represent its fluid nature and of Maxwell's equations to take into account its electromagnetic properties. This addition of models leads to a very large number of unknowns, with, for instance, in the case of the single fluid model, a total of 14: the mass density ρ , the fluid velocity $\mathbf{V}$, the current $\mathbf{J}$, the pressure P , the electric field $\mathbf{E}$ and the magnetic field $\mathbf{B}$. All those unknown are govern by a massive set of equations:

- the continuity equation $\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0$,
- the momentum equation $\rho \frac{\partial \mathbf{V}}{\partial t} = \mathbf{J} \times \mathbf{B} - \nabla P$,

- Ohm's law $\mathbf{E} + \mathbf{V} \times \mathbf{B} = \eta \mathbf{J}$,
- the closure equation $\frac{d}{dt} (P \rho^{-\gamma}) = 0$,
- the Maxwell-Faraday equation $\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$,
- the Maxwell-Ampère equation $\nabla \times \mathbf{B} = \mu_0 \mathbf{J}$,
- the Maxwell equation for the conservation of magnetic flux $\nabla \cdot \mathbf{B} = 0$,

with the 2 additional parameters: the resistivity η and the ratio of specific heats γ . Particular values for those 2 parameters can lead to common approximations such as ideal MHD ($\eta = 0$) as opposed to resistive MHD ($\eta > 0$), incompressibility ($\gamma = 0$) or isothermal heat fluxes ($\gamma = 1$). Another important parameter of MHD, the β parameter, is the ratio of the plasma (fluid) pressure P over the magnetic pressure $B^2/(2\mu_0)$, and is an important characteristic for fusion reactor design as β reflects the quality of the confinement for the strength of electromagnets used.

Despite the apparent heaviness of the equation system, many simplifications can be used to focus on various aspect of plasma configurations and dynamics. In particular, MHD is very efficient to determine global equilibria and their properties. In addition to the MHD simulation codes, many codes are dedicated to the computation of MHD equilibrium configurations, such as EFIT [Lao et al., 1985] or MEUDAS [Takeda and Tokuda, 1991]. Such configurations are structured around concentric surfaces of plasma, called **magnetic flux surfaces**, over which temperature and density are constant. Charged particles can move freely, while the movement across flux surfaces only results from collisions and only occur with a low frequency. Those collisions and cross-surface movement being what allows heat to be transported, MHD predicts high level of confinements, much higher than what is observed in reality.

1.3.2 Gyrokinetics

Historically, the very first theories developed to modelise plasmas were kinetic theories and the core equation of gyrokinetics, the **Vlasov equation**, was first formulated in 1938. However, their first successful application to magnetically confined plasmas came decades later, in [Taylor and Hastie, 1968]¹, in an effort to better capture the dynamics at the scale of the Larmor radius. It has been one of the key elements in the foundation of **neo-classical transport**, which exhibits much more complex particle trajectories than what was expected until then. The radial extension of those banana-shaped trajectories (see Fig. 1.6) allows heat to be transported much faster from the core plasma to the boundary compared to MHD's predictions.

¹Although the term "gyrokinetics" will only be introduced later in the following decade.

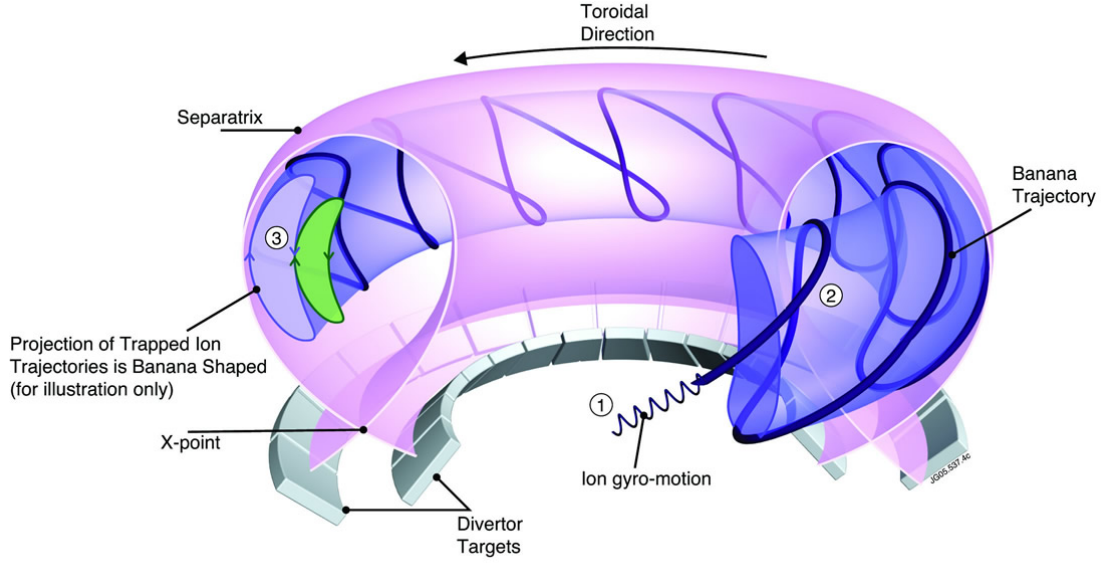


Figure 1.6: Decomposition of the trajectory of particles in a magnetically confined plasmas, as they are described by gyrokinetics. The helicoidal trajectory of particles around magnetic field lines curve at the macroscopic scale into banana-shaped orbits. [Image credit: [EUROfusion](#), under [Creative Commons 4.0](#)]

Formally, instead of the mass distribution, velocity and electric current found in MHD, gyrokinetics describes the plasma's state through distribution function – one per species – over a 6-dimensional phase space. Its formalism first considers the fast gyration of charged particles – ions and electrons – around magnetic field lines before the so-called gyro-ordering is used to separate the fast microscopic scales from the slower macroscopic ones before averaging some of that dynamics while retaining as much information as possible. This gyro-averaging effectively reduces the dimensionality of the phase space from 6 to 5 dimensions, greatly simplifying the problem, in particular from a computational point of view. To better take advantage of the gyro-ordering, gyrokinetics usually considers a smooth static equilibrium on top of which a small perturbation is added and its evolution studied.

Modern gyrokinetics focuses on "anomalous" transport, which leads to the greater heat losses observed in experimental reactors, beyond the current predictions of the neo-classical theory. Such anomalous transport is believed to originate from microscopic instabilities such as the Ion or Electron Temperature Gradient (respectively ITG and ETG) instabilities or Trapped Electron Modes (TEM) [Weiland, 1999]. Figure 1.7 illustrates the various scales involved in gyrokinetics and modern fusion plasma physics. The arrow connecting macroscopic magnetic shaping effects

1. INTRODUCTION

to anomalous transport with the label "influence?" does not reflect the universally accepted expectation but is the question that motivates the the present thesis. Chapter 2 introduces the basics of gyrokinetics further, as well as some variations which can be used to either simplify calculations or instead include more of the physics modelisation.

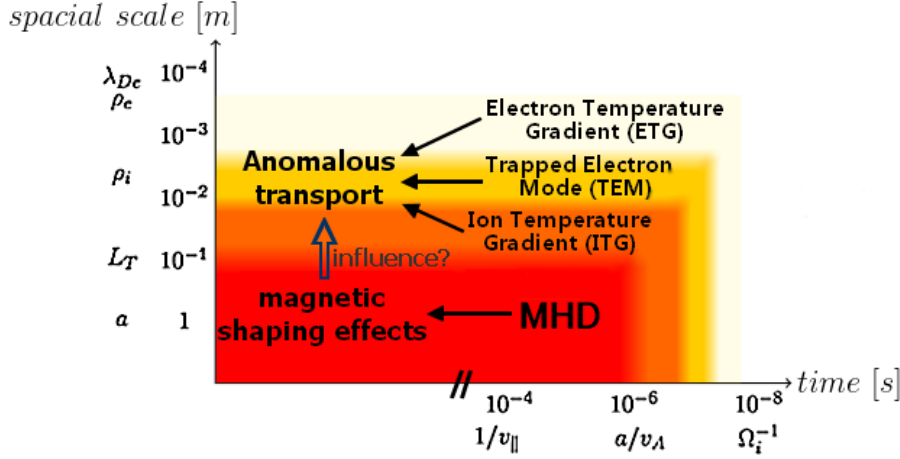


Figure 1.7: Various time and space scales considered in gyrokinetics.

1.4 About this thesis

1.4.1 Objectives

Anomalous transport is a very active field of research, both from the theoretical and numerical point of view. While there is a general consensus that its mechanism is related to microscopic instabilities such as ITG, ETG or TEM, the exact mechanism is still unclear. The present thesis aims at studying numerically the multi-scale interaction between the macroscopic magnetic field shaping and the microscopic dynamics of the ITG instability.

For this purpose, several steps will be necessary, starting with the development of a new simulation code. It will have to be full-f to properly capture the interaction between macro and micro-scales and in general geometry so that the influence of the deformation of its magnetic boundary can be studied. Once such a tool ready, its results concerning the dynamics of ZF and the influence of magnetic shaping on collisionless ZF damping will have to be verified and studied as it is an essential component of some aspects of the ITG instability. Only then will the study of the ITG instability itself begin.

1.4.2 Organisation of the chapters

The general outline of the study and its chapters is given below.

In chapter 2, we introduce the core ideas which characterise gyrokinetics, namely the gyro-centre coordinate system and how it allows a very separation of time and space scales, leading to the so-called gyro-averaging procedure. We then present how this approach can be used to derive the fundamental equation system used by GKNET, which consists of the gyrokinetic Vlasov equation and the associated electrostatic field equation: the quasi-neutrality condition. Various strategies used in simulating plasma dynamics using gyrokinetics are also presented, along with an overview of the major simulation codes already in use across the world.

In chapter 3, we present the key elements behind the design of the simulation code GKNET, which was entirely developed during this PhD. Its basic structure was derived from the much simpler slab geometry Vlasov code GKMUST, from which GKNET inherited some of its core features: the Runge-Kutta 4 time integration scheme, the 5-points stencil finite differences, the Gauss-Legendre quadrature in one of the 5 dimensions *et cetera*. This base was then extended to a full toroidal geometry and equipped with an innovative field solver in real space, allowing for an accurate treatment of Full Larmor Radius (FLR) effects – *i.e.* without Padé approximation –, while the use of Cartesian coordinates for the poloidal section allows for arbitrary magnetic configurations. Its implementation also makes use of MPI parallelisation along 4 of the 5 dimensions and other advanced techniques for parts of the code where the dimensionality is reduced.

In Chapter 4, we focus on the so-called Landau damping of the Zonal Flow (ZF), which is the magnetic flux surface averaged component of the electrostatic field. For this purpose, we introduce and parametrise the geometry of D-shaped magnetic configurations which will be used in this chapter and the next one. This parametrisation is then exploited with the development and implementation of an innovative mathematical method to solve the so-called ZF equation using a diagonalisation of its differential operator to better take into account its intrinsic geometric properties. This new tool is used to study the influence of magnetic elongation and triangularity on ZF damping, first confirming existing results stating that higher elongations cause a faster damping of ZF towards a higher residual, before introducing new results concerning triangularity which can have stronger effects on some parameters than is usually assumed. This influence is furthermore quantified with the introduction a new empirical law.

In chapter 5, such a study of magnetic shaping effects is continued with the ITG instability, along with an investigation of the differences between results from δf and full-f simulations. Full-f simulations are in particular found to yield much smaller growth rates and V-shaped ballooning mode structures, unlike their δf counterparts which are straight. In elongated domains, both growth rates and mode width are

1. INTRODUCTION

found to decrease, showing a stabilising influence of elongation as it would reduce heat diffusivity. The influence of triangularity is much weaker and transition from slightly destabilising at low elongations having no real impact for more elongated cases.

Chapter 2

The gyrokinetic formalism

Contents

2.1	Gyro-averaging of particle trajectories in a tokamak	17
2.1.1	Plasma confinement	17
	Magnetic configuration	17
	Particle's trajectories	18
2.1.2	The gyro-averaging procedure	20
	Particle gyromotion	20
	Gyro-ordering	21
2.2	Gyro-averaged dynamics	21
2.2.1	Hamiltonian and one-forms	22
	Hamiltonian of a single charged particle	22
	One-form representation	22
2.2.2	Gyro-averaged dynamics of the guiding-centre	23
2.2.3	Gyro-centre coordinates	25
	First order perturbation	25
	Electrostatic gyrokinetics	25
2.3	The gyrokinetic Vlasov equation	26
2.3.1	Full-f gyrokinetics	26
	Equations of motion	26
	Expression of the main coefficients	27
	Final expression of the Vlasov equation	28
2.3.2	Splitting between equilibrium distribution and perturbation . .	29

2. THE GYROKINETIC FORMALISM

	The δf equation	30
	Canonical equilibrium	31
2.4	Field equation	33
2.4.1	Gyrokinetic quasi-neutrality condition	33
2.4.2	Gyro-averaged terms in Fourier space	35
2.4.3	Adiabatic electrons	37
	Electron adiabatic response	37
	The Zonal Flow equation	37
2.5	Simulating gyrokinetics	38
2.5.1	Simulation geometry and representation	38
	Slab, toroidal and flux-tube geometries	38
	Lagrangian and Eulerian representations	38
2.5.2	Physics included in the simulation model	40
	Local and global simulations	40
	Ion and electron species	40
	Electromagnetic codes	41
2.5.3	Overview of the existing simulation codes	41
2.6	Summary	42

2.1 Gyro-averaging of particle trajectories in a tokamak

2.1.1 Plasma confinement

Magnetic configuration

Tokamaks rely on 3 types of coils to generate and shape the electromagnetic field that confines the plasma inside the toroidally shaped vacuum vessel. Figure 2.1 gives a general idea of the disposition of the various sets of coils around the plasma. The general shape of the magnetic field is generated by numerous **toroidal field coils** wrapped around the torus (light blue). The exact shape of the cross-section –usually a D– is constrained by large circular (outer) **vertical field coils** (grey) which are positioned horizontally around the whole torus. Finally, the **central ohmic transformer** (green) stands in the hole of the torus; when traversed by a variable electric current, a toroidal flow of charged particles is created in the plasma, in turn generating a poloidal magnetic field.

The total magnetic field which confines the charged particles is a combination of those toroidal and poloidal field lines. Mathematically, the magnetic field $\mathbf{B}$ is described under the form:

$$\mathbf{B} = \nabla\psi \times \nabla\varphi + F(\psi)\nabla\varphi \quad (2.1)$$

with $\nabla\varphi = \frac{1}{R}\mathbf{e}_\varphi$ the gradient of the toroidal angle, ψ the poloidal **flux function** and the function $F = RB_\varphi$ is a degree of freedom which is often chosen constant at B_0R_0 . The reduction of the ideal MHD equations to the cross-section leads to the **Grad-Shafranov equation**, which allows to calculate ψ for the chosen F . The **flux surfaces** on which ψ is constant play a major role as global MHD parameters such as the temperature or the pressure can be expressed as a function of ψ .

Despite the simplicity of figure 2.1, modern tokamaks generate complex magnetic configurations where not all magnetic surfaces are closed. While the innermost flux surfaces intersect the cross-section in the shape of a D, the outermost taper at the bottom before opening up again. The flux-surface at the transition between these 2 tendencies crosses itself at the so-called **X-point**. This technique separates the tightly confined core plasma where the nuclear fusion reaction occurs from the scrape-off layer, which drives impurities coming from the wall into the divertor at the bottom of the chamber, on the other side of the X-point. Since our study focuses on the core plasma, the flux-surfaces will be assumed to be closed and smoothly shaped, ignoring the outermost regions.

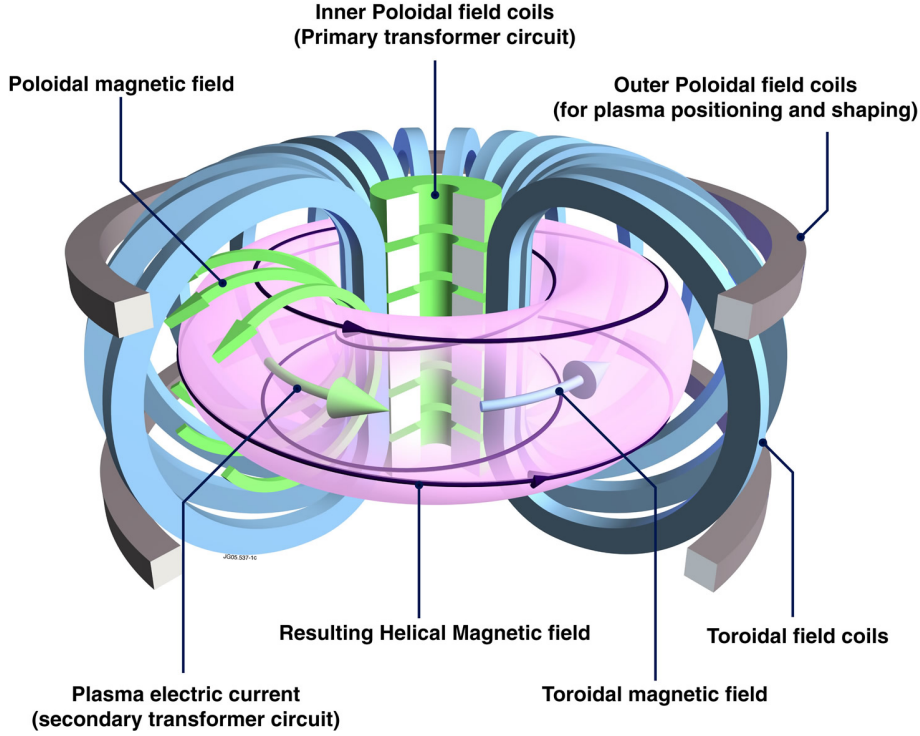


Figure 2.1: The toroidal confinement chamber contains 3 types of coils which generate and shape the magnetic field: the toroidal field coils (light blue) and the vertical field (grey) to create the toroidal magnetic field, and the central ohmic transformer (green) which drives an electric current through the plasma to generate the poloidal field and finalise the confinement. [Image credit: [EUROFusion](#).]

Particle's trajectories

The close gyromotion of the charged particles around the magnetic field lines will be the focus of the next section and shall be neglected for the moment to only describe the larger scale trajectories. The particles are thus trapped on the magnetic flux surface and their direction is given by the magnetic field vector $\mathbf{B}$ (although they may travel along it or in the opposite direction depending on the sign of their charge).

The indirect generation of the poloidal field by induction of a toroidal current requires a very large variation of current inside the central solenoid. This reduces the relative strength of the poloidal component of the magnetic field compared to the toroidal one ($\frac{B_p}{B_\phi} \sim \varepsilon \ll 1$), making the particle's trajectories over the flux surfaces very weakly coiled. This "windedness" is measured using the **safety factor** q which

is formally defined as

$$q(\psi) = \frac{1}{S_\psi} \oint \frac{\mathbf{B} \cdot \nabla \varphi}{\mathbf{B} \cdot \nabla \theta} dS = \frac{F(\psi)}{S_\psi} \oint \frac{r dS}{R |\nabla \psi|} \quad (2.2)$$

where the integral is performed over the corresponding flux surface and $S_\psi = \oint dS$.

The safety factor corresponds to the kind of the magnetic field and, in the case of rational values $q = N/M$, a particle will make N toroidal and M poloidal rotations before reverting to its initial position as illustrated on figure 2.2. The safety factor increases with the radius, from about 1 up to several units (3 in the case of ITER), until confinement is lost. When a window of interest is defined, the safety factor at mid-radius q_r and the shearing rate at mid-radius $\hat{s}_r = \frac{r}{q} \frac{dq}{dr}$ are often specified as parameters rather than the full q -profile as they alone characterise a large part of the physics.

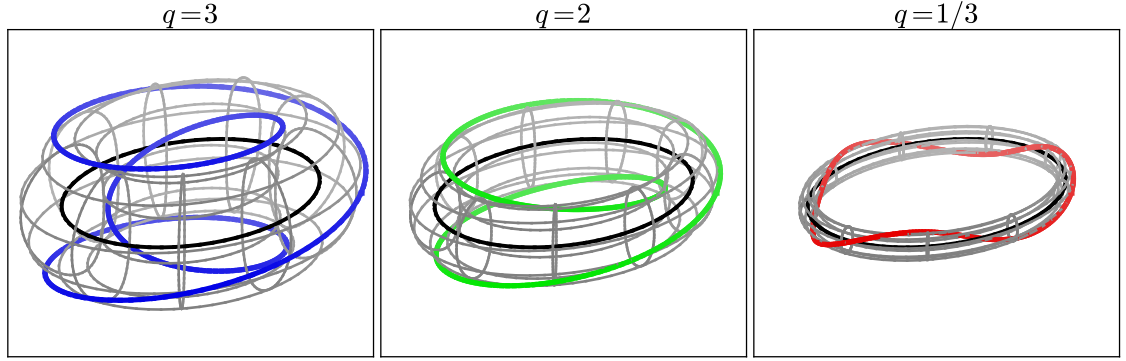


Figure 2.2: The safety factor q characterises how "winded" the trajectories and field lines are around the plasma. It corresponds for a closed trajectory to the ratio of number of poloidal turns over the number of toroidal turns before returning to the starting point. The safety factor increases with the radius until confinement is lost. The right figure, for $q = 1/3$ is shown only for illustration purposes as the minimal value near the centre is usually 1.

A simple configuration with circular cross-section and a magnetic field of the form

$$\mathbf{B} = \frac{B_0 R_0}{R} \begin{pmatrix} 0 \\ \frac{r}{q R_0} \\ 1 \end{pmatrix} \quad (2.3)$$

is often considered for its simplicity. It corresponds to the flux function $\psi = -B_0 \int \frac{r'}{q(r')} dr'$ and q is generally taken of the form $q(r) = q_b + q_a r^2$ although any choice of q -profile would be compatible with this definition of the magnetic field.

2.1.2 The gyro-averaging procedure

Particle gyromotion

Elementary electromagnetism teaches that that, in the presence of a magnetic field, charged particles will follow a helicoidal trajectory coiled around the field lines, in a so-called gyromotion, while also drifting along them (see left of Fig. 2.3). Denoting $\mathbf{b} = \mathbf{B}/B$ the direction of the magnetic field line, $v_{\parallel} = \mathbf{v} \cdot \mathbf{b}$ the velocity of the parallel drift and $\mathbf{v}_{\perp} = \mathbf{b} - v_{\parallel} \mathbf{b}$ the velocity of the gyromotion, the gyro-centre is located at $\mathbf{R} = \mathbf{x} - \boldsymbol{\rho}$ where $\boldsymbol{\rho} = \mathbf{v}_{\perp} \times \mathbf{b} / \Omega_s$ is the gyroradius and with $\Omega_s = e_s B / m_s$ for the species "s" considered (see right of Fig. 2.3).

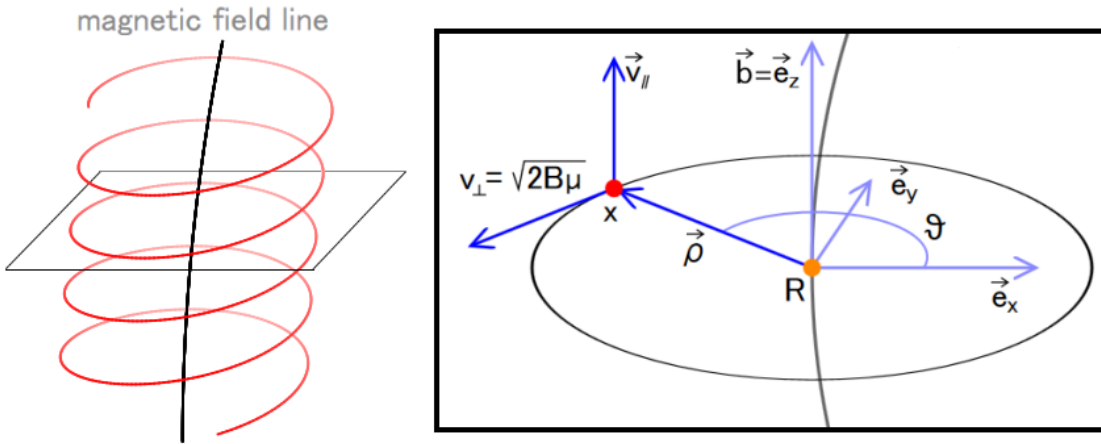


Figure 2.3: Left: helicoidal trajectory of a particle around a magnetic field line. Right: decomposition of this gyromotion as a parallel translation along the magnetic field line with a speed $v_{\parallel}$ and a gyration in the perpendicular plane at speed $v_{\perp}$.

Gyrokinetics is based first and foremost on its signature change of coordinates from particle coordinates $(\mathbf{x}, \mathbf{v})$ in phase space into the sextuplet $(\mathbf{R}, \mu, v_{\parallel}, \vartheta)$ where $\mu = \frac{mv_{\perp}^2}{2B}$ is the magnetic momentum and ϑ is the gyro-phase *i.e.* the angle corresponding to the gyro-motion. The second key idea behind gyrokinetics is based on a gyro-ordering, which separates time and space scales between the fast gyromotion and other slower motions. These slower motions include not only the parallel drift, but also even slower but also more complex drifts caused by the curvature of magnetic field lines as is the case for plasmas confined magnetically in toroidal fusion devices.

The gyro-ordering allows to average the equations over the gyro-phase, approximating the fastest dynamics by a simpler but accurate enough averaged treatment.

The mathematics of this gyro-ordering are given below while most of the chapter will explain how the equations are affected by this procedure.

Gyro-ordering

In addition to the separation of time scale mentioned above, the gyro-ordering considers the small scales effects of the gyro-motion on the confining magnetic field or the local losses of charge neutrality at the scale of the Debye length, but also the smallness of the perturbation δf added on top of the initial equilibrium F_0 . These scale separations can be expressed as [Brizard and Hahm, 2007]:

- $\frac{\partial/\partial t}{\Omega} \sim \varepsilon_\omega$ for the time-scale separation,
- $\rho_i \frac{\nabla B}{B} \sim \frac{\rho}{a_0} \sim \varepsilon_B$ for the space-scale separation (a_0 being the characteristic size of the device),
- $\frac{\delta f}{F_0} \sim \varepsilon_\delta$ for the smallness separation,

where the various ε are all assumed small compared to 1. Their relative smallness is of no importance and a common infinitesimal ε with no index is often used, as will be the case in what follows.

In practice, the exact value of those ε varies depending on species and between places within the plasma but the ordering is always acceptable. Simple calculations using standard tokamak parameters can show that the gyro-radius is of the order of a few millimetres for ions (and more than 40 times smaller for electrons), which, compared to the several-meters wide fusion reactor, yields $\varepsilon_B \sim 10^{-3}$. Similarly, as the parallel drift and gyromotion both occur at comparable speeds, around the thermal velocity, the ratio of time-scales is quite close to the ratio of space-scales $\varepsilon_\omega \sim \varepsilon_B \sim 10^{-3}$.

2.2 Gyro-averaged dynamics

The derivation of the gyro-averaged equations is a difficult task which has been refined with the years, using more and more formal tools. We present here a derivation based on a one-form of the Hamiltonian, inspired from derivations found in [Brizard and Hahm, 2007] or [Szepesi, 2012], themselves based on the much older [Cary and Littlejohn, 1983].

2.2.1 Hamiltonian and one-forms

Hamiltonian of a single charged particle

In classical Hamiltonian mechanics, the movement of a charged particle is characterised by its Lagrangian

$$\mathcal{L} = \frac{1}{2}m|\dot{\mathbf{x}}|^2 + e\mathbf{A} \cdot \dot{\mathbf{x}} - e\Phi, \quad (2.4)$$

where e is the particle's charge, m is its mass, Φ is an electrostatic potential and $\mathbf{A}$ a magnetic vector potential. Instead of conserving a symplectic representation of phase space $(\mathbf{x}, \mathbf{v})$, the introduction of the canonical conjugate momentum $\mathbf{p} = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{x}}} = m\dot{\mathbf{x}} + e\mathbf{A}$ allows to calculate the Hamiltonian of the particle as:

$$\mathcal{H} = \mathbf{p} \cdot \dot{\mathbf{x}} - \mathcal{L} = \frac{1}{2}mv^2 + e\Phi \quad (2.5)$$

In turn, this formalism allows to express concise equations of motions as a set of 6 first-order differential equations (in 3 dimensional space) given by

$$\dot{\mathbf{x}} = \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \quad \dot{\mathbf{p}} = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}} \quad (2.6)$$

Despite the very distinct methodology, Hamiltonian mechanics are merely a reformulation of classical Newtonian (and Lagrangian) mechanics. Although the equations it yields are usually not easier to solve at first, Hamiltonians are part of a extensive theoretical framework which gives access to advanced theoretical results and deeper properties of physical systems.

One-form representation

To simplify the description of more advanced transformations on the system's coordinates, a one-form can be used. For this purpose, the system's coordinates are gathered in a single variable $z = (z^t, z^{\mathbf{x}}, z^{\mathbf{v}}) = (t, \mathbf{x}, \mathbf{v})$ and the one-form γ is defined as:

$$\gamma = \mathbf{p} \cdot d\mathbf{x} - \mathcal{H}dt = (m\mathbf{v} + e\mathbf{A}(\mathbf{x})) \cdot d\mathbf{x} - \left(\frac{1}{2}mv^2 + e\Phi(\mathbf{x}) \right) dt, \quad (2.7)$$

whose component can alternatively be denoted as

$$\gamma_{\mathbf{R}} = m\mathbf{v} + e\mathbf{A}(\mathbf{x}) \quad \gamma_{\mathbf{v}} = 0 \quad \gamma_t = -\mathcal{H}. \quad (2.8)$$

In this formalism, the equations of motion are recovered by using the Euler-Lagrange equation:

$$\left(\frac{\partial \gamma_{\nu}}{\partial z^{\eta}} - \frac{\partial \gamma_{\eta}}{\partial z^{\nu}} \right) \frac{\partial z^{\nu}}{\partial t} = 0 \quad (2.9)$$

where Einstein notations have been used (implicit summations on ν), as will be the case from now on. In one-form notation, the product $\gamma_\nu dz^\nu$ is conserved in all coordinates systems and the one-form Γ associated with a different set of coordinates Z verifies:

$$\gamma_\nu dz^\nu = \Gamma_\nu dZ^\nu \quad \Leftrightarrow \quad \Gamma_\nu = \gamma_\eta \frac{\partial dz^\eta}{\partial dZ^\nu} \quad (2.10)$$

Our next step will be to determine the one-form governing the guiding-centre coordinates, ignoring the short time scale of the gyromotion. The small-scale perturbation will then be re-introduced in order to revert to the dynamics of the particle in gyro-centre coordinates.

2.2.2 Gyro-averaged dynamics of the guiding-centre

To perform the one-form change of variable, the relations in Eq. (2.10) first have to be inverted. For this purpose, let us first introduce the unit vector that points from the gyro-centre to the particle $\mathbf{e}_\rho(\vartheta) = \boldsymbol{\rho}/\rho$ its derivative, along the direction of the gyromotion $\mathbf{e}_\perp(\vartheta) = \mathbf{v}_\perp/v_\perp = \frac{\partial \mathbf{e}_\rho}{\partial \vartheta}$. The inverse change of coordinates then reads:

$$\mathbf{x} = \mathbf{R} + \boldsymbol{\rho}(\mathbf{R}) \quad \mathbf{v} = v_\parallel \mathbf{b}(\mathbf{x}) + \mathbf{v}_\perp \mathbf{e}_\perp, \quad (2.11)$$

It is important to highlight that ρ (and subsequently the magnetic field B contained in it) is evaluated at the guiding centre $\mathbf{R}$ while the vector $\mathbf{b}$ which guides $v_\parallel$ is evaluated at $\mathbf{x}$. In what follows, evaluation at $\mathbf{R}$ will be omitted.

As time is not transformed, the time component is not modified, only re-formulated using Z instead of z :

$$\Gamma_t = \gamma_t = -\frac{1}{2}mv_\parallel^2 - \mu B(\mathbf{x}) - e\Phi(\mathbf{x}) \quad (2.12)$$

To calculate the gyro-averaging of Γ_t or any other quantity, we simply integrate over the gyro-angle: $\langle \cdot \rangle = \frac{1}{2\pi} \oint \cdot d\vartheta$. An expansion of the quantities taken at $\mathbf{x}$ allows to obtain their gyro-average up to the first order in ε :

$$\langle \mathbf{A}(\mathbf{x}) \rangle = \langle \mathbf{A}(\mathbf{R}) + \rho (\nabla_{\mathbf{R}}) \mathbf{A} \cdot \mathbf{e}_\rho + O(\varepsilon^2) \rangle = \mathbf{A}(\mathbf{R}) + O(\varepsilon^2). \quad (2.13)$$

Applying our gyro-averaging operator on Γ_t thus yields:

$$\langle \Gamma_t \rangle = -\frac{1}{2}mv_\parallel^2 - \mu B - e\Phi + O(\varepsilon^2) \quad (2.14)$$

To continue the change of coordinate and gyro-averaging, we apply Eq. (2.10), keeping in mind that $\gamma_v = 0$:

$$\begin{aligned}
 \Gamma_{\mathbf{R}} &= \gamma_{\mathbf{x}} \frac{\partial z^{\mathbf{x}}}{\partial \mathbf{R}} \\
 &= (m\mathbf{v} + e\mathbf{A}(\mathbf{x})) \cdot \frac{\partial(\mathbf{R} + \rho\mathbf{e}_\rho)}{\partial \mathbf{R}} \\
 &= (m\mathbf{v} + e\mathbf{A}(\mathbf{x})) \cdot (\mathbf{1} - \mathbf{e}_\rho \otimes (\nabla_{\mathbf{R}}) \rho) \\
 &= m\mathbf{v} + e\mathbf{A}(\mathbf{x}) - e(\mathbf{A}(\mathbf{x}) \cdot \mathbf{e}_\rho) (\nabla_{\mathbf{R}}) \rho
 \end{aligned} \tag{2.15}$$

as $\mathbf{v} \perp \mathbf{e}_\rho$ (and where $\mathbf{1}$ is the identity matrix and $\otimes$ is the tensorial product). To apply the gyro-averaging operator, we simply need to notice that $\langle \mathbf{A}(\mathbf{x}) \cdot \mathbf{e}_\rho \rangle = 0 + O(\varepsilon)$ to obtain:

$$\langle \Gamma_{\mathbf{R}} \rangle = mv_{\parallel} \mathbf{b} + e\mathbf{A} + O(\varepsilon^2). \tag{2.16}$$

The μ component reads

$$\Gamma_{\mu} = \gamma_{\mathbf{x}} \frac{\partial z^{\mathbf{x}}}{\partial \mu} = (m\mathbf{v} + e\mathbf{A}(\mathbf{x})) \cdot \left(0 + \frac{\partial \rho}{\partial \mu} \mathbf{e}_\rho \right) = e \frac{\partial \rho}{\partial \mu} \mathbf{A}(\mathbf{x}) \cdot \mathbf{e}_\rho \tag{2.17}$$

which, given $\frac{\partial \rho}{\partial \mu} = -\rho/(2\mu)$, makes this only term of second order, therefore:

$$\langle \Gamma_{\mu} \rangle = O(\varepsilon^2) \tag{2.18}$$

Similarly, the $v_{\parallel}$ component cancels out before even being averaged:

$$\langle \Gamma_{v_{\parallel}} \rangle = \left\langle \gamma_{\mathbf{x}} \frac{\partial z^{\mathbf{x}}}{\partial v_{\parallel}} \right\rangle = \langle 0 \rangle = 0. \tag{2.19}$$

Finally, the gyro-angle component reads:

$$\Gamma_{\vartheta} = \gamma_{\mathbf{x}} \frac{\partial z^{\mathbf{x}}}{\partial \vartheta} = (m\mathbf{v} + e\mathbf{A}(\mathbf{x})) \cdot (0 + \rho \mathbf{e}_{\perp}) = m\rho v_{\perp} + e\rho \mathbf{A}(\mathbf{x}) \cdot \mathbf{e}_{\perp}, \tag{2.20}$$

from which the gyro-averaging only leaves one term below the second order:

$$\langle \Gamma_{\vartheta} \rangle = \frac{2\mu B}{\Omega} + O(\varepsilon^2) \tag{2.21}$$

Gathering the results from Eqs. (2.14), (2.16), (2.18), (2.19) and (2.21), the one-form for the guiding centre's coordinates is found to be:

$$\langle \Gamma \rangle = (mv_{\parallel} \mathbf{b} + q\mathbf{A}) \cdot d\mathbf{R} + \frac{2\mu B}{\Omega} d\vartheta - \left(\frac{1}{2} m v_{\parallel}^2 - \mu B - e\Phi \right) dt \tag{2.22}$$

The equation of motion for the guiding-centre could be derived from this result using the Euler-Lagrange equation Eq. (2.9). However, our objective is not the guiding-centre alone as it completely ignores the gyromotion. A one-form which takes it into account under the form of perturbation is required.

2.2.3 Gyro-centre coordinates

First order perturbation

The fluctuations at the scale of the gyro-radius will now be re-introduced in the gyromotion dynamics up to the first order in ρ . To distinguish the additional effects from the ones calculated above, the general method would be to add small-scale perturbations to the electrostatic potential Φ and the magnetic vector potential $\mathbf{A}$:

$$\Phi = \Phi_0 + \varepsilon \Phi_1 \quad \mathbf{A} = \mathbf{A}_0 + \varepsilon \mathbf{A}_1. \quad (2.23)$$

Using these decompositions, the initial one-form in particle coordinates (Eq. (2.7)) can also be split into $\gamma = \gamma_0 + \varepsilon \gamma_1$, where:

$$\gamma_0 = (m\mathbf{v} + e\mathbf{A}_0(\mathbf{x})) \cdot d\mathbf{x} - \left(\frac{1}{2}mv^2 + e\Phi_0(\mathbf{x}) \right) dt, \quad \gamma_1 = e\mathbf{A}_1(\mathbf{x}) \cdot d\mathbf{x} - e\Phi_1(\mathbf{x})dt. \quad (2.24)$$

The transformation of the lowest order term γ_0 of course remains exactly the same as previously calculated. However, in the general case, when attempting to determine the guiding centre one-form up to first order $\tilde{\Gamma} = \tilde{\Gamma}_0 + \varepsilon \tilde{\Gamma}_1$, the expansions used to translate quantities from $\mathbf{x}$ to $\mathbf{R}$ are more difficult to manipulate as some products of expanded terms may retain terms that were previously cancelled. This issue leads to the need for the Lie transform perturbation theory and quite complicated calculations as can be found in the original work that introduced it [Cary and Littlejohn, 1983], in a more modern form [Brizard and Hahm, 2007] or more in details in [Szepesi, 2012].

Electrostatic gyrokinetics

Within the scope of electrostatic dynamics, the magnetic field being external and fixed, it does not vary on small scales leaving $\mathbf{A}_1 = 0$. This assumption greatly simplifies the situation and allows to rewrite Eq. (2.24) as:

$$\gamma_0 = (m\mathbf{v} + e\mathbf{A}_0) \cdot d\mathbf{x} - \left(\frac{1}{2}mv^2 + e\Phi_0(\mathbf{x}) \right) dt \quad \gamma_1 = -e\Phi_1(\mathbf{x})dt. \quad (2.25)$$

Not only is this new form simpler, it removed the perturbed part of $\mathbf{A}$ whose expansion was no longer sufficient. With these simplifications, the gyro-averaging process can thus be performed on $\tilde{\Gamma}$, mostly without any change. However, while in the guiding-centre case $\langle \Phi_0(\mathbf{x}) \rangle = \Phi_0(\mathbf{R}) + O(\varepsilon^2)$, the gyro-averaging of the perturbed component of the electric potential $\langle \Phi_1(\mathbf{x}) \rangle$ cannot be simplified any further, yielding the perturbed gyro-averaged one-form:

$$\langle \tilde{\Gamma} \rangle = (mv_{\parallel} \mathbf{b} + q\mathbf{A}_0) \cdot d\mathbf{R} + \frac{2\mu B}{\Omega} d\vartheta - \left(\frac{1}{2}mv_{\parallel}^2 + \mu B + e\langle \Phi \rangle_{\mathbf{x}} \right) dt \quad (2.26)$$

where both components of Φ have been recombined and the notation $\langle \Phi \rangle_{\mathbf{x}}$ bears $\mathbf{x}$ as an index to specify the type of gyro-averaging: from particle coordinates to gyro-centre coordinates. The inverse operation (from gyro-centre coordinates to particle coordinates) will be introduced later and denoted $\langle \Phi \rangle_{\mathbf{R}}$.

2.3 The gyrokinetic Vlasov equation

An important property of Hamiltonian systems is the conservation of phase space measure: the volume of a region of phase space remains constant with time. An equivalent way of stating this property is Liouville's theorem, which states the conservation the distribution function f (any function of it) along nonlinear characteristics *i.e.* that its total derivative is zero:

$$\frac{Df(z)}{Dt} \equiv \frac{\partial f}{\partial t} + \dot{\mathbf{x}} \cdot \frac{\partial f}{\partial \mathbf{x}} + \dot{\mathbf{v}} \cdot \frac{\partial f}{\partial \mathbf{v}} = 0. \quad (2.27)$$

In plasma physics, this equation is known as the Vlasov equation (or collisionless Boltzman equation) and governs the time evolution of distribution of a species (ions, electrons *etc*) within a plasma. The gyrokinetic Vlasov equation, based on the gyro-centre coordinates that we just derived is stated as:

$$\frac{\partial f}{\partial t} + \dot{\mathbf{R}} \cdot \frac{\partial f}{\partial \mathbf{R}} + \dot{v}_{\parallel} \frac{\partial f}{\partial v_{\parallel}} + \dot{\mu} \frac{\partial f}{\partial \mu} = 0 \quad (2.28)$$

where the distribution function f is a function of time, the guiding centre coordinate $\mathbf{R}$, the parallel velocity $v_{\parallel}$ and the magnetic moment μ .

2.3.1 Full-f gyrokinetics

Equations of motion

The time derivatives of the coordinates can be obtained by solving the Euler-Lagrange equation Eq. (2.9), applied to the one-form Eq. (2.26):

$$\left(\frac{\partial}{\partial Z^{\eta}} \langle \tilde{\Gamma} \rangle_{\nu} - \frac{\partial}{\partial Z^{\nu}} \langle \tilde{\Gamma} \rangle_{\eta} \right) \frac{\partial Z^{\nu}}{\partial t} = 0. \quad (2.29)$$

Starting with 2 simplest components:

$$\eta = v_{\parallel}: m \mathbf{b} \cdot \dot{\mathbf{R}} - m v_{\parallel} = 0 \quad \Rightarrow \quad \mathbf{b} \cdot \dot{\mathbf{R}} = v_{\parallel} \quad (2.30a)$$

$$\eta = \mu: -\frac{2m}{q} \dot{\mu} = 0 \quad \Rightarrow \quad \dot{\mu} = 0 \quad (2.30b)$$

Before continuing with the component $\eta = \mathbf{R}$, we extract its term of the sum for $\nu \in \mathbf{R}$ which is a bit complicated to factorise and look at its individual components:

$$\eta = \mathbf{R}^1: \left(\frac{\partial \tilde{\Gamma}_2}{\partial Z^1} - \frac{\partial \tilde{\Gamma}_1}{\partial Z^2} \right) \frac{\partial Z^2}{\partial t} + \left(\frac{\partial \tilde{\Gamma}_3}{\partial Z^1} - \frac{\partial \tilde{\Gamma}_1}{\partial Z^3} \right) \frac{\partial Z^3}{\partial t} = \left[\dot{\mathbf{R}} \times (\nabla \times \tilde{\Gamma}_{\mathbf{R}}) \right]^{\mathbf{R}^1} \quad (2.31a)$$

$$\eta = \mathbf{R}^2: \left(\frac{\partial \tilde{\Gamma}_1}{\partial Z^2} - \frac{\partial \tilde{\Gamma}_2}{\partial Z^1} \right) \frac{\partial Z^2}{\partial t} + \left(\frac{\partial \tilde{\Gamma}_3}{\partial Z^2} - \frac{\partial \tilde{\Gamma}_2}{\partial Z^3} \right) \frac{\partial Z^3}{\partial t} = \left[\dot{\mathbf{R}} \times (\nabla \times \tilde{\Gamma}_{\mathbf{R}}) \right]^{\mathbf{R}^2} \quad (2.31b)$$

$$\eta = \mathbf{R}^3: \left(\frac{\partial \tilde{\Gamma}_1}{\partial Z^3} - \frac{\partial \tilde{\Gamma}_3}{\partial Z^1} \right) \frac{\partial Z^2}{\partial t} + \left(\frac{\partial \tilde{\Gamma}_2}{\partial Z^3} - \frac{\partial \tilde{\Gamma}_3}{\partial Z^2} \right) \frac{\partial Z^2}{\partial t} = \left[\dot{\mathbf{R}} \times (\nabla \times \tilde{\Gamma}_{\mathbf{R}}) \right]^{\mathbf{R}^3} \quad (2.31c)$$

Defining

$$\mathbf{B}^* \equiv \frac{1}{e} \nabla \times \tilde{\Gamma}^{\mathbf{R}} = \mathbf{B} + \frac{mv_{\parallel}}{e} \nabla \times \mathbf{b}, \quad (2.32)$$

the equation for $\eta = \mathbf{R}$ (in Eq. (2.29)) then reads:

$$e \dot{\mathbf{R}} \times \mathbf{B}^* = m \dot{v}_{\parallel} \mathbf{b} + \mu \nabla B + e \nabla \langle \Phi \rangle_{\mathbf{x}}. \quad (2.33)$$

This equation is not simple to exploit as it is, but it contains 2 time derivatives that we still need to express.

Expression of the main coefficients

Taking the cross section of this relation with $\mathbf{b}$:

$$\begin{aligned} e \left(\dot{\mathbf{R}} \times \mathbf{B}^* \right) \times \mathbf{b} &= (\mu \nabla B + e \nabla \langle \Phi \rangle_{\mathbf{x}}) \times \mathbf{b} \\ \Leftrightarrow e \underbrace{\dot{\mathbf{R}} (\mathbf{B}^* \cdot \mathbf{b})}_{\equiv B_{\parallel}^*} - e \mathbf{B}^* \underbrace{(\dot{\mathbf{R}} \cdot \mathbf{b})}_{=v_{\parallel}} &= \mathbf{b} \times (\mu \nabla B + e \nabla \langle \Phi \rangle_{\mathbf{x}}) \\ \Leftrightarrow \dot{\mathbf{R}} &= v_{\parallel} \frac{\mathbf{B}^*}{B_{\parallel}^*} + \frac{\mathbf{b}}{e B_{\parallel}^*} \times (\mu \nabla B + e \nabla \langle \Phi \rangle_{\mathbf{x}}) \end{aligned} \quad (2.34)$$

where we introduced $B_{\parallel}^* = \mathbf{B}^* \cdot \mathbf{b}$. Projecting Eq. (2.33) onto $\dot{\mathbf{R}}$ this time:

$$\begin{aligned} 0 &= (m \dot{v}_{\parallel} \mathbf{b} + \mu \nabla B + e \nabla \langle \Phi \rangle_{\mathbf{x}}) \cdot \dot{\mathbf{R}} \\ \Leftrightarrow 0 &= m v_{\parallel} \dot{v}_{\parallel} + v_{\parallel} (\mu \nabla B + e \nabla \langle \Phi \rangle_{\mathbf{x}}) \cdot \frac{\mathbf{B}^*}{B_{\parallel}^*} \\ \Leftrightarrow \dot{v}_{\parallel} &= \frac{-\mathbf{B}^*}{m B_{\parallel}^*} \cdot (\mu \nabla B + e \nabla \langle \Phi \rangle_{\mathbf{x}}) \end{aligned} \quad (2.35)$$

We now have obtained expressions for all the time derivatives present in the coefficients of the Vlasov equation Eq. (2.28). However, instead of keeping the term $\mathbf{B}^*/B_{\parallel}^*$ in the expression of $\dot{\mathbf{R}}$, we would rather expand it to obtain a more common form. Keeping in mind that $\mathbf{b} \cdot (\nabla \times \mathbf{b}) = O(\varepsilon)$, a first order expansion in ε yields

$$\begin{aligned} \frac{\mathbf{B}^*}{B_{\parallel}^*} &= \mathbf{b} - \frac{mv_{\parallel}}{eB} \mathbf{b} \cdot (\nabla \times \mathbf{b}) + \frac{mv_{\parallel}}{eB_{\parallel}^*} \nabla \times \mathbf{b} + O(\varepsilon^2) \\ &= \mathbf{b} + \frac{mv_{\parallel}}{eB_{\parallel}^*} [\nabla \times \mathbf{b} - \mathbf{b} \cdot (\nabla \times \mathbf{b})] + O(\varepsilon^2) \end{aligned} \quad (2.36)$$

the two parts of the first order term, contained within the square brackets, partially cancel each other leaving only its component perpendicular to $\mathbf{b}$ which can also be written:

$$\nabla \times \mathbf{b} - \mathbf{b} \cdot (\nabla \times \mathbf{b}) = \mathbf{b} \times [(\nabla \times \mathbf{b}) \times \mathbf{b}]. \quad (2.37)$$

Finally, using

$$\nabla \times \mathbf{b} = \nabla \times \left(\frac{\mathbf{B}}{B} \right) = \frac{1}{B} \nabla \times \mathbf{B} - \frac{1}{B^2} \nabla B \times \mathbf{B}, \quad (2.38)$$

$\dot{\mathbf{R}}$ can be reformulated as

$$\dot{\mathbf{R}} = v_{\parallel} \mathbf{b} + \frac{1}{eB_{\parallel}^*} \mathbf{b} \times \left(e \nabla \langle \Phi \rangle_x + \frac{E'}{B} \nabla B + \frac{mv_{\parallel}^2}{B} (\nabla \times \mathbf{B}) \times \mathbf{b} \right) \quad (2.39)$$

with the new coefficient $E' = mv_{\parallel}^2 + \mu B$.

Final expression of the Vlasov equation

We now have workable expressions of all the time derivatives present in the Vlasov equation Eq. (2.28). It can now be rewritten as

$$\frac{\partial f}{\partial t} + \dot{\mathbf{R}} \cdot \frac{\partial f}{\partial \mathbf{R}} + \dot{v}_{\parallel} \frac{\partial f}{\partial v_{\parallel}} = 0 \quad (2.40)$$

with the coefficients

$$\dot{\mathbf{R}} = v_{\parallel} \mathbf{b} + \frac{1}{eB_{\parallel}^*} \mathbf{b} \times \left(e \nabla \langle \Phi \rangle_x + \frac{E'}{B} \nabla B + \frac{mv_{\parallel}^2}{B} (\nabla \times \mathbf{B}) \times \mathbf{b} \right) \quad (2.41a)$$

$$\dot{v}_{\parallel} = \frac{-\mathbf{B}^*}{mB_{\parallel}^*} \cdot (e \nabla \langle \Phi \rangle_x + \mu \nabla B), \quad (2.41b)$$

The expression of $\dot{\mathbf{R}}$ can be broken down into:

- $\mathbf{E} \times \mathbf{B}$ -drift $\mathbf{v}_{\mathbf{E} \times \mathbf{B}} = \frac{1}{B_{\parallel}^*} (-\nabla \langle \Phi \rangle_x) \times \mathbf{B}$,
- curvature drift $\mathbf{v}_c = \frac{E'}{eBB_{\parallel}^*} \mathbf{b} \times \nabla B$,
- ∇B -drift $\mathbf{v}_{\nabla B} = \frac{mv_{\parallel}^2}{eBB_{\parallel}^*} (\nabla \times \mathbf{B})_{\perp}$.

This equation applies to the distribution function of either species within the plasma: ions and electrons, with their respective mass and charge. The interaction between those species is taken into account through the electric potential Φ as will be shown in section 2.4, below. A collision operator could also be added to the right side of the equation to account for the collisions both within one species and between different ones, although collision are not included in the present study.

When a Vlasov equation is used to evolve the electrons, they are said to be **kinetic electrons**. Alternatively, as the mass of the electrons m_e is very small compared to the one of the ions m_i , the approximation $m_e \rightarrow 0$ can be made, reducing their dynamics to an instantaneous response to the ions' configuration and thus making the integration of the Vlasov equation for their distribution unnecessary. While greatly loosening the constraint on the time step of simulations as the characteristic speeds becomes of the order of the ion thermal velocity but also yields an additional term in the field equation whose computation can require complex methods. The use and consequences of those so-called **adiabatic electrons** is explained with more details in section 2.4.3.

2.3.2 Splitting between equilibrium distribution and perturbation

To simplify the equation, the distribution function is sometimes split between a time-independent canonical equilibrium part F_0 (which might or might not be perfectly correct) and a small perturbed part δf as:

$$f(t, \mathbf{R}, \mu, v_{\parallel}) = F_0(\mathbf{R}, \mu, v_{\parallel}) + \delta f(t, \mathbf{R}, \mu, v_{\parallel}), \quad (2.42)$$

where we expect $\|\delta f\|_2 / \|F_0\|_2 \sim \varepsilon \ll 1$. F_0 was here said to be "a canonical equilibrium" rather than just "an equilibrium" as it may be obtained through simplifications that do not apply perfectly to the geometry or configuration considered. Rather than an actual equilibrium, F_0 is to be understood as a near-equilibrium distribution.

The δf equation

Once used in the Vlasov equation (2.40), it can be decomposed into:

$$\frac{\partial \delta f}{\partial t} + \dot{\mathbf{R}}_1 \frac{\partial \delta f}{\partial \mathbf{R}} + \dot{v}_{\parallel 1} \frac{\partial \delta f}{\partial v_{\parallel}} + \dot{\mathbf{R}}_0 \cdot \frac{\partial F_0}{\partial \mathbf{R}} + \dot{v}_{\parallel 0} \frac{\partial F_0}{\partial v_{\parallel}} = -S_0 - S_2, \quad (2.43)$$

where the coefficients $\dot{\mathbf{R}} = \dot{\mathbf{R}}_0 + \dot{\mathbf{R}}_1$ and $\dot{v}_{\parallel} = \dot{v}_0 + \dot{v}_1$ between their part containing Φ (indexed with 1) and their part without it (indexed with 0), and the right-hand side terms are:

$$S_0 = \dot{\mathbf{R}}_1 \frac{\partial F_0}{\partial \mathbf{R}} + \dot{v}_{\parallel 1} \frac{\partial F_0}{\partial v_{\parallel}} \quad (2.44a)$$

$$S_2 = \dot{\mathbf{R}}_0 \cdot \frac{\partial \delta f}{\partial \mathbf{R}} + \dot{v}_{\parallel 0} \frac{\partial \delta f}{\partial v_{\parallel}}. \quad (2.44b)$$

The first term, S_0 , corresponds to the potential error committed when defining the canonical equilibrium distribution (which is ideally zero) and the second term, S_2 , is the higher order residual. If the those two terms are kept, this decomposition is strictly equivalent to the initial modelisation, making the decomposition pointless. However artificially setting one or both terms to 0 can be of use for numerical simulations.

Whether the decomposition is used or not, gyrokinetic models are assumed to operate near an equilibrium, as was mentioned, for instance, when we assumed the vanishing of the electrostatic potential Φ_0 . Therefore an equilibrium distribution or an approximate one has to be provided as initial condition for simulation codes. Furthermore, as we will see in the next section, the canonical equilibrium appears in the field equation. For those 2 reasons, an F_0 is required at some point whether the δf decomposition is used or not and in the case of an approximate one, the term S_0 becomes a source of numerical error (especially since the norm of F_0 is much greater δf); using the decomposition and removing S_0 allows to mitigate this error.

The higher order term S_2 corresponds to the non-linearity of the Vlasov equation as the electrostatic potential Φ which is obtained by solving the field equation in which δf appears. Therefore, it is essential for some simulations aiming at studying the so-called non-linear phase, which will be briefly introduced in section 5.1. Removing S_2 can be of interest for simulations that stay in linear regime as it simplifies the use of methods such as Fourier transforms. The approaches (and the numerical codes based on them) in which S_2 is taken into account are said to be **full-f** (and might not even represent f as the sum of an equilibrium and perturbed part) while those which neglect it are called δf .

Canonical equilibrium

While there is no exact formal expression for an equilibrium distribution in general geometries, a Maxwellian distribution is often used as approximation:

$$F_0(\psi, \mathcal{H}_0) = \frac{n_0(\psi)}{(2\pi T(\psi)/m)^{3/2}} e^{-\mathcal{H}_0/T(\psi)} \quad (2.45)$$

where $\mathcal{H}_0 = \frac{1}{2}mv_{\parallel}^2 + \mu B$ is the Hamiltonian without the electrostatic potential (which vanishes at equilibrium), T is the temperature and n_0 is the equilibrium density. Here, the 2 latter are expressed as functions of the magnetic flux function Ψ which is itself a function of either $\mathbf{x}$ or $\mathbf{R}$ as it relates to the external magnetic field, which does not vary on the scale a gyroradius. A part of the normalisation factor, can be identified as the **thermal velocity** $v_{th} = \sqrt{T/m}$. Notice that this distribution is normalised for an integration over $\mathbf{v}$ as we always include the terms resulting from the change of variables in front of integrals over μ and $v_{\parallel}$ (*i.e.* $\iiint F_0 d^3v = \frac{B}{m} \iiint F_0 d\mu dv_{\parallel} d\vartheta = n_0$). For circular configurations, n_0 and T are usually functions of the radius of the form

$$n_0(\psi) = \exp\left(-\frac{\delta r}{L_n} \tanh\left(\frac{r(\psi) - r_{mid}}{\delta r}\right)\right) \quad (2.46a)$$

$$T(\psi) = \exp\left(-\frac{\delta r}{L_T} \tanh\left(\frac{r(\psi) - r_{mid}}{\delta r}\right)\right) \quad (2.46b)$$

of which figure 2.4 shows an overview. The parameter $1/L$ tends to make the slope steeper while δr makes the sloped part broader. Increasing either also shifts the point where the profile reaches halfway between its left and right values. Typically, a density profile would look like the blue curve on the left graph ($a/L \simeq 1$) while a temperature profile would correspond to the green curve on the same graph ($a/L \simeq 3$).

Here $r(\psi)$ is simply a re-labelling of the flux surfaces and r_{mid} half its maximum. While it does correspond to the radius in a circular configuration, its definition changes for more complicated magnetic fields. To avoid confusion with some circular radius, it will be introduced as α later, in section 4.1.

While such a canonical equilibrium is suitable for theoretical studies or simple simulations, it can be insufficient for more general cases. The choice of an unsuitable equilibrium distribution can lead to the development of large-scale steady flows which can prevent the onset of turbulence. Studies such as [Idomura et al., 2003] suggest that to avoid this problem, F_0 needs to conserve as closely as possible 3 motion invariants: the canonical toroidal angular momentum $P_{\varphi} = e\psi + mRv_{\parallel}$, the energy $\mathcal{H}_0$ and the magnetic moment μ . For this purpose, [Dif-Pradalier et al., 2008] introduced a slight modification to the Maxwellian distribution in which n and T

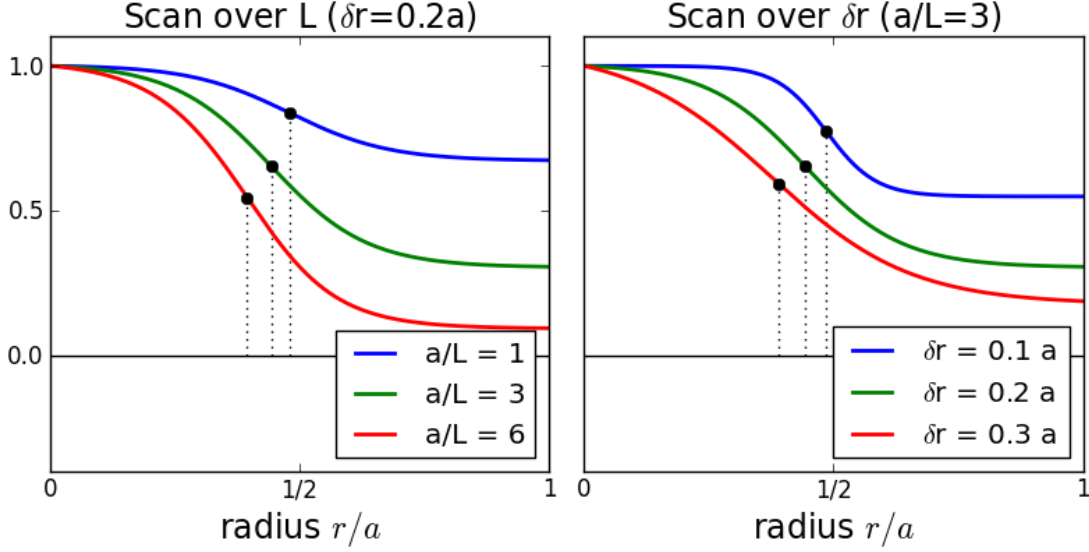


Figure 2.4: General aspect of density or temperature profiles as described by Eqs. (2.46a) and (2.46b). The black dot on each curve marks where the curve is halfway between its left and right values. Notice that the rescaling factor $\exp((\delta r/L) \tanh(-r_{mid})\delta r)$ was used to keep the curves within easily comparable ranges.

are no longer simply radial but function of a new coordinate $\bar{r}$ to replace $r(\psi)$ and defined as:

$$\bar{r} = r_0 - \frac{q_0}{r_0} (\psi - \psi(r_0)) - \frac{mq_0}{eB_0r_0} (Rv_{\parallel} - R_0\bar{v}_{\parallel}) \quad (2.47a)$$

$$\text{where } \bar{v}_{\parallel} = \text{sign}(v_{\parallel}) \sqrt{\frac{2}{m}} \sqrt{\mathcal{H}_0 - \mu B_{max}} H(\mathcal{H}_0 - \mu B_{max}) \quad (2.47b)$$

where the values bearing the index 0 are taken at mid-radius, H is the Heavyside function and B_{max} is the maximum magnetic field over the whole domain.

Alternatively, numerical codes can be used to compute an equilibrium distribution either by directly attempting to minimise S_0 or by using other modelisations such as MHD (see in particular [Takeda and Tokuda, 1991] for a description of the MHD equilibrium code MEUDAS). The second option, while potentially source of a higher numerical error, has the advantage of being more realistic as MHD is known to perform better with large-scale structures under very general hypotheses.

2.4 Field equation

The Vlasov equation governs the evolution of the distribution function of each species present in the plasma within given electric and magnetic fields. As the distribution of the species change, those fields also need to be updated in order to close the equation system. The electric and magnetic fields are also one of the ways the species interact with each other, along with the collisions which are not considered in this study.

2.4.1 Gyrokinetic quasi-neutrality condition

Within the electrostatic assumption we made earlier, only the perturbed electrostatic potential Φ . For this purpose, gyrokinetics relies on a reformulation of the Poisson equation called the gyrokinetic quasi-neutrality condition which expresses how any from charge neutrality can only happen on small spacial scales of the order of the Debye length λ_D .

While the distribution function of each species has now been transformed into their respective gyro-centre coordinates, the field equation combines them all and it would be impractically complicated to mix several coordinate systems or arbitrarily choose one of them as a reference. Instead, the quantities derived from the species distribution functions will have to be converted back into particle space coordinates, retaining the effects of the gyro-averaging they underwent while in gyro-centre coordinates. Those quantities are found in the right-hand side of the Poisson equation:

$$\nabla^2 \phi(\mathbf{x}) = -\frac{1}{\varepsilon_0} \sum_{\sigma} e_{\sigma} n_{\sigma}(\mathbf{x}) \quad (2.48)$$

where the summation goes over all the species present in the plasma and e_{σ} and n_{σ} are respectively the charge and density distribution of the species σ . Here, the density is defined as:

$$n_{\sigma}(\mathbf{x}) = \int f(\mathbf{x}, \mathbf{v}) d\mathbf{v} = \frac{B}{m_{\sigma}} \oint f_{\vartheta\sigma}(\mathbf{R}, \mu, v_{\parallel}, \vartheta) \delta(\mathbf{R} + \boldsymbol{\rho} - \mathbf{x}) d^3 R d\mu dv_{\parallel} d\vartheta \quad (2.49)$$

where $f_{\vartheta\sigma}$ is the distribution function before any gyro-averaging has been performed, and where the combination of the Dirac operator δ and the integration over $d^3 R$ allows the pinpoint a location $\mathbf{x}$ for a given gyro-angle φ . As this notation is a bit heavy, before continuing, let us introduce the operator $\langle \cdot \rangle_{\mathbf{R}} = \frac{1}{2\pi} \oint \delta(\mathbf{R} + \boldsymbol{\rho} - \mathbf{x}) d^3 R d\vartheta$ which is a gyro-averaging in guiding-centre coordinates, hence the similar notation as $\langle \cdot \rangle_{\mathbf{x}}$ used before.

The distribution $f_{\vartheta\sigma}$ is also the solution of a Vlasov equation we never expressed so far: in guiding centre coordinates without gyro-averaging, based on $\tilde{\Gamma}$. To express it using quantities we have already been using so far, one can either perform

proper Lie-transform perturbation decomposition as in [Brizard and Hahm, 2007], or calculate the difference between the Vlasov equation based on $\tilde{\Gamma}$ and Eq. (2.40) as in [Lee, 1983]. In both cases, a δf expansion also has to be used and the following expression is obtained:

$$f_{\vartheta\sigma}(\mathbf{R}, \mu, v_{\parallel}, \vartheta) = f_{\sigma}(\mathbf{R}, \mu, v_{\parallel}) - \frac{e_{\sigma}}{T_{\sigma}(\mathbf{x})} (\Phi(\mathbf{x}) - \langle \Phi \rangle_{\mathbf{x}}(\mathbf{R})) F_{0\sigma}(\mathbf{R}, \mu, v_{\parallel}) \quad (2.50)$$

i.e. the sum of the gyro-averaged distribution function, plus an oscillating part. The density then becomes

$$n_{\sigma}(\mathbf{x}) = \frac{2\pi B}{m_{\sigma}} \iint \left\langle f_{\sigma} - \frac{e_{\sigma}}{T_{\sigma}(\mathbf{x})} [\Phi(\mathbf{x}) - \langle \Phi \rangle_{\mathbf{x}}(\mathbf{R})] F_{0\sigma} \right\rangle_{\mathbf{R}} d\mu dv_{\parallel} \quad (2.51a)$$

$$= \bar{n}_{\sigma}(\mathbf{x}) - \frac{e_{\sigma} n_{0\sigma}(\mathbf{x})}{T_{\sigma}(\mathbf{x})} \Phi(\mathbf{x}) + \frac{e_{\sigma} n_{0\sigma}(\mathbf{x})}{T_{\sigma}(\mathbf{x})} \langle \langle \Phi \rangle \rangle_{\sigma}(\mathbf{x}) \quad (2.51b)$$

where $n_{0\sigma}$ is the equilibrium density already mentioned in Eq. 2.45 and we introduced the density of gyro-centres $\bar{n}_{\sigma}(\mathbf{x}) = \frac{2\pi B}{m_{\sigma}} \iint \langle f_{\sigma} \rangle_{\mathbf{R}} d\mu dv_{\parallel}$ as well as the so-called double gyro-averaging of the electrostatic potential:

$$\langle \langle \Phi \rangle \rangle_{\sigma}(\mathbf{x}) = \frac{1}{n_{0\sigma}} \frac{2\pi B}{m_{\sigma}} \iint \langle F_{0\sigma} \langle \Phi \rangle_{\mathbf{x}} \rangle_{\mathbf{R}} d\mu dv_{\parallel}. \quad (2.52)$$

As neither Φ nor the simple averagings depend on $v_{\parallel}$, the integral over it can be moved inside the brackets and applied solely to $F_{0\sigma}$ to leave only

$$F_{M\sigma}(\mathbf{R}, \mu) \equiv 2\pi \int_{-\infty}^{\infty} F_{0\sigma}(\mathbf{R}, \mu, v_{\parallel}) dv_{\parallel} = \frac{n_{0\sigma}}{v_{th\sigma}^2} e^{-\mu B/T_{\sigma}}. \quad (2.53)$$

Furthermore, as mentioned in section 2.3.2, all the observables found in this expression do not vary on the scale of the the gyroradius. Just like the density or the temperature, $F_{M\sigma}$ can be expressed as a function of either $\mathbf{x}$ or $\mathbf{R}$.

Replacing the density n_{σ} by its new expression in Poisson's equation Eq. (2.48), we finally obtain the gyrokinetic quasi-neutrality condition:

$$-\nabla^2 \phi(\mathbf{x}) + \sum_{\sigma} \frac{1}{\lambda_{D\sigma}^2} [\Phi(\mathbf{x}) - \langle \langle \Phi \rangle \rangle_{\sigma}(\mathbf{x})] = \frac{1}{\varepsilon_0} \sum_{\sigma} e_{\sigma} \bar{n}_{\sigma}(\mathbf{x}) \quad (2.54)$$

where we used the Debye length $\lambda_{D\sigma} = \left(\frac{\varepsilon_0 T_{\sigma}}{e_{\sigma}^2 n_{0\sigma}} \right)^{1/2}$ for the species σ ¹ which are related to the total Debye length by the relation $\lambda_D^{-2} = \sum \lambda_{D\sigma}^{-2}$. The term between square brackets is sometimes referred to as the polarisation term.

¹ The Debye length characterises the distance over which a charge imbalance can influence the charged particles of the plasma before its effects are screened. It is a local property of the whole plasma and not of species taken individually. The constants $\lambda_{D\sigma}$ used here are mere notations with no actual physical meaning.

The electrostatic potential is expected to vary over the scale of the thermal gyroradius of the ion species, making the Laplacian term $\nabla^2 \Phi$ of the order of $\Phi/\rho_{i,th}^2$ at most. While in magnetically confined fusion plasmas, the electron's gyroradius is smaller than the Debye length (a simple calculation with ITER-like conditions yields $\lambda_D/\rho_{e,th} \simeq 3$ and [Jenko and Kendl, 2002] suggests 1.8 for the stellarator Wendelstein 7-AS), the ion gyroradius is several tens of times larger as $\rho_{i,th}/\rho_{e,th} \simeq \sqrt{m_i/m_e} \simeq 42$ for 1H and ever more for heavier ions. For this reason, the Laplacian term is often neglected and left out in the expression of the field equation.

2.4.2 Gyro-averaged terms in Fourier space

For simplicity, the index σ is dropped for the time of this section as all the considerations are general and apply to all species.

The gyrokinetic quasi-neutrality equation Eq. (2.54) we just derived contains 2 new gyro-averaging operators $\langle \cdot \rangle_{\mathbf{R}}$ and $\langle \cdot \rangle_{\mathbf{x}}$ in addition to $\langle \cdot \rangle_{\mathbf{x}}$ found in the Vlasov equation Eq. (2.40). While the initial definition we used when introducing $\langle \cdot \rangle_{\mathbf{x}}$ is not very practical, it can easily be defined under the same form as the 2 latter. To summarise:

$$\langle \cdot \rangle_{\mathbf{x}} = \frac{1}{2\pi} \int \cdot \delta(\mathbf{R} + \boldsymbol{\rho} - \mathbf{x}) d^3x d\vartheta \quad (2.55a)$$

$$\langle \cdot \rangle_{\mathbf{R}} = \frac{1}{2\pi} \int \cdot \delta(\mathbf{R} + \boldsymbol{\rho} - \mathbf{x}) d^3R d\vartheta \quad (2.55b)$$

$$\langle \cdot \rangle = \frac{1}{n_0} \frac{B}{m} \int \langle F_M(\mathbf{R}, \mu) \langle \cdot \rangle_{\mathbf{x}} \rangle_{\mathbf{R}} d\mu \quad (2.55c)$$

These gyro-averagings are one of the signatures of gyrokinetics and have been at the core of numerous theoretical and numerical methods. The Dirac operators restrict the integrals to trajectories which are simple in the case of the simple averagings (a circle) but quite complicated in the case of the double averaging. Section 3.3 introduces a method which exploits such path integral interpretations and is widely used in PIC codes. However, most methods rather rely on Fourier transforms in local Cartesian coordinates to express them. Decomposing:

$$\Phi(\mathbf{x}) = \int d\mathbf{k} \hat{\Phi}_{\mathbf{k}} e^{i\mathbf{k} \cdot \mathbf{x}} = \int d\mathbf{k} \hat{\Phi}_{\mathbf{k}} e^{i\mathbf{k} \cdot \mathbf{R} + i\mathbf{k} \cdot \boldsymbol{\rho}} \quad (2.56)$$

the integration over ϑ can be expressed quite simply as

$$\langle \Phi \rangle_{\mathbf{x}} = \int d\mathbf{k} \hat{\Phi}_{\mathbf{k}} e^{i\mathbf{k} \cdot \mathbf{R}} \frac{1}{2\pi} \oint e^{i\mathbf{k} \cdot \boldsymbol{\rho}} d\vartheta = \int d\mathbf{k} \hat{\Phi}_{\mathbf{k}} e^{i\mathbf{k} \cdot \mathbf{R}} \frac{1}{2\pi} \oint e^{k_{\perp} \rho \cos \vartheta} d\vartheta \quad (2.57)$$

with $k_{\perp} = \mathbf{k} - (\mathbf{k} \cdot \mathbf{b})\mathbf{b}$ is the poloidal component of $\mathbf{k}$. This final integral term happens to be one of the definitions of the Bessel function of the first kind J_0 ,

allowing the simple expression:

$$\langle \Phi \rangle_{\mathbf{x}} = \int d\mathbf{k} \hat{\Phi}_{\mathbf{k}} J_0 \left(\frac{k_{\perp} v_{\perp}}{\Omega} \right) e^{i\mathbf{k} \cdot \mathbf{x}}. \quad (2.58)$$

The same of course applies to the other version of the simple averaging $\langle \cdot \rangle_{\mathbf{R}}$ as the expression is symmetric with respect to the sign in front of ρ .

Similarly, the double gyro-averaging can be expressed simply in Fourier space:

$$\langle\langle \Phi \rangle\rangle \simeq \frac{1}{n_0} \frac{B}{m} \int_0^\infty F_M \langle \langle \Phi \rangle_{\mathbf{x}} \rangle_{\mathbf{R}} d\mu \quad (2.59a)$$

$$= \frac{1}{n_0} \frac{B}{m} \int d\mathbf{k} \hat{\Phi}_{\mathbf{k}} \int_0^\infty F_M J_0 \left(\frac{k_{\perp} v_{\perp}}{\Omega} \right)^2 d\mu e^{i\mathbf{k} \cdot \mathbf{x}} \quad (2.59b)$$

$$= \frac{2}{v_{th}^2} \int d\mathbf{k} \hat{\Phi}_{\mathbf{k}} \int_0^\infty J_0 \left(\frac{k_{\perp} v_{\perp}}{\Omega} \right)^2 \exp \left(\frac{-v_{\perp}^2}{2v_{th}^2} \right) v_{\perp} dv_{\perp} e^{i\mathbf{k} \cdot \mathbf{x}} \quad (2.59c)$$

where we used the fact that F_M is roughly constant on the scale of the gyroradius to move F_M out of the gyro-averaging's bracket. After replacing F_M with its expression Eq. (2.53), the integral can be calculated with the introduction of a new function $\Gamma_0 : b \mapsto I_0(b)e^{-b}$ (I_0 being the modified Bessel function of the first kind) as:

$$\frac{1}{v_{th}^2} \int_0^\infty J_0 \left(\frac{k_{\perp} v_{\perp}}{\Omega} \right)^2 \exp \left(\frac{-v_{\perp}^2}{2v_{th}^2} \right) v_{\perp} dv_{\perp} = \Gamma_0 \left(\frac{k_{\perp}^2 v_{th}^2}{\Omega^2} \right). \quad (2.60)$$

which, using $\rho_{th} = v_{th}/\Omega$, allows to rewrite:

$$\langle\langle \Phi \rangle\rangle = \int d\mathbf{k} \hat{\Phi}_{\mathbf{k}} \Gamma_0(k_{\perp}^2 \rho_{th}^2) e^{i\mathbf{k} \cdot \mathbf{x}} \quad (2.61)$$

This expression is sometimes found further simplified using a **Padé approximation** (or more simply Taylor expansion) of Γ_0 . For $k_{\perp} \rho_{th}$ small:

$$\Gamma_0(k_{\perp}^2 \rho_{th}^2) \simeq \frac{1}{1 + k_{\perp}^2 \rho_{th}^2} \simeq 1 - k_{\perp}^2 \rho_{th}^2 \quad (2.62)$$

which transforms Eq. (2.61) into:

$$\langle\langle \Phi \rangle\rangle \simeq \int d\mathbf{k} \hat{\Phi}_{\mathbf{k}} (1 - k_{\perp}^2 \rho_{th}^2) e^{i\mathbf{k} \cdot \mathbf{x}} = \Phi + \rho_{th}^2 \nabla_{\perp}^2 \Phi \quad (2.63)$$

While the expression of the 2 gyro-averagings Eq. (2.58) and Eq. (2.61) are quite simple and can greatly help for theoretical studies, their usage for numerical purposes is more complicated in most cases. The initial decomposition in "local Cartesian coordinates" is not an easy operation to implement, especially when the mesh size is often of the order of the thermal gyro-radius. A more global Fourier transform in polar coordinates can be used for circular cross-sections, far from the magnetic axis. The Padé approximation can circumvent this issue for the double gyro-averaging, but the constraint $k_{\perp} \rho_{th} \ll 1$ forbids the study of some physical phenomena.

2.4.3 Adiabatic electrons

Electron adiabatic response

The gyro-ordering introduced at the beginning of this chapter compared the spacial and temporal scales of the ion dynamics to the device scale. However, the smallness of the mass of the electron compared to any ion species places its dynamics in even smaller and faster scales. To simplify calculations or computations and not have to fully take into account the scales associated with electrons, their dynamics is sometimes approximated to an adiabatic response.

Taking the limit of massless electrons $m_e \rightarrow 0$ while keeping their kinetic energy constant implies an infinite velocity and thus an instantaneous response to any force imbalance. Looking once again the Vlasov equation Eq (2.40) and the expression of $\mathbf{R}$, the infinite $v_{\parallel}$ dominates the 2 right terms which are kept finite; since nothing can be said *a priori* on the term containing Φ , it can be kept as well. Keeping only these 2 terms, we can see that the movement of the electrons will be restricted on a magnetic flux surfaces and that any electrostatic imbalance on it will be instantly short-circuited by the electrons. It can then be shown that the $\Phi - \langle\langle\Phi\rangle\rangle_e$ in Eq. (2.54) is thus be replaced by $\Phi - \bar{\Phi}$ where $\bar{\Phi}$ is the flux-surface averaging of the electrostatic potential. The quasi-neutrality condition with adiabatic electrons is thus rewritten as:

$$\frac{1}{\lambda_{Di}^2} [\Phi - \langle\langle\Phi\rangle\rangle] + \frac{1}{\lambda_{De}^2} [\Phi - \bar{\Phi}] = \frac{1}{\varepsilon_0} (q_i \bar{n}_i + q_e \bar{n}_e). \quad (2.64)$$

The Zonal Flow equation

The magnetic flux averaged term $\bar{\Phi}$ in Eq. (2.64) plays a crucial role in ZF dynamics. It can be easily isolated from the rest of the equation and calculated separately. For this, averaging the field equation over the magnetic flux surfaces yields the ZF equation:

$$\bar{\Phi} - \overline{\langle\langle\Phi\rangle\rangle} = \frac{T_i m_i^2}{e_i n_0} \int B^* \langle f_i - f_0 \rangle dv_{\parallel} d\mu \quad (2.65)$$

which allows us to compute $\bar{\Phi}$ before the field equation itself is solved. The term can then be moved to the right hand side of the field equation and the resolution continue easily. Since the 3-dimensional electrostatic potential Φ is averaged over the flux-surfaces, the ZF equation is simply a one-dimensional equation, depending only the radius in the circular case or more generally a flux-surface labelling.

However, despite the simplicity of the expression of the ZF equation, calculating the coefficients of the operator $\overline{\langle\langle\cdot\rangle\rangle}$ and the flux averaging of the right hand side of the equation can become quite complicated, especially in non-circular magnetic configurations. Two different ZF solvers will be introduced in Chapter 4.

2.5 Simulating gyrokinetics

The high dimensionality of gyrokinetics makes simulations extremely heavy, with numerical grids of several billion points required even for basic benchmark computations. For this reason, it is important when designing a code to think ahead of the advantages and weaknesses the various possible approaches. Furthermore, to lower the computational cost and allow the refinement of the model with the inclusion of more advanced elements, several strategies are commonly employed.

2.5.1 Simulation geometry and representation

Slab, toroidal and flux-tube geometries

An obvious strategy to reduce computational cost is restrict the simulation domain a smaller plasma volume, where the phenomena of interest takes places. Although the present thesis revolves around toroidal code in which all of the plasma, or at least most of it, is simulated, Chapter 3 mentions the code GKMUST, which uses the so-called **slab geometry**. As shown on Fig. 2.5, this geometry focuses on a small area of the plasma, slightly to the outside of the magnetic axis, completely neglecting the flux surfaces' curvatures and with periodic boundary conditions linking the top and bottom ends. This very restricted geometry can be very useful to study the formation of ballooning structures and the dynamics of magnetic islands as it focuses on the relevant area [Hilscher, 2013].

A different approach which allows considering a much larger part of a toroidal plasma while effectively reducing the volume being simulated is the **flux-tube** approach. Such codes follow the evolution of a plasma volume enclosed in magnetic field lines and sufficiently small so that the transversal variations of global quantities such as the density and temperature profiles can be neglected.

Lagrangian and Eulerian representations

It is common in fluid dynamics to exploit the duality between the Lagrangian and Eulerian representations of fluids. Plasma being a particular type of fluid, a similar duality exists in gyrokinetics too. The most direct approach in integrating the Vlasov equation is the Lagrangian one. The 5-dimensional phase space is simply discretised on a grid, without requiring any transformation. As one can expect, this direct method is very heavy, even more since it requires the use of explicit integration schemes with very restrictive CFL conditions.

While the Vlasov equation is in essence of the Lagrangian category, **Particle-In-Cell** (PIC) codes are a family of Eulerian methods which interpret the distribution function as a discrete sum of super-particles. The coupling of the fluid equation with the field equation implies that the super-particles must be re-aggregated on a

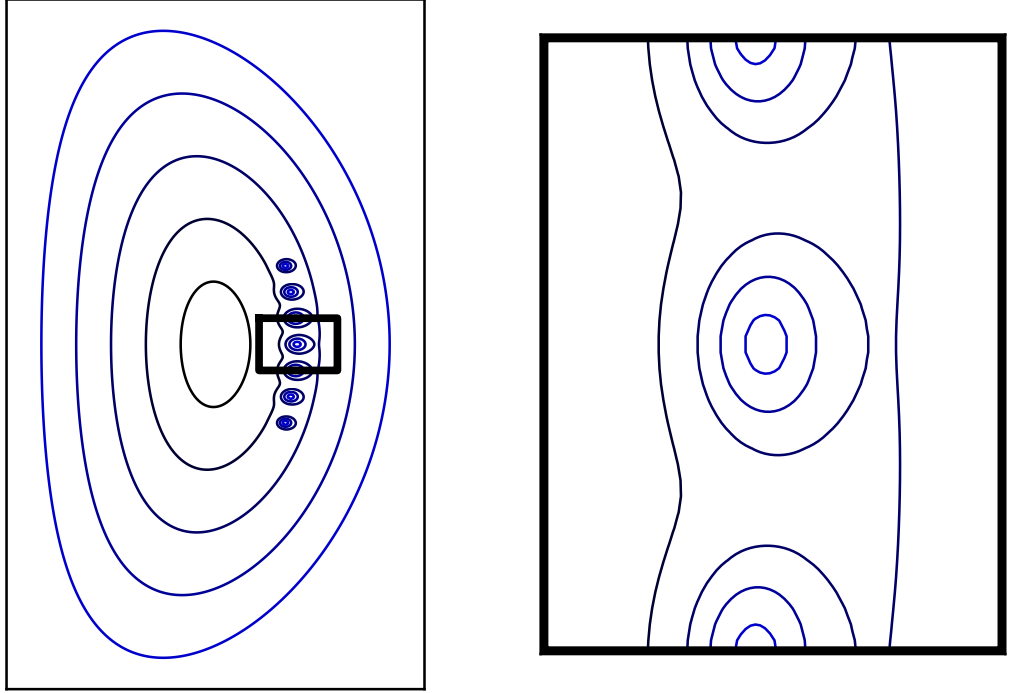


Figure 2.5: The so-called slab geometry focuses on a small area of interest for which curvature can be neglected and periodic boundary conditions are assumed in the vertical direction.

Lagrangian grid while updating the fields before reverting to the Eulerian representation by re-sampling new super-particles.

PIC codes are the most popular tool as they are capable of describing many physical phenomena for a relatively small computational cost. The price for this lightness is having to deal with an inherent numerical noise whose scaling is problematic for highly accurate simulations, as it only decreases with $1/\sqrt{N}$, where N is the number of super-particles. Moreover, it has been observed that in the case of nonlinear simulations, the initial numerical noise can only increase with time. The control and reduction of numerical noise has been at the heart of most recent research in PIC codes.

2.5.2 Physics included in the simulation model

Local and global simulations

The section 2.3.2 above has introduced the possible splitting of the distribution function between a time-independent canonical equilibrium part F_0 and a small perturbation δf . This concept plays an important role in the writing a simulation code as some categories of codes have such a distinction "hard-wired" into their design. Indeed, numerically, a " δf " only evolves that δf perturbation component while the equilibrium distribution plays the role of a fixed background which also fixes the bulk of the density or temperature profiles. However, in the absence of an exact theoretical expression for the equilibrium, the background provided by F_0 contains numerical error which can perturbate the onset of the perturbation, especially in the case of stiff profiles. While some progress has been made in the handling and correction of such equilibrium error, δf are still known to yield weaker dependencies with respect to plasma shaping for some phenomena compared to what has been experimentally observed [Belli et al., 2008].

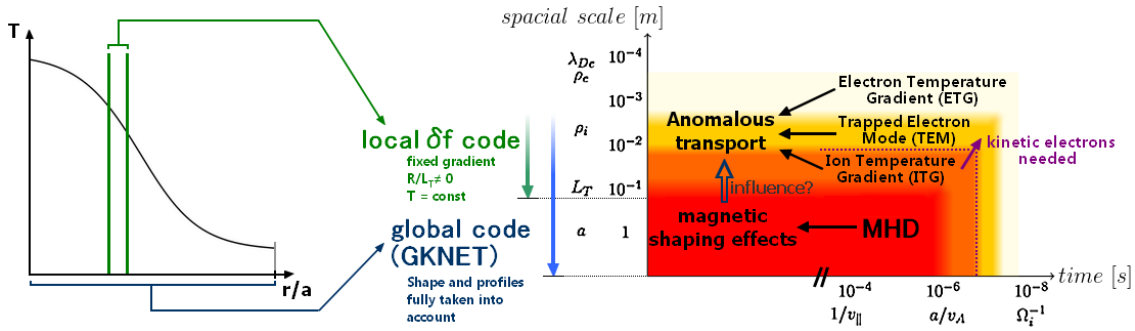


Figure 2.6: Various scales of time and space involved in plasma dynamics along with their interaction through the influence of the magnetic configuration's shape back on the microscopic instabilities. The left part also highlights how much can be expected to be properly taken into account by local δf and global full-f codes.

Alternatively, codes which are said to be full-f codes do not impose such a strong distinction on the equilibrium part F_0 and the perturbation δf as both are evolved with time, following the same equation. One of the objective of the present thesis is to demonstrate the advantage of such full-f codes, in particular in the study of magnetic shaping effects on micro-instability.

Ion and electron species

The inclusion of several species in the plasma is fairly simple in principle. Each species is described by its own distribution function and the right hand side of the

field equation simply sums all the contributions to determine the electrostatic potential. The interactions between the species are thus found in the field equation, as well as in the collision term of the Vlasov equation, if they are considered. However, while the addition of ion species is a quite direct task, the dynamics of electrons is different in some regards and greatly complicates the situation.

In the adiabatic electron approximation, the exact dynamics of electrons is replaced by their adiabatic response to the ions' configuration. The reason for such an approximation comes from the fact that electrons are much lighter (1836 times lighter than the lightest ion which is a mere proton), making their dynamics' characteristic time scale much smaller than the one of ions, while also reducing the influence they can have on them. It is thus common for simulation codes to make use of the adiabatic electron approximation, which allows for a 43-fold ($\simeq \sqrt{1836}$) reduction of the simulation timestep.

Electromagnetic codes

When deriving the first order perturbation of the one-form Eq. 2.24, we made the assumption that the magnetic field did not vary on the small scale of the gyro-radius, therefore removing the term $\mathbf{A}_1$. While very helpful, this assumption was a mere model choice and not an absolute necessity. Simulation codes whose core model does not rely on this assumption are said to be electromagnetic, by opposition to our own electrostatic case, as they update the magnetic field using the Maxwell Ampère law in addition to the electrostatic potential with the quasi-neutrality condition Eq. (2.54).

2.5.3 Overview of the existing simulation codes

Finally, we present here a non-exhaustive list of existing codes, along with some of their characteristics and a reference. In addition to the terms already introduced above, the following abbreviations are used:

- KE = Kinetic Electrons,
- EM = ElectroMagnetic,
- HE = Hybrid Electrons [Idomura, 2016].

Name	Type	Notes	Ref.
GENE	Flux-tube + Eulerian	KE, EM	[Jenko, 2000]
GYRO	Flux-tube + Eulerian	KE, EM	[Candy and Waltz, 2003]
GKW	Flux-tube + Eulerian	KE, EM	[Peeters et al., 2009]
GS2	Flux-tube	KE, EM	[Dorland et al., 2000]
GTC	PIC	KE, EM	[Lin et al., 1998]
GT3D	PIC	KE, EM	[Idomura et al., 2003]
ORB5	PIC	KE, EM	[Jolliet et al., 2007]
XGC1	PIC	KE, EM	[Chang et al., 2009]
GYSELA	Semi-Lagrangian	full-f circular	[Grandgirard et al., 2016]
GT5D	Eulerian	full-f, HE	[Idomura et al., 2008]
GKNET	Eulerian	full-f	[Obrejan et al., 2017]

2.6 Summary

Taking advantage of the gyro-ordering, which separates slow macroscopic scales from fast microscopic ones, we have transformed the one-form of the Lagrangian of a single particle from a 6 dimensional phase space into a 5 dimensional gyro-centre, coordinate system from which the dynamics associated to the gyro-phase had been averaged. The gyrokinetic Vlasov equation was then derived from this gyro-averaged Lagrangian, as the first half of our gyrokinetic model. The system was completed by applying the same gyro-averaging procedure to the Poisson equation, thus deriving the gyrokinetic quasi-neutrality condition. A few common simplifications were presented, such as the adiabatic electrons, which replaces kinetic electrons by their adiabatic response, or the delta-f decomposition, which linearises the equation system around the initial equilibrium distribution.

Finally, the various strategies commonly used in simulating magnetically confined fusion plasmas were presented, along with a short list of some of the major simulation codes. This overview gives a context for the introduction of our own code, GKNET, which is noticeably different that most codes due to its full-f nature.

Chapter 3

Design of GKNET

Contents

3.1	The slab geometry code GKMUST	45
3.1.1	General design	45
	Regular numerical grid	45
	Non-uniform grid in μ	46
3.1.2	Higher order finite differences	46
3.1.3	Time integration	48
	Explicit and implicit methods	48
	The forth-order explicit Runge-Kutta method (RK4)	48
3.2	From slab to toroidal geometry	49
3.2.1	GKMUST and the several GKNETs	49
3.2.2	Numerical grid and boundary conditions	51
3.2.3	Variable time step	52
	Simple case: CFL conditions of a 1D transport equation	52
	Applying linear tools to a nonlinear equation	54
3.3	Real space averaging method	55
3.3.1	Simple gyro-averaging	55
3.3.2	Double gyro-averaging	57
	From simple to double gyro-averaging	57
	Integration over μ	59
3.4	Field solver	60
3.4.1	Implementation of the gyro-averagings	60

3. DESIGN OF GKNET

	Interpolation onto the grid	60
	Storage of the coefficients and computation time	61
3.4.2	Fixed-point method	62
	Fixed-point form of the field equation	63
	Speeding up the convergence	64
3.5	Parallelisation	66
3.5.1	Cross-section shape in Cartesian coordinates	66
3.5.2	Sub-parallelisation μ and $v_{\parallel}$ of the gyro-averagings	68
3.5.3	Allocation of available processes	68
3.6	Summary	69

GKNET is an electrostatic gyrokinetic full-f code with adiabatic electrons and the particularity of using a field solver in real space. Solving the field equation in real space instead of relying on Fourier space representation of the gyro-averaging operators allows for better treatment of the physics at the scale of the Larmor radius. Furthermore, the use of Cartesian coordinates in the poloidal cross-section alleviates some potentially problematic restrictions on the time step and allows for arbitrary magnetic configurations.

The code was developed over the course of several years, starting off as a simple slab geometry code in which the field equation was solved in Fourier space. After a field solver in real-space was developed and implemented, the code evolved into cylindrical geometry and later a circular toroidal geometry in polar coordinates. The change to Cartesian coordinates then allowed to take into account arbitrary cross-section shapes. This chapter explains the key features of the code, starting from the core of the code as it was inherited from the slab geometry code GKMUST, before introducing the field solver in real space and other key features of GKNET.

3.1 The slab geometry code GKMUST

3.1.1 General design

Regular numerical grid

The numerical grid of GKMUST was regular in the 2 spacial dimensions as well as in $v_{\parallel}$ to maximise numerical accuracy of the finite difference schemes used to compute the respective derivatives. This idea was kept in the implementation of GKNET, with, of course, the addition of a regular grid along the new spacial dimension: the toroidal angle. The derivative of the distribution function with respect to θ , does not appear in the Vlasov equation (collision operators involving such a derivative were not used), therefore allowing for more general grids in that direction. The μ discretisation is instead not regular but generated using a Gauss-Legendre quadrature as is explained more in details in the next section.

For the sake of generality, the space dimensions will either be referred to as x , y (and z for GKNET), as is the case in the code, or by their actual physical name (r , θ , R etc). While the space dimensions are physically bounded, the speed ones are artificially bounded by the choice of a cut-off value; in both cases the size of the interval is denoted L_i with i the dimension's name and N_i is the number of mesh points. The mesh points can then be written from x_1 to x_{N_x} , the position of those points and the corresponding boundary conditions for the dimensions other than μ being explained later in section 3.2.2.

Non-uniform grid in μ

Regular grids allow for much larger accuracies when computing finite differences. However, with no derivative in μ present in the equation system, this strategy is irrelevant. Instead, the choice of discretisation in μ influences the accuracy of the simulation through the integration over μ performed by the field solver to compute the right-hand side of the field equation Eq. (2.64). This integration is performed with a Gauss-Legendre quadrature in order to allow for high accuracy with as few points as possible. The method samples the integrated function at sample points which do not form a regular grid.

Although μ takes values ranging from 0 to μ , the distribution function decreases roughly like a Maxwellian function, allowing the usage of a cut-off value L_μ such that $f(L_\mu) \ll 1$. Considering the integration of a function h over an interval $[0, L_\mu]$, the Gauss-Legendre quadrature approximates

$$\int_0^{L_\mu} h(\mu) d\mu = \frac{L_\mu}{2} \int_{-1}^1 h\left(\frac{L_\mu}{2}(\mu' + 1)\right) d\mu' \simeq \frac{L_\mu}{2} \sum_{n=1}^{N_\mu} w_n h(\mu_n) \quad (3.1)$$

where the weights w_n and abscissae μ_n are calculated using the Legendre polynomials (hence the name). The Legendre polynomials have several equivalent definitions, one of them being, for the n -th polynomial $P_n(x)$:

$$P_n(x) = \frac{1}{2^n n!} \cdot \frac{d^n}{dx^n} [(x^2 - 1)^n]. \quad (3.2)$$

From them, the abscissae $\mu'_n \equiv 2\mu_n/L_\mu - 1$ (*i.e.* such that $\mu_n = \frac{L_\mu}{2}(\mu'_n + 1)$) are formally defined as the N_μ roots of P_{N_μ} and the corresponding weights as:

$$w_n = \frac{2}{(1 - (\mu'_n)^2) P'_{N_\mu}(\mu'_n)^2}. \quad (3.3)$$

The Gauss-Legendre quadrature is known to be exact when h is a polynomial of order up to $2N_\mu - 1$.

As the mesh number N_μ is expected to be a power of 2, pre-computed sets of coefficients for $N_\mu \in \{8, 16, 32, 64\}$ are integrated in the code. The number of those coefficients can be halved by using the evenness of P_n for n even, which implies: $\mu'_n = \mu'_{N_\mu+1-n}$ and $w_n = -w_{N_\mu+1-n}$.

3.1.2 Higher order finite differences

First off, let us define the notations used in this section and the next ones. For simplicity, the discretisation of a variable will be denoted with the same symbol

with time indices placed as exponents and the remaining ones (spacial or speed) as subscripts. For a variable A at time t_n and space coordinate x_i :

$$A_i^n \equiv A(t_n, x_i), \quad (3.4)$$

with as many space indices as necessary depending on the context and with some of those indices omitted when not needed. All grids will be assumed to be regular. Parentheses with such exponents and subscripts can also be used to refer to the evaluation of a composed expression at those indices.

In addition to the time derivative, the collisionless Vlasov equation eq. (2.40) contains 4 other derivatives: the 3 spacial dimensions and the parallel velocity $v_{\parallel}$. Finite differences are the most common approach to calculate space derivatives of partial differential equations and a natural choice in this situation. One of the simplest finite difference schemes for a first order derivative on a regular mesh is the central finite differences:

$$\left[\frac{\partial A}{\partial x} \right]_i^{CD3} = \frac{A_{i+1} - A_{i-1}}{2\Delta x}. \quad (3.5)$$

It is easy to verify with a Taylor expansion that this central finite difference method is of order 2 *i.e.* that the error term is $O(\Delta x^2)$. However, for numerical simulations, higher levels of accuracy are usually preferable. GKMUST used a 4-th order accurate central finite difference scheme, which requires a 5-points stencil:

$$\begin{aligned} \left[\frac{\partial A}{\partial x} \right]_i^{CD5} &= \frac{-A_{i+2} + 8A_{i+1} - 8A_{i-1} + A_{i-2}}{12\Delta x} \\ &= \frac{1}{3\Delta x} \left(2(A_{i+1} - A_{i-1}) - \frac{1}{4}(A_{i+2} - A_{i-2}) \right). \end{aligned} \quad (3.6)$$

Higher order finite difference schemes are easy to derive using Taylor expansions, however the number of neighbouring points in the stencil increases with the order and with it, the computation time of the derivatives as well as the number of ghost cells required outside the boundaries of the domain. Since the number of ghost cells is expected to be small compared to the size of the domain, this latter factor can seem irrelevant. However it can be a one of the key factors that limits the efficiency of a parallelisation as the same number of ghost cell is required around each sub-block of the domain. As the number of cores increases, the proportion of ghost cells becomes larger. This leads either to high redundancy of the computations performed (if a core computes itself the value of the variables at the ghost cells) or large amounts of communications (if the cores share the value of the variables at their neighbours' ghost cells).

3.1.3 Time integration

The time integration scheme in the numerical resolution of a partial derivative equation is a crucial decision. Similarly to the previous section, the order of the time integration speed will determine how accuracy varies as a function of the time step, making higher order scheme preferable to lower order ones. However two other aspects have to be taken into account: whether the scheme is explicit or implicit and its stability conditions.

Explicit and implicit methods

The two simplest and most common examples of time integration schemes are the forward and backward Euler method. Considering a differential equation of the form $\partial_t A = \mathcal{L}(t, A)$, the forward Euler method reads:

$$\frac{A^{n+1} - A^n}{\Delta t} = \mathcal{L}(t_n, A^n) \quad \Leftrightarrow \quad A^{n+1} = (1 + \Delta t \cdot \mathcal{L}) A^n, \quad (3.7)$$

allowing the direct calculation of A^{n+1} from A^n ; it is said to be an **explicit** method. On the other hand, the backward Euler method reads:

$$\frac{A^{n+1} - A^n}{\Delta t} = \mathcal{L}(t_n, A^{n+1}) \quad \Leftrightarrow \quad A^{n+1} = (1 - \Delta t \cdot \mathcal{L})^{-1} A^n. \quad (3.8)$$

requiring the resolution of an equation to find A^{n+1} ; it is said to be **implicit**. Other implicit methods may contain both A^n and A^{n+1} in the right-hand side term, just as both t_n and t_{n+1} could have been used. In the case of the Vlasov-Poisson simulated by GKMUST, the time parameter specifies what time's distribution function is used to compute the electric potential.

It is clear that implicit methods require considerably more computation time than explicit ones, even in the linear case (a linear differential equation will yield a linear equation to solve at each step). However, for some problems which are said to be stiff, explicit methods may require extremely small time steps to prevent the error from diverging while implicit methods usually do not suffer from such strict limitations. To avoid the divergence of the error, time integration schemes must be used within their stability conditions, usually found through the so-called **Courant-Friedrichs-Lewy (CFL) condition**, whose calculation is explained in section 3.2.3. The simulation of stiff problems can thus be much faster with an implicit method using a large time step, compared to an explicit method with a prohibitively small one.

The forth-order explicit Runge-Kutta method (RK4)

The Runge-Kutta methods are a class numerical methods whose schemes can be of any order of accuracy as well as both explicit or implicit. Amongst those

methods, the forth-order explicit scheme, often referred to as **RK4** is probably the most common, and it is the one used in GKMUST.

In this scheme, the calculation of A^{n+1} is done through 4 steps:

$$A^{n+1} = A^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (3.9)$$

where the 4 coefficients are given by

$$\begin{cases} k_1 = \mathcal{L}(t_n, A^n) \\ k_2 = \mathcal{L}\left(t_n + \frac{\Delta t}{2}, A^n + \frac{\Delta t}{2}k_1\right) \\ k_3 = \mathcal{L}\left(t_n + \frac{\Delta t}{2}, A^n + \frac{\Delta t}{2}k_2\right) \\ k_4 = \mathcal{L}(t_n + \Delta t, A^n + \Delta t k_3). \end{cases} \quad (3.10)$$

In the case of the Vlasov-Poisson system, where the field equation and Vlasov equation are solved separately, $\mathcal{L}$ depends on time through the electric field: $\mathcal{L}(t_n, \bullet)$ corresponds to a Vlasov equation whose electric field has been calculated using A^n . Evaluating $\mathcal{L}$ at different times means that both the Vlasov equation and the field equation are solved 4 times between each time step.

The reason for the choice of this method over the – explicit – forward Euler method is that in addition to accuracy, it also allows for larger time steps without diverging (although not as large as most implicit method would). A full stability analysis showing this can be found in section 3.2.3.

3.2 From slab to toroidal geometry

3.2.1 GKMUST and the several GKNETs

Despite major differences in the implementation, the general architecture of a gyrokinetic code does not vary much between a slab code and a toroidal one from a programming point of view. Fig. 3.1 shows a flow chart which applies to both GKMUST or early versions of GKNET, before the addition of features such as the variable time step.

The time integration (green box) is performed using an explicit 4-th order Runge-Kutta method in which a time step is decomposed into 4 elementary sub-steps (dark blue boxes) as explained above in section 3.1.3. An elementary sub-step consists in the resolution of the field equation to determine the gyro-averaged electric field $\langle E \rangle = \langle \nabla \Phi \rangle$ which can then be used to determine the coefficients of the Vlasov equation so the perturbation component of the ion distribution function can be advanced in time.

While the general organisation has not changed, the actual implementation of these elementary sub-steps has changed in important ways. Fig. 3.2 gives an

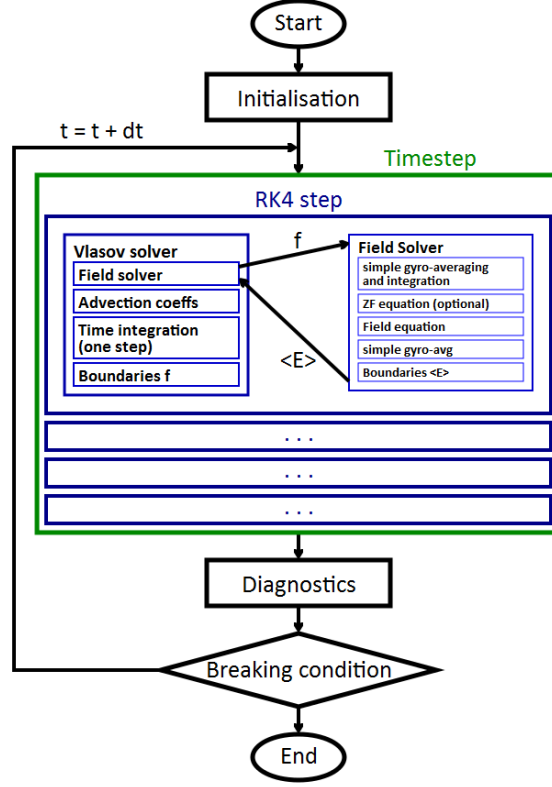


Figure 3.1: Flow chart of the initial structure of GKNET.

overview of some of the important upgrades along the way, as well as a general view of the various forking of the code. Indeed, slab geometry codes are by nature local codes, and for this reason, GKMUST relied on a δf linearisation for its distribution, and so did some later versions in circular configurations. Furthermore, while the computation of the gyro-averaged terms and subsequent resolution of the field equation relied on a Fourier approximation, this simplification requires particular geometries and boundary conditions: in a circular domain, a polar Fourier transform has trouble near the centre while a Cartesian would not be compatible with the boundary. The Fourier approach was thus convenient in the slab coordinates of GKMUST or in for versions with polar coordinates but exclude the area near the magnetic axis.

It is important to keep in mind that this thesis is based solely on only one of the branching in GKNET. Our aim being the full-f study of magnetic shaping effects, my own work is limited to the development and utilisation of the full-f code in arbitrary magnetic configurations. This version corresponds to the left branch on Fig. 3.2, the one enclosed in the red area.

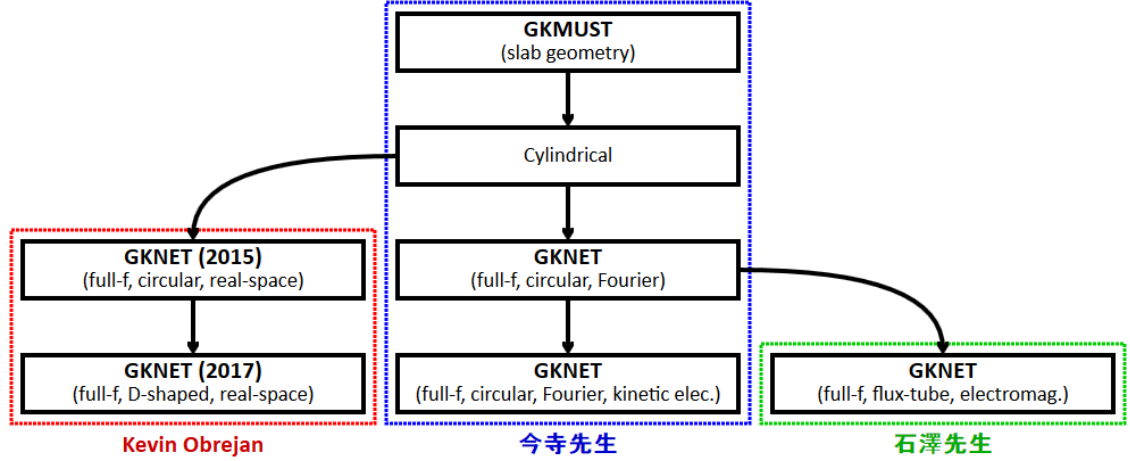


Figure 3.2: The history of the development of GKNET and its many intermediate steps and parallel versions.

3.2.2 Numerical grid and boundary conditions

From the programming point of view, an important change from GKMUST to the most recent version of GKNET is the change from polar (cylindrical) to Cartesian coordinates. In addition to the change in the coefficients of the equation system, this led to some more complicated changes regarding the numerical grid and parallelisation. Some of these complications are explained in later sections of this chapter, and some that would arise when tackling new problems are explained in later chapters, in particular chapter 4 with the geometric issues.

An important decision in the development of our version of GKNET was the use of a regular grid to ensure higher accuracy when computing spacial derivatives. Representing a general domain on a regular (R, Z) grid implies having part of the grid outside of the simulation domain. The boundary is assumed convex and its inside and outside are defined by the sign of the flux function ψ . During the early stages of the initialisation of the code, each line is scanned and the simulation domain is described using 2 arrays which specify for each line, the first and last indices of points inside. While all the arrays representing fields (electric fields, distribution, Vlasov equation coefficients, *etc*) contain values for all the grid points, inside or outside of the simulation domain, the ones outside are simply ignored in all loops and excluded from all computations.

Section 3.5 presents some of the consequences of this distinction between inside and outside points on the parallelisation of the code.

3.2.3 Variable time step

We mentioned in section 3.1.3 that when solving numerically a partial differential equation, the choice of integration schemes could influence how large a time step Δt could be used without seeing the error diverging. This stability condition is given by the CFL condition and depends on the integration scheme used for time as well as the one for all the other derivatives involved.

Simple case: CFL conditions of a 1D transport equation

To introduce the tools necessary for the study of the CFL conditions, let us consider for a moment a 1D transport equation for which $\mathcal{L}(A^n) = \nu \cdot \partial_x A^n$ where the transport speed ν is a constant. In section 3.1.3, we expressed the forward and backward Euler schemes as

$$A^{n+1} = (1 + \Delta t \cdot \mathcal{L})A^n \quad \text{and} \quad A^{n+1} = (1 - \Delta t \cdot \mathcal{L})^{-1}A^n. \quad (3.11)$$

Similarly, it can be easily verified using Taylor expansions that any N-th order explicit scheme leads to the general expression

$$A^{n+1} = \sum_{k=0}^q \frac{(\Delta t \mathcal{L})^k}{k!} A^n = P_q(\Delta t \mathcal{L}) A^n \quad (3.12)$$

where the k-th functional power of $\Delta t \mathcal{L}$ is not to be confused with the upperscript index A^n . The limit for $q \rightarrow \infty$ is then the familiar $A^{n+1} = \exp(\Delta t \mathcal{L}) A^n$.

It is clear that the error cannot diverge over time if the norm of the linear operator $P_q(\Delta t \mathcal{L})$ *i.e.* its largest eigenvalue in absolute value¹ is smaller or equal to 1 at each iteration. This constraint can be used to determine an upper bound for Δt (although this upper bound is not optimal). As any eigenvector V of $\mathcal{L}$ of eigenvalue λ , V will also be an eigenvalue of $P_q(\Delta t \mathcal{L})$ with the eigenvalue $P_q(\Delta t \lambda)$, the stability condition is equivalent to keeping the eigenvalues of $\Delta t \mathcal{L}$ inside a certain region of the complex plane. The left plot of figure 3.3 shows the contour $P_q(X) = 1$ the stability regions for the first 6 orders of the explicit Runge-Kutta methods.

In this simple bounded 1D case, ∂_x is fairly easy to diagonalise on $\mathbb{C}$ by using a complex Fourier transform as $\partial_x e^{ikx} = ike^{ikx}$. Applying the 5-points central

¹It is important at this point to keep in mind that the method is to be applied in a numerical context, and that therefore the $\mathcal{L}$ used here is assumed to be discretised in space. This means that the operators considered operate in finite dimension: the discrete and residual spectrum can be ignored, only the discrete spectrum needs to be considered.

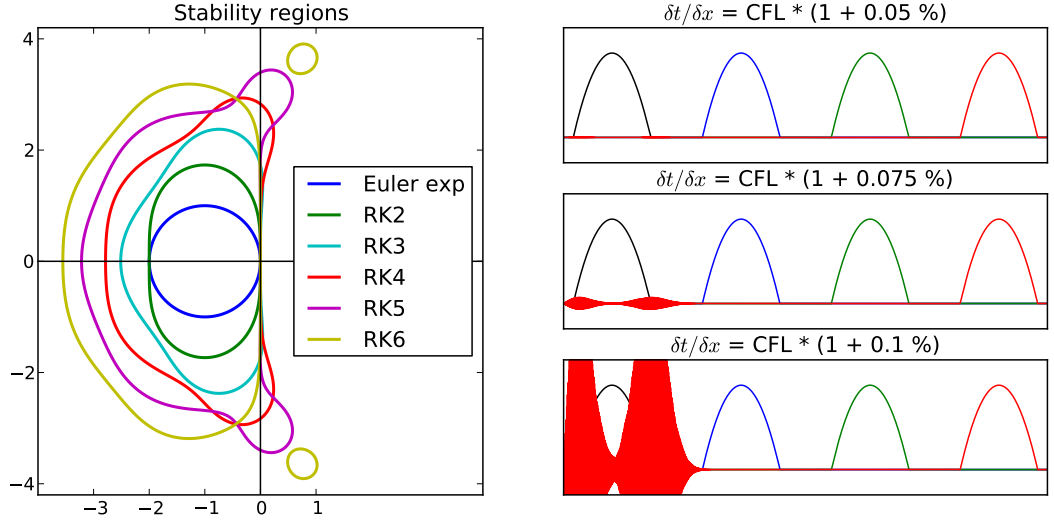


Figure 3.3: Left: stability regions for the eigenvalues of the first 6 explicit Runge-Kutta methods in the complex plane. Right: 4 time slices of the transport of a hump from left to right (black, blue, green and finally red) using an RK4 time integration and 5-points central differences for 3 different ratios $\Delta t / \Delta x$ slightly above the CFL conditions, causing the highest modes to grow exponentially near discontinuities (here C^1 -discontinuity).

differences Eq. (3.6) to this function:

$$\left[\frac{\partial}{\partial x} (e^{ikx}) \right]^{CD5} = \frac{1}{3\Delta x} \left[2(e^{ik\Delta x} - e^{-ik\Delta x}) - \frac{1}{4}(e^{2ik\Delta x} - e^{-2ik\Delta x}) \right] e^{ikx} \quad (3.13a)$$

$$\frac{1}{3\Delta x} (e^{ik\Delta x} - e^{-ik\Delta x}) \left(2 - \frac{1}{4}(e^{ik\Delta x} + e^{-ik\Delta x}) \right) e^{ikx} \quad (3.13b)$$

$$\frac{4i}{3\Delta x} \sin(k\Delta x) \left(1 - \frac{1}{4} \cos(k\Delta x) \right) e^{ikx}. \quad (3.13c)$$

For a fixed k , a simple Taylor expansion for $\Delta x \rightarrow 0$ allows to recover the result from the continuous case mentioned above. Applying $P_q(\Delta t \mathcal{L})$:

$$P_q(\Delta t \mathcal{L}) e^{ikx} = P_q \left(i \frac{4\nu \Delta t}{3\Delta x} \sin(k\Delta x) \left(1 - \frac{1}{4} \cos(k\Delta x) \right) \right) e^{ikx}. \quad (3.14)$$

Looking at the left plot of Fig. 3.3 once again, the intersection of the imaginary axis with the boundary of the stability region for RK4 can be evaluated to be roughly 2.828. Here, the largest possible wave number $k = \frac{2\pi}{2\Delta x}$ (corresponding to

an alternation of $+1 -1 +1 -1$ etc) forbids any sort of expansion of the sine and cosine but we can simply impose

$$\left| \frac{4\nu\Delta t}{3\Delta x} \sin(k\Delta x) \left(1 - \frac{1}{4} \cos(k\Delta x) \right) \right| \leq 1.029 \frac{4|\nu|\Delta t}{3\Delta x} \leq 2.828 \quad (3.15)$$

where the upper bound 1.029 is a numerical estimate. From here, a possible CFL condition is

$$\Delta t \frac{|\nu|}{\Delta x} \leq 2.061. \quad (3.16)$$

The right plot of Figure 3.3 shows a numerical verification of this CFL condition starting from an initial hump which is transported to the right (from black to red) slightly above the CFL conditions. This confirms that the CFL condition calculated is correct and that simulations that do not verify it see an exponentially growing high frequency error contaminate the data.

Applying linear tools to a nonlinear equation

Of course, this simple example is much more basic than the actual Vlasov-Maxwell system for several reasons, some of which are simple to deal with and some other more complicated. When the transport speed is not constant over space, stability at all point requires that the CFL condition be verified everywhere, and in particular for the largest transport speed $\|\nu\|_\infty$. This extends to all dimensions Z^1, Z^2 etc, if there is more than one, all contributions being simply summed and Eq. (3.16) is replaced by:

$$\Delta t \sum_i \frac{\|\nu_i\|_\infty}{\Delta Z^i} \leq 2.061. \quad (3.17)$$

It is important to notice here that in the case of non Cartesian coordinates, what we referred to as transport speed would include everything in front of the derivative being discretised. For instance, in the $\frac{1}{R} \frac{\partial}{\partial \varphi}$ of the ∇ operator, $\frac{1}{R}$ is to be included in the transport speed ν_φ .

Unfortunately, most tools for a stability analysis are mostly designed for linear problems and very little can be rigorously done when the discretised operator $\mathcal{L}$ is non-linear, as is the case with the Vlasov-Poisson system. When computing the elementary steps necessary to advance by one time step, $\mathcal{L}$ is evaluated for different distributions, leading to different electrostatic potential Φ , which appear in the transport speeds ν_i . Since the time step Δt must remain constant throughout the 4 elementary step that make up one time step, the initial choice must account for how larg $\|\nu\|_\infty$ can possibly grow.

A first intuition could be to use a safety margin and set Δt to, for instance, 75% of the value given by the CFL conditions Δt_{CFL} before integration. However the

margin of 25% is somewhat arbitrary and while it proved much more than necessary in most cases, running slower than necessary most of the simulation before failing a whole simulation because one particular time step required even more than that can be frustrating. A more advanced and more flexible strategy was thus devised, based on 2 key ideas: using a variable safety margin and saving the current distribution before integration so it can be retrieved and time rewound if the CFL conditions are no longer verified after the time integration. The algorithm can be summarised this way, denoting m the safety margin, which is initially set to 0.75:

1. store a copy of the current distribution,
2. compute Δt_{CFL} and set $\Delta t := m\Delta t_{CFL}$,
3. integrate the time step,
4. compute the new Δt_{CFL} , if $\Delta t_{CFL} < \Delta t$, set $m := 0.75\Delta t_{CFL}/(\Delta t/m)$, restore the saved distribution, revert to step 2,
5. advance time $t := t + \Delta t$, update the margin $m = (1 + 9m)/10$, repeat from step 1.

The formula used to update the safety margin, $m = (1 + 9m)/10$, increases it slowly, raising it from 0.75 to 0.99 in about 30 time steps, although in the linear phase (a quite "smooth" part of simulations), a few hundred steps can go without any CFL overflows being met. Not only does this method allow to fix CFL overloads when they occur, this slowly increasing variable margin allows a great gain in speed in such "smooth" periods compared to a fixed, large, margin.

3.3 Real space averaging method

3.3.1 Simple gyro-averaging

In [Lee, 1983], W. W. Lee introduced a method that allowed to compute the simple gyro-averaging in real space instead of using the more common Fourier representation. The gyro-averaging is essentially nothing more than the average over the particles' circular gyrations in real space. Despite the distinction, in earlier sections, of 2 simple-averagings denoted $\langle \cdot \rangle_{\mathbf{x}}$ and $\langle \cdot \rangle_{\mathbf{R}}$, this distinction is purely from the point of view of physics. Mathematically and numerically, the position coordinate $\mathbf{x}$ or $\mathbf{R}$ is the mute variable within an integral or a sum. Therefore the distinction will not be made in this section and the index on $\langle \cdot \rangle$ dropped.

The integral definition of the simple gyro-averaging, given by Eq. (2.55a), is a line integral whose path is defined by the Dirac $\delta(\mathbf{R} + \boldsymbol{\rho} - \mathbf{x})$. If we assume $\boldsymbol{\rho} = v_{\perp}/\Omega_s$

constant over the small scale of a Larmor radius, it is easy to see that the path of the integral is a circle of centre $\mathbf{R}$ and of radius ρ inside the plane perpendicular to the magnetic field line. Numerically, the averaging of a field $\langle \Phi \rangle$ at a point P is simply computed as the average of Φ over a given number of sample points M uniformly distributed on this circle (the green points on the black circle in Fig. 3.4). Later on, this set of points will be sometimes be referred to as **patterns**.

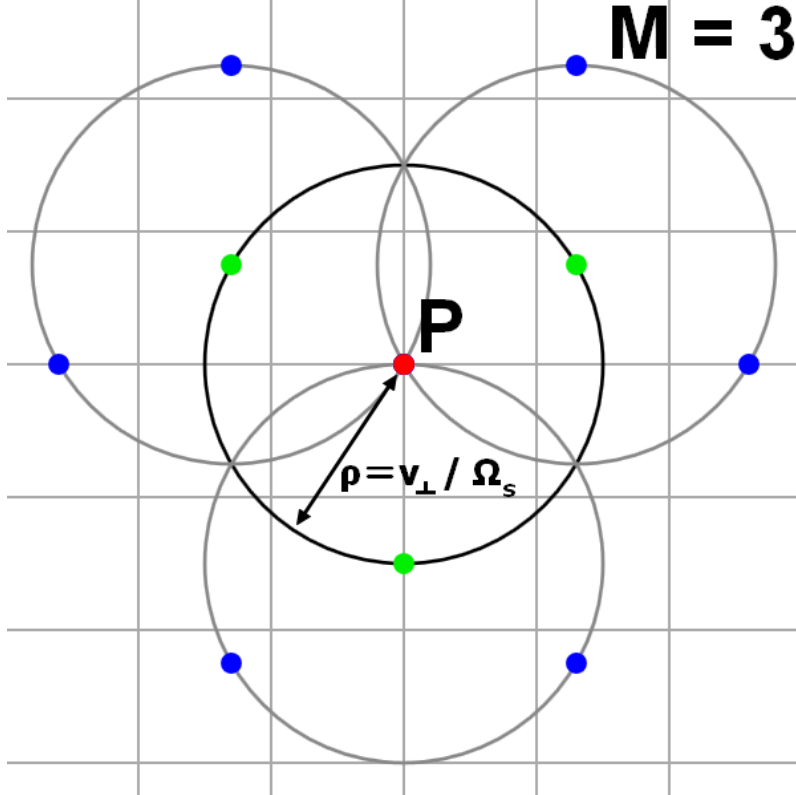


Figure 3.4: Primary (black) and secondary (grey) circles sampled to compute the gyro-averaging (green points) and double gyro-averaging (blue and red points) in real space for $N_c = 1$ radius and $M = 3$ points per circle.

The error of such a sampling can be estimated easily in Fourier space by assuming local Cartesian coordinates compared to Eq. (2.58). For a given mode $\mathbf{k}$, the real space averaging on M points, $\widehat{\langle \Phi \rangle}_{\mathbf{k}}^{RS}$, reads:

$$\widehat{\langle \Phi \rangle}_{\mathbf{k}}^{RS} = \widehat{\Phi}_{\mathbf{k}} \cdot \frac{1}{M} \sum_{m=0}^{M-1} e^{-ik_{\perp} \rho \cos(\vartheta_m)}, \quad \vartheta_m = 2\pi \frac{m}{M} \quad (3.18)$$

which, using a Jacobi-Anger expansion becomes

$$\begin{aligned}
 \widehat{\langle \Phi \rangle}_{\mathbf{k}}^{RS} / \widehat{\Phi}_{\mathbf{k}} &= \frac{1}{M} \sum_{m=0}^{M-1} \sum_{n=-\infty}^{\infty} J_n(k_{\perp} \rho) e^{-in\vartheta_m} \\
 &= \frac{1}{M} \sum_{n=-\infty}^{\infty} J_n(k_{\perp} \rho) \sum_{m=0}^{M-1} (e^{-2i\pi n/M})^m \\
 &= \sum_{n=-\infty}^{\infty} J_n(k_{\perp} \rho) \delta_{M|n} \\
 &= J_0(k_{\perp} \rho) \\
 &\quad + J_M(k_{\perp} \rho) + J_{-M}(k_{\perp} \rho) \\
 &\quad + J_{2M}(k_{\perp} \rho) + J_{-2M}(k_{\perp} \rho) + \dots
 \end{aligned} \tag{3.19}$$

where $\delta_{M|n} = 1$ if M is a divisor of n , 0 otherwise.

The leading term in J_0 is the expected result. Using $J_{-M} = (-1)^M J_M$, we can see that for small values of $k_{\perp} \rho$, the leading error term will be either $2J_M(k_{\perp} \rho)$ for M even or $2J_{2M}(k_{\perp} \rho)$ for M odd, which is therefore a much more favourable case despite the common use of 4 or 8 in PIC codes. As can be seen in Fig. 3.5, even for low number of points, the leading error term remains very small until J_0 reaches its first zero around $k_{\perp} \rho = 2.4$ which physically already corresponds to sufficiently small scales. However the approximation can be considered accurate even past this value as measure of the narrow intervals around J_0 's zeros can be neglected for integrations over ρ and the extremely unlikely event where a finite mode number would be exactly in those areas can be checked numerically before a simulation.

Since the points are unlikely to be aligned with the grid, the values are interpolated. This interpolation is the only part of the solver which depends on the choice of coordinates and can either be done in 2D (in the poloidal plane only) or in 3D, allowing to take into account the non perpendicularity of the magnetic field lines to the poloidal plane. This polynomial interpolation yields a numerical error which is also easy to estimate and control.

3.3.2 Double gyro-averaging

From simple to double gyro-averaging

Following the same idea, Lee's method was later extended to the double gyro-averaging by Z. Lin [Lin and Lee, 1995]. Instead of defining directly the double averaging $\langle\langle \Phi \rangle\rangle$ as given in eq. (2.55c), which is integrated over μ , let us first consider the simpler case of the single radius double averaging $\langle\langle \Phi \rangle\rangle^{\rho}$ (before integration). It is simply the composition of two simple averagings and can thus be performed similarly

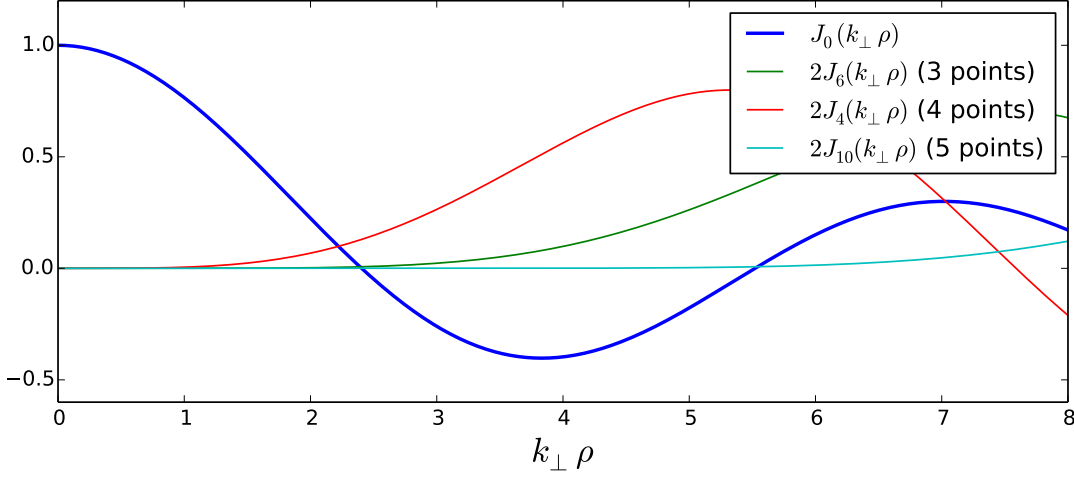


Figure 3.5: When using a local approximation, the gyro-averaging operator multiplies each Fourier mode by a Bessel function J_0 ; the approximation of this operator using evenly spaced sample points on a circular trajectory yields error terms which are small for low values of $k_{\perp}\rho$ compared to J_0 , even more so as the number of sample points increases. The top plot shows the expected result J_0 (blue) and the leading term of the error is shown for $M = 3$ points (green), $M = 4$ points (red), $M = 5$ points (cyan); the bottom plot shows the relative amplitude of the error term over the correct result.

by sampling over M^2 "secondary points" on "secondary" circular orbits centred on each "primary point" (the blue and red points on the grey circles in Fig. 3.4).

This single radius double averaging $\langle\langle\Phi\rangle\rangle^\rho$ needs to be multiplied by a Maxwellian and integrated over $\mu = m_s v_{\perp}^2 / (2B)$ to obtain the actual averaging $\langle\langle\Phi\rangle\rangle$ as it appears in the GK quasi-neutrality condition Eq. (2.64). Numerically, this integration is approximated by a weighted sum of $\langle\langle\Phi\rangle\rangle^\rho$ for C given radii:

$$\langle\langle\Phi\rangle\rangle = \sum_j c_j \langle\langle\Phi\rangle\rangle^{\alpha_j \rho_i}, \quad (3.20)$$

where the 2C coefficients $(c_j)_j$ and $(\alpha_j)_j$ now have to be adjusted. For this purpose, the Fourier approximation gives us:

$$\mathcal{F}(\langle\langle\Phi\rangle\rangle) \simeq \left(\sum_j c_j J_0(k_{\perp} \alpha_j \rho_i)^2 \right) \Phi \simeq \Gamma_0(k_{\perp}^2 \rho_i^2) \Phi. \quad (3.21)$$

Integration over μ

From this approximation, the coefficients could either be determined using a Taylor expansion or through the minimisation of the error in a given norm (mostly the L^2 and L^∞ norms). Lin's strategy is a mix of those two ideas, using a Taylor expansion at $u \equiv k_\perp \rho_i = 0$ to set 2 constraints which are then used in the minimisation of the error in L^2 norm. Using $\Gamma_0(u^2) \simeq 1/(1 - u^2) \simeq 1 + u^2$ and $J_0(u) \simeq 1 - u^2/4$, the constrained optimisation problem reads:

$$\text{minimise } \left\| \Gamma_0(u^2) - \sum_j c_j J_0(\alpha_j u)^2 \right\|_2^2 = \int_0^{u_{max}} \left(\Gamma_0(u^2) - \sum_j c_j J_0(\alpha_j u)^2 \right)^2 du \quad (3.22a)$$

$$\text{subject to } \begin{cases} \sum c_j = 1 \\ \frac{1}{2} \sum c_j \alpha_j^2 = 1 \end{cases} \quad (3.22b)$$

where the upper bound u_{max} depends on the largest mode number necessary relevant for the physics being simulated. The 2 constraints guarantee a high precision for low values of $k_\perp v_\perp$ while the usage of the optimisation problem extends the approximation up to $u_{max} = 2$ and the following values are given in the article:

- for $N_r = 1$ ring: $c = (1)$ and $\alpha = (\sqrt{2})$ (constraints only),
- for $N_r = 2$ rings: $c = (0.7194, 0.2806)$ and $\alpha = (0.9130, 2.2339)$,
- for $N_r = 3$ rings: $c = (0.3583, 0.5004, 0.1413)$ and $\alpha = (0.5443, 1.4142, 2.5138)$.

More rings of course allows for higher precision. However, with this objective function, increasing the number of rings usually leads to larger radii being included (3.67 for 5 points). Those larger radii can be an issue as, once implemented and parallelised, they lead to larger communications between processes, even more so as the double averaging doubles the final radius of the complete pattern. To avoid this, we used a slightly more complex objective function which combines the L^2 and L^∞ norms to compute our coefficients:

$$\frac{1}{\sqrt{u_0}} \left\| \Gamma_0(u^2) - \sum_j c_j J_0(\alpha_j u)^2 \right\|_2 + \frac{1}{N_r} \left\| \Gamma_0(u^2) - \sum_j c_j J_0(\alpha_j u)^2 \right\|_\infty \quad (3.23)$$

where both norms are of course taken on the same interval $[0, u_{max}] = [0, 2]$. The $1/N_r$ factor in front of the L^∞ norm is an arbitrary choice to better balance between error reduction and smaller circles. With this mixed objective function, we obtain higher accuracies:

3. DESIGN OF GKNET

	radii α	weights c	L^∞ norm	L^2 norm
Lin 3	0.5443 1.4142 2.5138	0.3583 0.5004 0.1413	$4.66 \cdot 10^{-3}$	$1.27 \cdot 10^{-3}$
Mixed 3	0.723442 1.695352 2.812279	0.521551 0.408559 0.069889	$4.03 \cdot 10^{-4}$	$2.59 \cdot 10^{-4}$
Mixed 4	0.621462 1.294250 1.941850 2.926465	0.392345 0.324042 0.234330 0.049282	$2.95 \cdot 10^{-4}$	$1.85 \cdot 10^{-4}$
Mixed 5	0.576634 1.121481 1.535199 1.988237 2.942365	0.332544 0.265425 0.159406 0.195693 0.046932	$2.79 \cdot 10^{-4}$	$1.77 \cdot 10^{-4}$

A python code is used to generate those coefficients. The exact result obtained fluctuates depending on the initial guess before the optimisation process but beyond 5 points, the gain in accuracy does not improve much more. Higher value are not used for simulations.

3.4 Field solver

3.4.1 Implementation of the gyro-averagings

The numerical implementation of the computation of the gyro-averagings in real space as presented above is quite direct and becomes very simple once the initial coefficients have been computed. Since the sample points are unlikely to be exactly on grid points, interpolations are required in addition to the method presented above. The patterns are contained in a plane perpendicular to the magnetic field $\mathbf{B}$ and are thus not exactly contained in a poloidal plane but form a small angle with it, making the interpolation not 2D but 3D.

Interpolation onto the grid

A simple but efficient interpolation method is the Lagrangian polynomials. For a function f being interpolated on a set of n points $x_1, x_2, \dots, x_n$, the interpolation reads:

$$f(x) \simeq \sum_{i \leq n} f(x_i) \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}, \quad (3.24)$$

which makes the polynomial of degree $n-1$ and thus the accuracy of the interpolation of order $n-1$ as well.

It is important to notice that during the initialisation of the code, it is the interpolation coordinate x that is known and not the value of f at the mesh points. What is actually pre-computed and stored is the coefficient in front of each $f(x_i)$, making the interpolation then become a simple weighted sum $f(x) = \sum w_i f(x_i)$. Once

integrated within the full gyro-averaging calculation, this makes the whole process become two nested sums (for each sample point and for each grid interpolation used to interpolate the value at that sample point) with a lot of overlap (neighbouring sample points will share some interpolation grid points). The advantage of this overlap will be explained in the next section.

This method is easily extended to 3 dimensions and can allow for different interpolation order along each dimension. Typically, interpolations along the 2 poloidal directions are of order 3 or 5 (an odd order means an even number of points, which is more natural for symmetry reasons around the cell) while along the toroidal direction the order is taken to 0 or 1 (as the points are usually very close to the central point's poloidal plane).

Storage of the coefficients and computation time

As the pattern's points are computed and interpolated on the grid during the initialisation of the simulation, they are stored as lists of mesh grid coordinates and weights and the actual computation of a gyro-averaging then becomes a mere weighted sum. Figure 3.6 shows how the coefficients of the double gyro-averaging are distributed over the grid in a dense block. Although the figure only represents 2 circles with 5 points per circle, the grid points used for the interpolation overlap heavily, lowering the number required for the computation: the $2 \cdot 5^2 = 50$ pattern points are computed using only 41 grid points. Those overlapping mesh points are detected during the initialisation and redundant points are merged by summing all the weights that correspond to them. Increasing the number of points per circle or adding new rings of smaller radius would also have no influence on the final number of grid points.

The important parameters which increase the number of grid points included and thus the computation time are the radius of the largest ring – or rather, its ratio to the mesh size – and the interpolation order. When the code is parallelised, those 2 parameters also increase the amount of data that needs to be shared between processes managing neighbouring chunks of the mesh. However, in practice, this method gives good results even when the cells of the grid are quite large compared to the pattern: benchmark simulations are routinely performed with mesh size of around 4 and largest radius around 2.5 (marking the pattern of total width 10).

Aside from computation time, storing tens of triplets of coordinates and weights for each grid point could be an issue in some situations. Section 3.5 will introduce ideas and tricks to exploit the lower dimensionality of the gyro-averagings (4D for the simple gyro-averaging and 3D for the double gyro-averaging) thus reducing memory consumption. Additionally, as already mentioned, our simulations assume the magnetic field to be independent of the toroidal angle, thus allowing to store only one pattern per poloidal grid point and greatly reducing memory requirements.

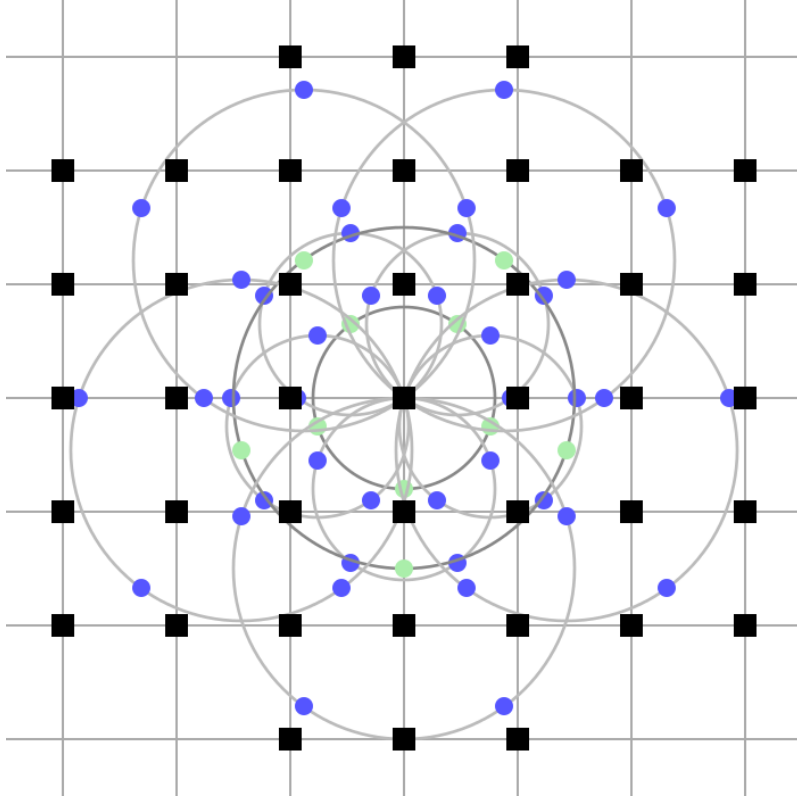


Figure 3.6: Grid points used to compute a double gyro-averaging (black squares) with $N_c = 2$ radii and $M = 5$ points per circle, using a first order interpolation (on 2 points in each direction).

However, for implementations that either do not assume such an independence or are to be integrated within an electromagnetic code (where $\mathbf{B}$ evolves with time), additional simplifications might be needed. For instance, if $\mathbf{B}$ is approximated as perfectly perpendicular to the poloidal field, the position of the patterns' points relative to the centre only depend on its norm B (through $\rho_i = v_{th,i}/\Omega_i$); it is thus possible to pre-compute patterns for a discrete set of possible values of B and then choose on the fly the pattern whose associated value is the closest to the actual value of B at the point considered. Such a method could greatly reduce memory consumption while barely increasing the computation time.

3.4.2 Fixed-point method

For some types of simulations, such as linear ITG growth rate simulations, the magnetic flux surface averaged term $\bar{\Phi}$ of the field equation Eq. (2.64) can be ignored

as the ZF does not play an important role until the saturation phase is reached. This fact is important, this flux surface averaged term, while trivial to compute in polar coordinates, requires advanced techniques in Cartesian coordinates (even in the circular case). When included, this term is computed before solving the bulk of the field equation, using methods that are explained in the next section and in the next chapter, and added to the integral term on the right-hand side term. The very first versions of GKNET with Cartesian coordinates did not include this term and the ZF solver was only added a bit later.

Fixed-point form of the field equation

For now, we will thus consider a simpler field equation with no magnetic flux surface averaged term and with a general right-hand side simply denoted C , which may or not contain $\bar{\Phi}$:

$$(1 + \tau)\Phi - \langle\langle\Phi\rangle\rangle = C, \quad (3.25)$$

where $\tau \equiv T_e/T_i$. In the general magnetic configuration case, where Fourier transforms are not available, this equation is merely a 3D linear problem. Instead of either finding a linear algebra library that would fit in the rest of the architecture of the code or implement a parallelised linear solver.

However, instead of using a general solving method, GKNET's field solver exploits the specificity of this particular equation. Let us first rewrite Eq. (3.25) under the form:

$$C + (M_{DAVG} - \tau)\Phi = \Phi, \quad (3.26)$$

with M_{DAVG} the linear application or matrix for the double gyro-averaging and where the identity application has been omitted for concision. This form is a so-called fixed-point form, as, starting from a given value Φ_0 and iterating $C + (M_{DAVG} - \tau)\Phi_n = \Phi_{n+1}$, it will converge to a solution of the equation as long as the application on the left-hand side is a contraction (*i.e.* its Lipschitz constant is strictly inferior to 1). In the present case of an affine application, this condition is equivalent to having the eigenvalues of its linear part $M_{DAVG} - \tau$ smaller than 1 in absolute value (*i.e.* $\text{Sp}(M_{DAVG} - \tau) \subset]-1, 1[$). Denoting λ the largest eigenvalue in absolute value, it is easy to see that $\|\Phi_n - \Phi_\infty\| < \lambda^n \|\Phi_0 - \Phi_\infty\|$, which means that λ is the convergence speed of the fixed-point method.

Looking at the Fourier representation of the double gyro-averaging, one can see that the eigenvalues of $\langle\langle\Phi\rangle\rangle$ should roughly range from $\Gamma_0(0) = 1$ down to a small positive number $0 < \Gamma_0(k_{\perp, \max} \rho_i) \ll 1$ associated with the highest wave number available on the mesh. Furthermore, although the assumption that $\tau = 1$ is very common for simulations with only one ion species, we can be more general and simply require of it to be strictly positive. Combining these two observations, we immediately see that the linear part of Eq. (3.26) will have its eigenvalues within

the semi-open interval $[1 - \tau, -\tau[$, making the affine application on the left-hand side a contraction and thus a good candidate for a fixed-point method as long as $\tau < 1$.

Speeding up the convergence

To improve the convergence properties of fixed-point iterations, we can add $(\gamma - 1)\Phi$ on both side of Eq. (3.26), for some $\gamma \in \mathbb{R}$, leading to the final form:

$$\frac{1}{\gamma} (C + (M_{DAVG} + \gamma - 1 - \tau)\Phi) = \Phi. \quad (3.27)$$

Under this form, the spectrum of the linear part of the equation will be included in the interval $\mathfrak{I} = \left[1 - \frac{1+\tau}{\gamma}, 1 - \frac{\tau}{\gamma}\right]$ which is of width $1/\gamma$ and centred on $1 - (1/2 + \tau)/\gamma$. Intuitively, we could choose $\gamma = \tau + \frac{1}{2}$ as to centre the interval on 0, thus transforming it into $[-1/(2\tau + 1), 1/(2\tau + 1)[$, guaranteeing the convergence of the fixed-point method as soon as $\tau > 0$, with convergence speed $1/(2\tau + 1)$. Figure 3.7 shows this interval $\mathfrak{I}$ as a function of γ , for 3 different values of τ , showing in particular that low values of τ lead to a very slow convergence, which is a real issue when using kinetic electrons (a case equivalent to $\tau = 0$).

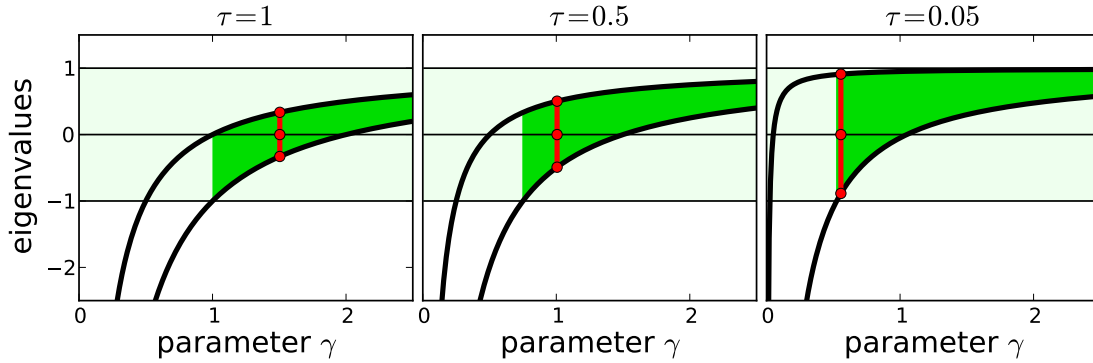


Figure 3.7: Minimum and maximum eigenvalues (thick black lines) of the function to which the fixed-point method is applied, as a function of the parameter γ , and for 3 different values of τ . The green area corresponds to the conditions where the fixed-point method would indeed converge, with the red line showing the optimal value of γ , allowing for the fastest convergence.

However, this theoretical consideration does not properly take into account the fact that the smallest eigenvalue of M_{DAVG} is not 0 but a number difficult to estimate in general situations, nor the weights which are ignored outside of the boundary. For

these reasons, in the code, we rather choose to compute numerically an optimal value of γ during the initialisation of the solver, which is both quite simple and fast. For this purpose, instead of considering eigenvalues directly, we will use the natural inner product on the matrix space $\mathbb{R}^{n \times n}$:

$$\langle \cdot, \cdot \rangle : (A, B) \mapsto \text{tr}(AB^T) \quad (3.28)$$

and its associated norm $\|\cdot\| : A \mapsto \sqrt{\langle A, A \rangle}$. Not using an eigenvalue based norm is acceptable since all norms are equivalent in finite dimension (and this particular one equals n^2 times the largest eigenvalue in the diagonalisable case). Then, considering only the linear part as the constant part does not play a role in the minimisation of the norm, we can decompose

$$\begin{aligned} \left\| \frac{1}{\gamma}(M_{DAVG} + \gamma - 1 - \tau) \right\|^2 &= \frac{1}{\gamma^2} \|M_{DAVG} - 1 - \tau\|^2 \\ &\quad + \frac{2}{\gamma} \langle M_{DAVG} - 1 - \tau, 1 \rangle + n^2, \end{aligned} \quad (3.29)$$

where $\|1\|^2$ was replaced by its value n^2 . Therefore:

$$\begin{aligned} \frac{\partial}{\partial \gamma} \left(\left\| \frac{1}{\gamma}(M_{DAVG} + \gamma - 1 - \tau) \right\|^2 \right) &= 0 \\ \Leftrightarrow \frac{-2}{\gamma^3} \|M_{DAVG} - 1 - \tau\|^2 - \frac{2}{\gamma^2} \langle M_{DAVG} - 1 - \tau, 1 \rangle &= 0 \\ \Leftrightarrow \gamma &= \frac{-\|M_{DAVG} - 1 - \tau\|^2}{\langle M_{DAVG} - 1 - \tau, 1 \rangle}. \end{aligned} \quad (3.30)$$

In the context of a discretised space, as is the case within a simulation code, the denominator can be simplified into $\frac{1}{N} \text{tr}(M_{DAVG}) - 1 - \tau$ with N the number of grid points. Using our real space representation of the double gyro-averaging, $\frac{1}{N} \text{tr}(M_{DAVG})$ would then correspond to the average weight of the central point.

The numerical value of γ for $\tau = 1$ is usually found to be around 2.1, with a tendency to decrease as the resolution increases. This could be expected as the sample points outside the boundaries have their weight artificially set to 0, reducing the (positive) eigenvalues of M_{DAVG} , making $\|M_{DAVG} - 1 - \tau\|^2$ grow larger, while the denominator is not affected by extra-diagonal points. With higher resolutions or higher interpolation orders, more exterior points are ignored, slightly increasing the numerical γ .

The stopping criterion of the fixed-point method is usually taken as $\|\Phi_{n+1} - \Phi_n\| < 5 \cdot 10^{-4} \|C\|$. Despite this strict criterion, the method was found to almost always converge in less than 5 iterations for $\tau = 1$.

3.5 Parallelisation

GKNET inherited the overall parallelisation architecture of GKMUST: a "hard-wired" parallelisation in the μ dimension (each core can only contain a single value of μ , hence the number of cores must be at least the same as the number of mesh points in this direction), and a 3-dimensional parallelisation in position space. However, many changes and improvements were later made as the code evolved, with, in particular, the extension of the parallelisation in the fifth-dimension $v_{||}$. Some more advanced methods are explained below. Notice that we will always assume the number of processes available to be a power of 2 for simplicity (and optimal use of the super-computer's architecture).

3.5.1 Cross-section shape in Cartesian coordinates

While the slab geometry of GKMUST or the circular domain in polar coordinates of early versions of GKNET were easy to split evenly between processes, the task is more complicated with arbitrary cross-section shapes and Cartesian coordinates. As the shape of the cross-section is assumed to be the same for all toroidal planes (in tokamaks), let us only consider a single cross-section to be parallelised for the rest of this section, ignoring the other directions.

As explained in Sec. 3.2.2, the shape of the domain in the cross-section is taken into account by considering a regular rectangular mesh, whose points outside of the simulation domain are simply excluded from computation. An obvious consequence of this approach is that the mesh is split horizontally into slices of identical height, the number of simulated grid point may vary greatly (with much fewer simulated points in the slices at the top and bottom than in the ones near the magnetic axis), thus leading to a lot of idle time for the most sparse slices.

Ideally, the grid points of the cross-section would be split evenly between processes with no respect for columns and lines, however this would make the data to be shared between processes much more complicated to define. As triangularity introduces a left-right asymmetry, the current algorithm does not subdivide the cross-section horizontally but only cuts it in horizontal slices of variable height to keep the exchange of shared data simple enough. Notice that this is a strategical choice and not a technical limitation as the code since the MPI architecture of the code treats x the same as the other dimensions. Furthermore, a given slice can only share data with its direct neighbours (above and below), making the minimum height of a slices be equal to either the 2 ghost cells used for the spacial derivatives or half the size of simple or double gyro-averaging pattern. Assuming a convex simulation domain, the height of the slices is then determined as follows:

1. from the top, set all slices to their minimum height, except for the last one,

2. from the top, going downwards, push each boundary down (if possible given the minimum height) if and as long as it reduces the number of cells in the largest of the 2 neighbouring slices,
3. from the bottom, going upwards, do the same but moving boundaries upward,
4. if steps 2 or 3 did find a boundary to move repeat from step 2.

This simple method leads to distributions such as the one shown on Fig. 3.8, where an elliptic domain of elongation $\kappa = 1.4$ is shown split between 8 horizontal slices.

Although this approach sets a maximum to the number of cores which can be allocated to the parallelisation of the cross-section, it is unsure that a finer parallelisation would be beneficial, as communications would take much longer. Furthermore, with 3 other dimensions to allocate cores in, this limit has never been an issue so far.

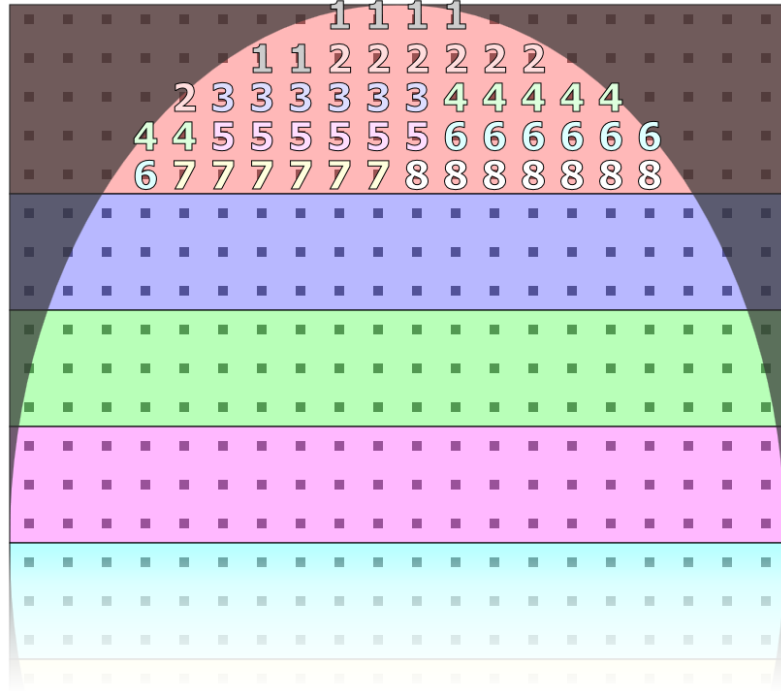


Figure 3.8: Representation of an elliptic cross-section of elongation 1.4 whose grid points have been spread between 8 horizontal slices (only 6 coloured areas shown). The first 4 slices are of heights: 52, 48, 58 and 60. The digits from 1 to 8 in the first area correspond to a sub-division of the mesh points into 8 blocks of uniform sizes (6 or 7) for the sub-parallelisation in μ and $v_{\parallel}$ of the gyro-averagings.

3.5.2 Sub-parallelisation μ and $v_{\parallel}$ of the gyro-averagings

While the time integration of the Vlasov equation is a 5-dimensional problem which has thus been parallelised accordingly, the resolution of the field equation and the computation of the gyro-averagings are performed in lower dimensionality. The simple averagings $\langle \cdot \rangle_x$ and $\langle \cdot \rangle_R$ depend on μ through their radius but not on $v_{\parallel}$ – and $\iint \langle \cdot \rangle d\mu dv_{\parallel} = \int \langle \int \cdot dv_{\parallel} \rangle d\mu$, ensuring that the object being gyro-averaged does not depend on $v_{\parallel}$ – making it 4-dimensional. The double gyro-averaging already contains integrations over μ and $v_{\parallel}$ in this definition, making it 3-dimensional. This reduction of dimensionality allows for great improvements of the parallelisation of those parts of the code.

The previous section mentioned constraints on the sub-division of the mesh in the cross-section through its shape (horizontal slices) and size (the minimum height of each slice). In the case of a reduction of dimensionality, the problem does not apply: the data is already available and redundant.

Figure 3.8 shows the sub-division of the light red data block into 8 smaller blocks represented as numbers from 1 to 8. If we consider, for instance, the simple gyro-averaging performed on data that has been integrated in $v_{\parallel}$ before the resolution of the field equation, each numbered block would correspond to the data to be computed by one of 8 cores allocated in the $v_{\parallel}$ dimension; as the initial data is the same for all of those cores, some might even be able to perform this computation without even having to request for neighbouring data to be shared prior to computation. Similarly, the double gyro-averaging and iterations of the fixed-point method are 3-dimensional, allowing to sub-divide data blocks into even smaller chunks.

3.5.3 Allocation of available processes

During the early stages of the initialisation phase, the code sub-divides the 5 dimensional grid and allocates processes to it. The processes are split into N_u groups to each receive a single mesh point in the μ dimensions. A strategy is then required to decide how to subdivide the remaining 3 dimensions: y , z and v , as the efficiency of the parallelisation will depend greatly on this choice.

An initial parallelisation strategy consisted in first subdividing as much as possible the z dimension, then if necessary continue with the y dimension, and finally the v dimension. The reason for this ordering comes from operations performed within the field solver. Spatial dimensions (y and z) offer a reduction of the data held and shared by cores even in the field solver where dimensionality is reduced (see below), and among them, the priority to z comes from the fact that the gyro-averagings are usually computed within poloidal planes, making the sharing of data along z direction unnecessary.

However, the 5-points stencil finite differences used when integrating the Vlasov equation require on either end of each dimension. As the parallelisation of a given dimension increases and the number of computed cells per process decreases, the communications needed to update those ghost cells weights more and more, until initially less optimal dimensions become better candidates. With this in mind, a new parallelisation strategy based on the number of computed cells in each dimensions was devised. Defining the "priority" of a dimension as the number of (computed) cells, multiplied by 4 for z and by 2 for y , the strategy is as follows: double the number of processes allocated to the dimension with highest priority (in case of a draw, z wins over y which wins over v), update priorities and repeat until all cores have been allocated. The factors 2 and 4 are used to take into account our previous considerations on the advantages of z over y and y over v , although their exact value is somewhat arbitrary. To illustrate the strategy, let us consider the allocation 4096 cores (remaining after mu parallelisation) of a grid $N_z = 128$, $N_y = 128$, $N_v = 64$ as an example:

# proc.	# proc. in z (priority)	# proc. in y (priority)	# proc. in v (priority)
1	1 (512)	1 (256)	1 (64)
2	2 (256)	1 (256)	1 (64)
4	4 (128)	1 (256)	1 (64)
8	4 (128)	2 (128)	1 (64)
16	8 (64)	2 (128)	1 (64)
32	8 (64)	4 (64)	1 (64)
64	16 (32)	4 (64)	1 (64)
128	16 (32)	8 (32)	1 (64)
256	16 (32)	8 (32)	2 (32)
512	32 (16)	8 (32)	2 (32)
1024	32 (16)	16 (16)	2 (32)
2048	32 (16)	16 (16)	4 (16)
4096	64 (8)	16 (16)	4 (16)

where the green cells mark to the dimensions of highest priority, a darker tone being added to the one that is next in line to receive additional parallelisation. This improved strategy allowed for great improvements of the parallelisation efficiency, although we do not provide statistics here.

3.6 Summary

Starting from a simple slab gyrokinetic simulating code, we have newly developed the electrostatic toroidal full-f gyrokinetic code with adiabatic electrons GKNET.

Its Cartesian grid in the poloidal cross-section and its uncommon field solver, which computes the gyro-averaged and polarisation term by sampling the particle's trajectories in real space, allow both a proper treatment of full-FLR effects and arbitrary magnetic configurations, in particular D-shapes. GKNET is parallelised along 4 of the 5 dimensions of the system using MPI, with some advanced techniques to subdivide some parts of the grid in order to take advantage of all the cores available even when the dimensionality is reduced to 4 or even 3, in the field solver.

With the design and properties of GKNET explained, the study of shaping effects can now be introduced. It begins, in the next chapter, with the case of Zonal Flow dynamics in D-shaped domains.

Chapter 4

Zonal Flow dynamics in D-shaped configurations

Contents

4.1	Simulating zonal flow dynamics in non circular domains	72
4.1.1	Constructing a flux function	73
4.1.2	Parametrisation of the flux surfaces	73
4.1.3	Zonal Flow solver by projection	75
4.2	Diagonalisation of the Zonal Flow equation	76
4.2.1	Eigenfunction equation	76
4.2.2	Eigenfunctions and eigenvalues	78
4.2.3	Correspondence to the Fourier description	79
4.3	Zonal Flow solver by diagonalisation	80
4.3.1	Diagonalisation based solver	80
4.3.2	Comparative performances	82
4.4	Shaping effects	82
4.4.1	Influence of elongation	83
4.4.2	Influence of triangularity	85
4.5	Summary	87

Zonal Flow (ZF) is the magnetic flux surface averaged component of the electrostatic field and is known to play an important role in regulating turbulence such as the ITG or TEM instabilities and governing the transport level. ZF's lowest modes grow quickly as soon as a charge discrepancy appears between flux surfaces and form large scale structures. However, this growth is mitigated by mechanisms such as the collisionless Landau damping which cause it to oscillate and damp its amplitude towards a non-zero residual level. Despite its inherent multi-scale level, the influence of the magnetic configuration's shape on this phenomenon have been studied very little, and always with local theory or local simulation codes.

The Cartesian coordinates used in the code, while useful for general shapes and to loosen the CFL conditions, are not well adapted to describe the D-shaped magnetic flux surfaces, in particular near the magnetic axis. The change of coordinate to a radial description requires high mesh numbers to converge to accurate results. Since most of the dynamics of the GAM damping is dominated by the resolution of the ZF equation, a better approach that takes better into account the shape of the magnetic flux surfaces is capital.

This chapter thus introduces an innovative mathematical method to solve the so-called ZF equation using a diagonalisation of its differential operator which allows to better take into account the flux-surfaces' intrinsic geometric properties. This new tool is then used to study numerically the influence of magnetic shaping effects on ZF damping through elongation and triangularity.

4.1 Simulating zonal flow dynamics in non circular domains

Flux-surfaces and magnetic field are 2 complementary conceptualisations of one thing. However, from the standpoint of a numerical physicist, the flux function is what defines the magnetic field and the geometry of the problem. Many simulation codes define their boundary using a parametrisation introduced by Miller in [Miller et al., 1998] in 1998:

$$\begin{cases} R = R_0(\alpha) + \alpha \cos(\theta + (\sin^{-1}(\delta(\alpha))) \sin \theta) \\ Z = \alpha \kappa(\alpha) \sin(\theta) \end{cases} \quad (4.1)$$

where α is a linear labelling of flux surfaces (used instead of r to avoid confusion in non-circular cases) ranging from 0 to the minor radius a_0 , κ is the elongation and δ is the triangularity. However no theoretical expressions of $R_0(\alpha)$, $\kappa(\alpha)$ and $\delta(\alpha)$ are given in the original publication and assumptions on those seem to vary depending on authors (in particular since they often only consider the plasma edge and need not consider the variation of the parameters within). To avoid confusion, we derive

below an explicit parametrisation, which is compatible with Miller's but where the shaping parameters (the elongation κ , the triangularity δ and the Shafranov shift Δ) are constant.

4.1.1 Constructing a flux function

Formally, the flux function ψ is the solution of the Grad-Shafranov equation for a given pressure profile $p(\psi)$ and toroidal component of the magnetic field $F(\psi)$. It thus defines both the flux-surfaces as its surface of iso-values $\{\psi = \text{const.}\}$ and the radial variation of ψ across the flux-surfaces. However, no exact solution of the Grad-Shafranov exists in general configurations and when designing a simulation code, it is often preferred setting separately $F(\psi)$, the flux-surfaces and their radial profile following approximations. This artificial flux function might be the solution of "a" Grad-Shafranov for a particular choice of $p(\psi)$ or it simply might be unrealistic.

As gyrokinetics does not consider pressure explicitly in its equation, this approximation is commonly regarded as acceptable as long as the chosen flux-surfaces and radial profile, taken separately, make sense physically. The Miller parametrisation mentioned in the introduction of this section is derived as an approximation of a solution of the Grad-Shafranov equation. Our own parametrisation, which is introduced below, is a approximation of any 2D function given the geometric expectations we have on the general form of the flux function (upside-down symmetry and well-shaped).

The safety factor q is defined as the average of $\frac{\alpha \cdot B_p}{R \cdot B_\phi}$ on a flux surface. Given the general of the magnetic field $\mathbf{B} = \nabla\psi \times \nabla\varphi + F(\psi)\nabla\varphi$ and our choice of $F(\psi) = B_0 R_0$, the safety factor reads:

$$q = \iint \frac{\alpha \cdot B_p}{R \cdot B_\phi} dS = \iint \frac{r B_0 R_0}{R \|\nabla\psi\|} dS \quad (4.2)$$

where the integral is performed on a flux-surface *i.e.* a surface where ψ is constant. This expression can be used to set a radial $\psi(\alpha)$ given a $q(r)$ profile.

4.1.2 Parametrisation of the flux surfaces

The magnetic flux surfaces correspond to the iso-surfaces of the flux function ψ . They can be parametrised by expanding ψ around its extremum ψ_0 near the magnetic axis at $(R, Z) = (R_0, 0)$. In the case of a magnetic configuration with up-down symmetry, using the coordinates (x, y) where $(R, Z) = (R_0(1+x), R_0 y)$, only the even terms in y remain:

$$\frac{\psi - \psi_0}{N_\psi} = 2Ax^3 + x^2 + Bx + 2Cy^2x + Dy^2 + O(\varepsilon^4). \quad (4.3)$$

with N_ψ a normalising term. Given the expected general aspect of ψ (well-shaped), it is clear that D and $(\psi - \psi_0)/N_\psi$ must be positive (or we might define a saddle or nothing at all). Rewriting this last term as $(\psi - \psi_0)/N_\psi = p^2$, a factorisation can be forced:

$$p^2 = (Ax^2 + x + Cy^2)^2 + E^2y^2 + O(\varepsilon^4) \quad (4.4)$$

where the additional terms of this imperfect factorisation are all contained inside $O(\varepsilon^4) = O((x + y)^4)$. From here, if we ignore the asymptotic term, it is easy to see that there exist a θ such that:

$$\begin{cases} Ax^2 + x + Cy^2 = p \cdot \cos \theta \\ \sqrt{D}y = p \cdot \sin \theta. \end{cases} \quad (4.5)$$

The simpler expression for y allows to identify $D = 1/\kappa^2$ and see that p runs from 0 to ε . Replacing y by its expression in x :

$$x = \frac{1}{2A} \left(-1 \pm \sqrt{1 - 4A(Cp^2\kappa^2 \sin^2 \theta - p \cos \theta)} \right). \quad (4.6)$$

Keeping only the solution closest to 0 (the + of $\pm$), an expansion gives us

$$\begin{aligned} x = & p \cos \theta - Cp^2\kappa^2 \sin^2 \theta \\ & - A(p^2 \cos^2 \theta - 2Cp^3\kappa^2 \cos \theta \sin^2 \theta) \\ & + 2A^2p^3 \cos^3 \theta + O(p^4). \end{aligned} \quad (4.7)$$

Therefore, denoting $p = \alpha/R_0$ and identifying the remaining coefficients $A = \Delta/\varepsilon$ and $C = (\delta + \Delta)/(\varepsilon\kappa^2)$, with a bi of trigonometry, we finally obtain the second order accurate parametrisation

$$\begin{cases} R = R_0 + \alpha \cos(\theta) - \frac{\alpha^2}{a_0}(\Delta + \delta \sin(\theta)^2) \\ Z = \alpha\kappa \sin(\theta) \end{cases} \quad (4.8)$$

for which the next order term is a $2\Delta\frac{\alpha^3}{a_0^2}(\Delta + \delta \sin^2 \theta) \cos \theta$ to be added the parametrisation of R . The parametrisation this becomes accurate to the third order when $\Delta = 0$. Figure 4.1 shows an example of flux surfaces obtained with this parametrisation.

Our own numerical tests using Solov'ev solutions of the Grad-Shafranov equation exhibit a very good agreement with this parametrisation of the flux surfaces. Additionally, despite the apparent differences with the parametrisation introduced by Miller, the two parametrisations are actually very close. According to Miller's own remarks from numerical simulations: $\delta(\alpha) \propto \alpha$, $\partial_\alpha R_0 \propto \alpha$ and $\partial_\alpha \kappa$ is usually found to be very small.

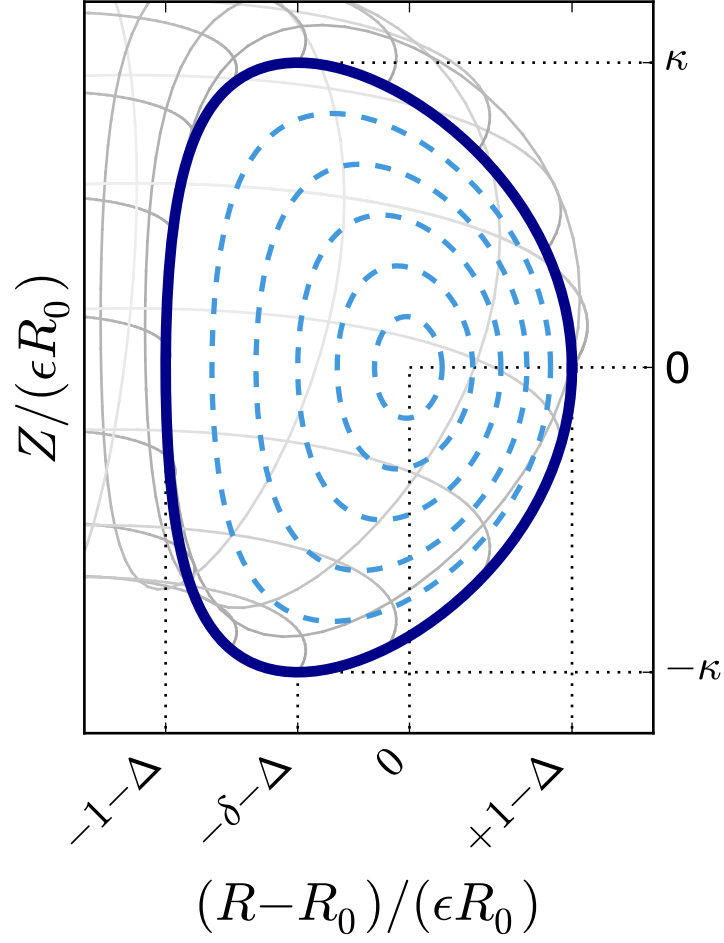


Figure 4.1: Geometry of the magnetic flux surfaces in the poloidal section and definition of the major axis R_0 , inverse aspect ratio ε , elongation κ , triangularity δ and Shafranov shift Δ .

4.1.3 Zonal Flow solver by projection

The first versions of GKNET used a projections along the magnetic flux surfaces onto a 1D radial mesh (that can be interpreted as the segment $R_0 \leq R \leq (1 + \varepsilon)R_0$, $Z = 0$). As the projection of the 3D points of the Cartesian grid do not form a regular mesh, the projected data is fitted using least squares (see left column of Fig. 4.2). The construction of the matrix for the 1D problem given by Eq. (2.65) is performed by projecting the double averaging trajectories centred on several points on each magnetic flux surface in a similar way (see right column of Fig. 4.2).

This so-called projection solver was later complemented by the diagonalisation

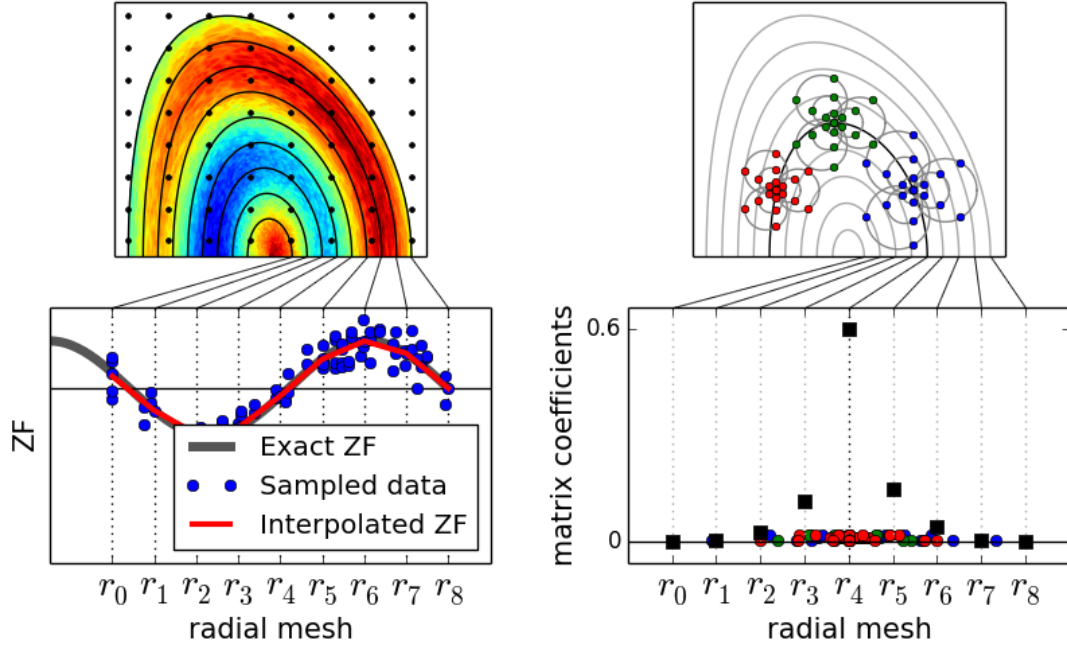


Figure 4.2: Left: projection of the 3D data onto the radial mesh line $R > R_0$, $Z = 0$ and re-sampling using least squares; the black dots on the left figure represent the Cartesian poloidal mesh while the solid black lines correspond magnetic flux surfaces associated with each radial mesh point. Right: computation of the coefficients of the matrix for the magnetic flux averaged double gyro-averaging $\langle\langle \cdot \rangle\rangle$ by computing the real space trajectories for the double averaging (3 points per circle, 3 radii, the sizes have been exaggerated here for visibility) at several points of the magnetic flux surface and projecting them on the radial mesh (coloured points on the bottom plot) and their binned sum (black points); the matrix coefficients are the ones for the line associated to the radial mesh point r_4 .

solver presented in the next chapter. Both versions are still available as the projection solver is more general, despite requiring higher resolutions to obtain results of the same quality in simulations where the diagonalisation solver can be used.

4.2 Diagonalisation of the Zonal Flow equation

4.2.1 Eigenfunction equation

Using the idea of the sampling over secondary circular orbits or, more formally, starting from the integral definition given by Eq. (2.55c), the double gyro-averaging

of a given field g at a point (R, Z) (here g is an arbitrary field independently of any physical meaning) corresponds to:

$$\langle\langle g \rangle\rangle(R, Z) = \int_0^\infty \frac{1}{4\pi^2} \iint g(R + \varepsilon_R, Z + \varepsilon_Z) F_M(R, Z) d\theta_1 d\theta_2 d\rho \quad (4.9)$$

with $\begin{cases} \varepsilon_R = \rho(\cos \theta_1 + \cos \theta_2) \\ \varepsilon_Z = \rho(\sin \theta_1 + \sin \theta_2) \end{cases}$

where θ_1 and θ_2 are respectively the angles for the rotation over the primary and secondary circles (cf. Fig. 3.4) and where we assumed that the Maxwellian F_M and, through it, the ion thermal velocity v_{th} , was spatially slowly varying and could be considered roughly constant on the scale of such trajectories.

The flux surface average of this expression is then given by:

$$\overline{\langle\langle g \rangle\rangle}(\alpha) = \left(\iint \frac{R}{R_0} \frac{dl}{d\theta} d\theta d\varphi \right)^{-1} \iint \frac{R}{R_0} \langle\langle g \rangle\rangle \frac{dl}{d\theta} d\theta d\varphi, \quad (4.10)$$

where $dl/d\theta = \sqrt{(dR/d\theta)^2 + (dZ/d\theta)^2}$. Using a third order expansion in ρ/a_0 , this massive integral simplifies into

$$\overline{\langle\langle g \rangle\rangle}(\alpha) = \frac{\rho_i}{2} \left(G_1 g'' + \frac{G_2}{\alpha} g' \right) + g + O((\rho/a_0)^4) \quad (4.11)$$

where g' denotes $\frac{\partial g}{\partial \alpha}$ and

$$\begin{cases} G_1 = \frac{1}{S_\alpha} \int \left(\frac{R}{R_0} \right)^3 \left(1 - \left(\frac{d\psi/d\alpha}{R_0} \right)^2 \|\nabla \alpha\|^2 \right) \|\nabla \alpha\|^2 \frac{dl}{d\theta} d\theta \\ G_2 = \frac{\alpha}{S_\alpha} \int \left(\frac{R}{R_0} \right)^3 \left(1 - \left(\frac{d\psi/d\alpha}{R_0} \right)^2 \|\nabla \alpha\|^2 \right) [\partial_{RR}^2 \alpha + \partial_{ZZ}^2 \alpha] \frac{dl}{d\theta} d\theta \\ S_\alpha = \int \frac{R}{R_0} \frac{dl}{d\theta} d\theta \end{cases} \quad (4.12)$$

So far, the parametrisation of the flux surfaces derived in sec. 4.1.2 has not been taken into account and the expression of the two geometric coefficients G_1 and G_2 is still independent of it. Their expression in the magnetic configuration considered here will be given later, after the equation defining the eigenfunction has been obtained. The remaining higher order, contained in $O((\rho/a_0)^4)$, also contain higher order derivatives of g .

In order to diagonalise the ZF equation and the operator $\overline{\langle\langle \cdot \rangle\rangle}$, we must now look for couples (λ, g_λ) where λ is the eigenvalue and g_λ the eigenfunction (which is a function of α only) that verifies

$$\lambda g_\lambda(\alpha) = \overline{\langle\langle g_\lambda \rangle\rangle}(\alpha), \quad (4.13)$$

giving the equation that defines the eigenfunctions:

$$G_1 g''_{\lambda} + \frac{G_2}{\alpha} g'_{\lambda} + \omega^2 g_{\lambda} = 0 \quad (4.14)$$

where $\omega^2 = \frac{2}{\rho_i^2}(1 - \lambda)$. The assumption that $\omega^2 \geq 0$ *i.e.* that $\lambda \leq 1$, although not formally demonstrated here, comes from the fact that $\|\langle\langle g_{\lambda} \rangle\rangle\|_2 \leq \|g_{\lambda}\|_2$. Although ω depends on the eigenvalue λ , the dependency is not explicitly denoted in what follows.

Once the exact configuration specified and this eigenfunction equation solved, only a discrete set of values of ω^2 will be compatible with the boundary conditions, allowing us to deduce the eigenvalues λ .

4.2.2 Eigenfunctions and eigenvalues

In order to solve the eigenfunction equation eq. (4.14), a more explicit expression of the two geometric terms G_1 and G_2 is required. Using the parametrisation of the magnetic flux surfaces derived in sec. 4.1.2, the various derivatives of α can be calculated, leading to:

$$\begin{cases} G_1 = G_{1,2} + G_{1,2} \cdot \left(\frac{\alpha}{a_0}\right)^2 + O(\varepsilon^4, \delta^4, \Delta^4) \\ G_2 = G_{2,2} + G_{2,2} \cdot \left(\frac{\alpha}{a_0}\right)^2 + O(\varepsilon^4, \delta^4, \Delta^4) \end{cases} \quad (4.15)$$

where the leading terms $G_{1,0}$ and $G_{2,0}$ only depend on the elongation κ and are of the order of 1 and the two following coefficients $G_{1,2}$ and $G_{2,2}$ of the second order in ε , δ and Δ . The exact expression of these 4 coefficients is a bit complicated, containing several elliptic integrals.

The resolution of Eq. (4.14) relies on the fact that the coefficients $G_{1,2}$ and $G_{2,2}$ are small (being of the second order in ε , δ and Δ).

To give a general idea of some properties of the eigenmodes and eigenvalues, it is worth mentioning that if we only consider the lowest order term *i.e.* for a quasi-circular configuration with a large aspect ratio, we can assume both $G_{1,0} \simeq G_{2,0} \simeq G_0 \simeq \frac{1+\kappa^2}{2\kappa^2}$ (κ close to 1) and $G_{1,2} = G_{2,2} = 0$, the eigenfunction equation for this simple case eigenpairs $(\lambda^0, g_{\lambda^0})$ reads:

$$g''_{\lambda^0} + \frac{1}{\alpha} g'_{\lambda^0} + \omega_0^2 g_{\lambda^0} = 0, \quad \omega_0^2 = \frac{\omega^2}{G_0}. \quad (4.16)$$

This equation happens to be, up to a rescaling, one of the definitions of the Bessel function J_0 and thus $g_{\lambda^0}(\alpha) \propto J_0(\omega_0 \alpha)$. Using the boundary condition $g_{\lambda^0}(a_0) = 0$,

we can finalise the diagonalisation:

$$\begin{cases} g_{\lambda^0}(\alpha) = g_{\lambda^0}(0) J_0(\omega_0 \alpha) = J_0\left(j_{0,k} \frac{\alpha}{a_0}\right) \\ \lambda_k^0 = 1 - \left(\frac{v_{th}}{\Omega_0}\right)^2 \omega_0^2 = 1 - \frac{1}{G_0} \left(\frac{v_{th}}{\Omega_0}\right)^2 \left(\frac{j_{0,k}}{a_0}\right)^2 \end{cases} \quad (4.17)$$

where $j_{0,k}$ is the k -th zero of J_0 .

In the case of a proper resolution, where $G_{1,2}$ and $G_{2,2}$ are taken into account, the eigenfunctions take a slightly more complicated form:

$$g_\lambda(\alpha) = g_\lambda(0) \left(\frac{\omega\alpha}{2}\right)^{-\beta} \left[J_\beta(\omega\alpha) + \left(\frac{G_{12}}{\omega^2}\right) g_\lambda^1(\alpha) + \left(\frac{G_{12}}{\omega^2}\right)^2 g_\lambda^2(\alpha) \right] \quad (4.18)$$

where $\beta = (G_{2,0} - G_{1,0})/(2G_{1,0})$ and the two higher order functions are given by:

$$g_\lambda^1(\alpha) = C_{11} \left(\frac{\omega\alpha}{2}\right)^3 J_{\beta+1}(\omega\alpha) + C_{12} \left(\frac{\omega\alpha}{2}\right)^2 J_{\beta+2}(\omega\alpha) \quad (4.19)$$

and

$$\begin{aligned} g_\lambda^2(\alpha) = C_{20} \left(\frac{\omega\alpha}{2}\right)^6 J_\beta(\omega\alpha) + C_{21} \left(\frac{\omega\alpha}{2}\right)^5 J_{\beta+1}(\omega\alpha) \\ + C_{22} \left(\frac{\omega\alpha}{2}\right)^4 J_{\beta+2}(\omega\alpha) + C_{23} \left(\frac{\omega\alpha}{2}\right)^3 J_{\beta+3}(\omega\alpha). \end{aligned} \quad (4.20)$$

Once this general form of the eigenfunctions is known, only the discrete values of ω compatible with the boundary conditions remain to be determined using the boundary condition $g(a_0) = 0$ to obtain the eigenvalues. In this general case, there is no way to express the eigenvalues formally but as the higher order term are small, they are obtained numerically by starting from the simple case eigenvalues λ_k^0 as given in Eq. (4.17) and refining by one or two iterations of a Newton method.

4.2.3 Correspondence to the Fourier description

Since these eigenfunction are not usually used in the resolution of the ZF equation, one may wonder how they relate to the usual methods used in circular magnetic configurations.

As mentioned in sec. 2.4.2, it is very common to compute the double averaging in Fourier space, where it corresponds to the multiplication of each radial Fourier mode $\mathbf{k}$ of Φ by $\Gamma_0(k_\perp^2 \rho_i^2)$:

$$\mathcal{F}(\langle\langle\Phi\rangle\rangle)_{\mathbf{k}} = \Gamma_0(k_\perp^2 \rho_i^2) \mathcal{F}(\Phi)_{\mathbf{k}}. \quad (4.21)$$

This expression is then sometimes further simplified by applying the Padé approximation (Eq. (2.62)) to Γ_0 :

$$\Gamma_0(k_\perp^2 \rho_i^2) \simeq \frac{1}{1 + k_\perp^2 \rho_i^2} \simeq 1 - k_\perp^2 \rho_i^2. \quad (4.22)$$

The Fourier transform of Bessel functions is not easy to represent and the approximation $k_\perp = k_r$ is not good near the magnetic centre. However, after a few zeros, the function J_0 becomes quite close to a cosine and the space between two consecutive zeros $j_{0,k}$ and $j_{0,k+1}$ converges to π . Considering an eigenfunction of the almost circular case $g_{\lambda^0}(\alpha) = J_0(j_{0,n}\alpha/a_0)$ with $k < n$ large enough to be far from the magnetic axis, for $j_{0,n}\alpha/a_0 \in [j_{0,k}, j_{0,k+1}]$:

$$J_0\left(j_{0,k} \frac{\alpha}{a_0}\right) \propto \sin(k_r \cdot (\alpha - a_0 j_{k,n}/j_{0,n})) \quad (4.23)$$

where

$$k_r = \frac{2\pi}{2a_0(j_{0,k+1} - j_{0,k})/j_{0,n}} \simeq \frac{j_{0,n}}{a_0}. \quad (4.24)$$

The double averaging of such g_{λ^0} near the magnetic centre using a Fourier representation would thus correspond to a multiplication by

$$\Gamma_0(k_r^2 \rho_i^2) \simeq 1 - \left(\frac{v_{th}}{\Omega_0}\right)^2 \omega_0^2. \quad (4.25)$$

which is indeed the eigenfunction we found in eq. (4.17). To further compare the 2, Fig. 4.3 gives a comparative view of the Fourier modes and the modes defined by our diagonalisation. Although both indeed look similar far away from the centre, the resemblance does not hold when the whole radial interval is considered as the very first peak of our eigenmodes is always larger and taller than the following ones. This difference comes from the fact that the Fourier approach is based on an approximation of local Cartesian coordinates, which does not apply well near the magnetic axis, where the curvature of the flux-surfaces becomes large.

4.3 Zonal Flow solver by diagonalisation

4.3.1 Diagonalisation based solver

The diagonalisation can be directly exploited to solve the ZF equation Eq. (2.65), having the same eigenfunctions as the flux averaged double averaging operator. Decomposing the right-hand side term on the eigenbasis:

$$\overline{\Phi} - \overline{\langle\langle\Phi\rangle\rangle} = RHS = \sum c_k g_{\lambda_k}, \quad (4.26)$$

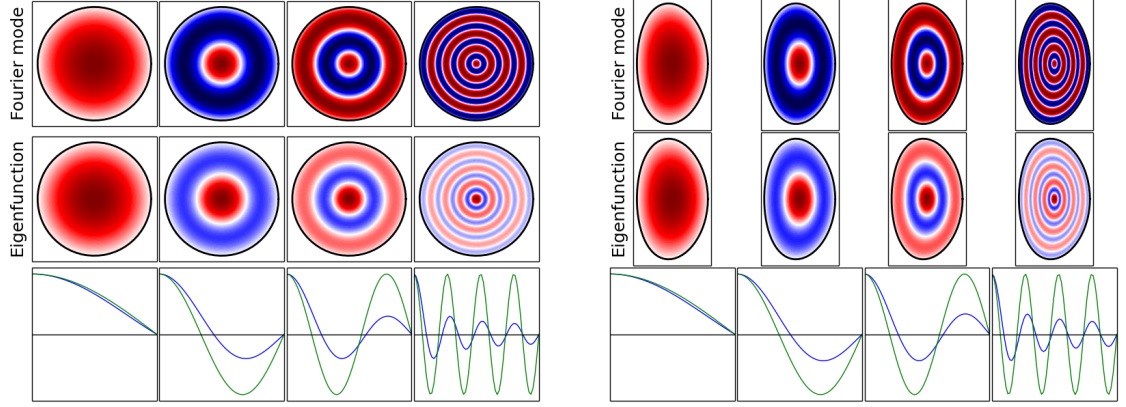


Figure 4.3: Surface plot comparing the first 3 modes as well as the 7th (starting from mode 0) of the Fourier base (top line) and our eigenbase (middle line) in the circular (left) and ITER-like ($\kappa = 1.7$ $\delta = 0.3$, right) cases. The bottom line gives a better view of the radial evolution of the 2 (green is Fourier, blue is our eigenmode).

the solution of the equation is simply equal to:

$$\bar{\Phi} = \sum c_k \lambda'_k g_{\lambda_k}, \quad \lambda'_k = 1/(1 - \lambda_k). \quad (4.27)$$

By extending the 1D eigenfunctions to 3D, Φ can be decomposed directly onto the eigenbasis by solving the linear system

$$\begin{cases} \sum c_k \langle g_{\lambda_k} | g_{\lambda_1} \rangle = \langle \Phi | g_{\lambda_1} \rangle \\ \sum c_k \langle g_{\lambda_k} | g_{\lambda_2} \rangle = \langle \Phi | g_{\lambda_2} \rangle \\ \vdots \end{cases} \quad (4.28)$$

where the unknown are the weights ($c_1, c_2, \dots$) and where $\langle \cdot | \cdot \rangle$ is the dot product

$$\langle f | g \rangle = \iiint R f(R, Z, \phi) g(R, Z, \phi) dR dZ d\phi \quad (4.29)$$

Although the eigenfunctions g_{λ_k} are not orthogonal to each other (*i.e.* the matrix of the linear system is not diagonal), the resolution is a fairly light task.

It is a well known result, in the local approximation, that the first eigenvalue of the ZF equation is very large and followed by a fast decrease (as $1/k_r^2$). This property can also be seen here in the low order case as the factor $1/j_{0,k}^2$ decreases roughly quadratically. Consequently, small eigenbases can be used safely without losing much numerical precision; our simulations usually only use the first 10-15 pairs of eigenfunctions and eigenvalues.

4.3.2 Comparative performances

Although the zeros of the eigenfunctions cannot be calculated theoretically, they are quite close from the previously mentioned quasi-circular case. Numerically, starting from those values, a few iterations of a Newton method allow to quickly compute the actual zeros, allowing to obtain the eigenvalues and evaluate the eigenfunctions at each point of the poloidal section. When using the solver by diagonalisation, those few quick steps are the only initialisation needed before beginning a simulation. This initialisation process is much simpler and faster than the computation and inversion of the matrix for the least square used in the projection solver. Similarly, each time the ZF equation is solved, it is much faster to project the RHS term into the eigenbasis as explained above than to apply the least square method of the projection solver.

However, when it comes to the ZF solver, computation time is rarely an issue as it usually amounts for less than a percent of the total computation time. The number of cores used for a simulation is also often of the order of the number of grid points in a poloidal section, as simulation on 8192 cores of a circular 128×128 grid only has $16\,384 \cdot \pi^2/4 = 12\,800$ active mesh points, corresponding to a bare 1.6 mesh point per core.

Instead, the real strength of the diagonalisation solver lies in its accuracy and the lower number of grid points required to reach a converged result. Fig. 4.4 shows the same simulation performed with both solvers at different grid resolution, clearly showing that the diagonalisation solver can capture the GAM damping with a 64×64 poloidal resolution while the projection solver requires at least twice that (4 times as many points) before an acceptable result is obtained. This higher accuracy is a crucial advantage as the poloidal grid's resolution is usually the constraining factor in the CFL condition, meaning that a reduction from 128×128 to 64×64 is likely to yield an 8-fold computation time reduction.

4.4 Shaping effects

To evaluate the quality of our ZF solver, we show here a collisionless GAM damping test and further investigate the effects of non-circular section plasma on the ZF and GAM dynamics. In the following simulations [Rosenbluth and Hinton, 1998], we set as initial perturbation of the form $\delta f = 10^{-3} \sin\left(\frac{\pi}{2}(1 - r/a)\right)$ leading to the collisionless damping of (GAMs) in a homogeneous plasma with density and temperature profiles flat and equal for both ions and electrons ($L_n = L_T = \infty$ and $T_i = T_e$). Simulations were performed using a minor radius $a_0 = 140\rho_{ti}$ (where ρ_{ti} is the ion Larmor radius evaluated with the thermal velocity) and $a_0/R_0 = \varepsilon = 0.1$. The q-profile was chosen to verify $q_s = q(a_0/2) = 2$ and the magnetic shear $s(a_0/2) = 0.6$.

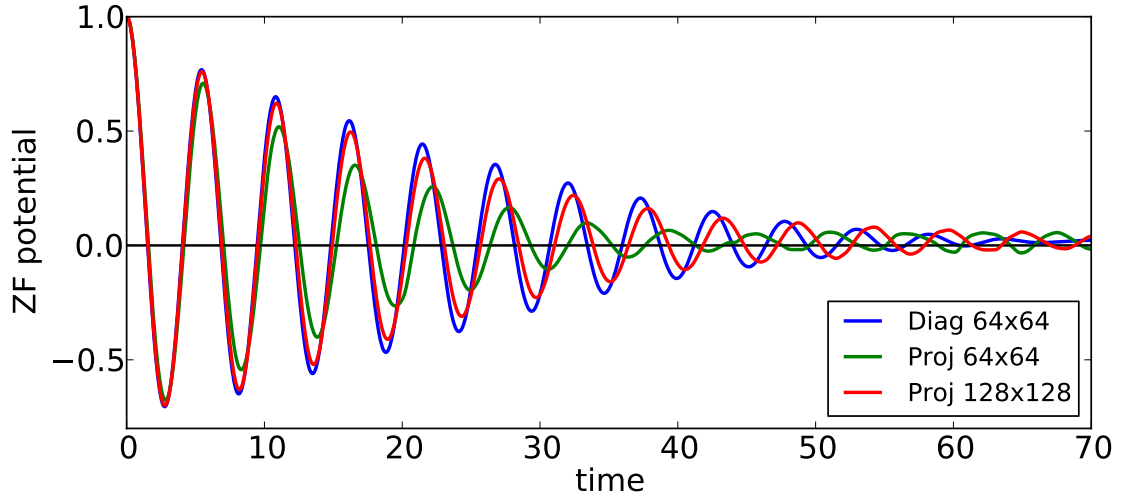


Figure 4.4: The diagonalisation solver is found to be converged on a 64×64 grid in the cross-section (blue) while the projection solver requires at least 128×128 (red, compared to the 64×64 case in green) to reach an equivalent accuracy.

These values were chosen for comparison with numerical results from the other articles [Angelino et al., 2008].

Most parameters of the mesh were fixed $(N_Z, N_\Phi, N_v, N_u) = (64, 1, 128, 8)$ while N_R decreased with the elongation such that $N_R = N_Z/\kappa$, maintaining square mesh cells in the poloidal plane, simply because a higher resolutions did not improve much the results. The ZF equation was solved using a projection on the first 15 eigenvalues. Simulation ran using the projection ZF solver for comparison used the same parameters except for $N_Z = 256$ and the corresponding $N_R = N_Z/\kappa$. As the new ZF solver naturally takes into account the geometric properties of the system, it can accurately capture the GAM damping on grids of much lower resolutions than the projection ZF solver.

The results are presented through the following 3 parameters: residual ZF level, the oscillation frequency ω and the damping rate γ .

4.4.1 Influence of elongation

The phenomenon of GAM damping has been extensively studied in the local approximation of δf codes in the circular case [Rosenbluth and Hinton, 1998, Sugama and Watanabe, 2006] and is commonly used as a benchmark test for gyrokinetics simulation codes [Obrejan et al., 2015]. For non circular cases, on the other hand, the information available is more sparse. Our numerical results for the

elliptic case with $\kappa \in [1, 2]$ are shown on Fig. 4.5, along with some theoretical or numerical results from other sources [Angelino et al., 2008].

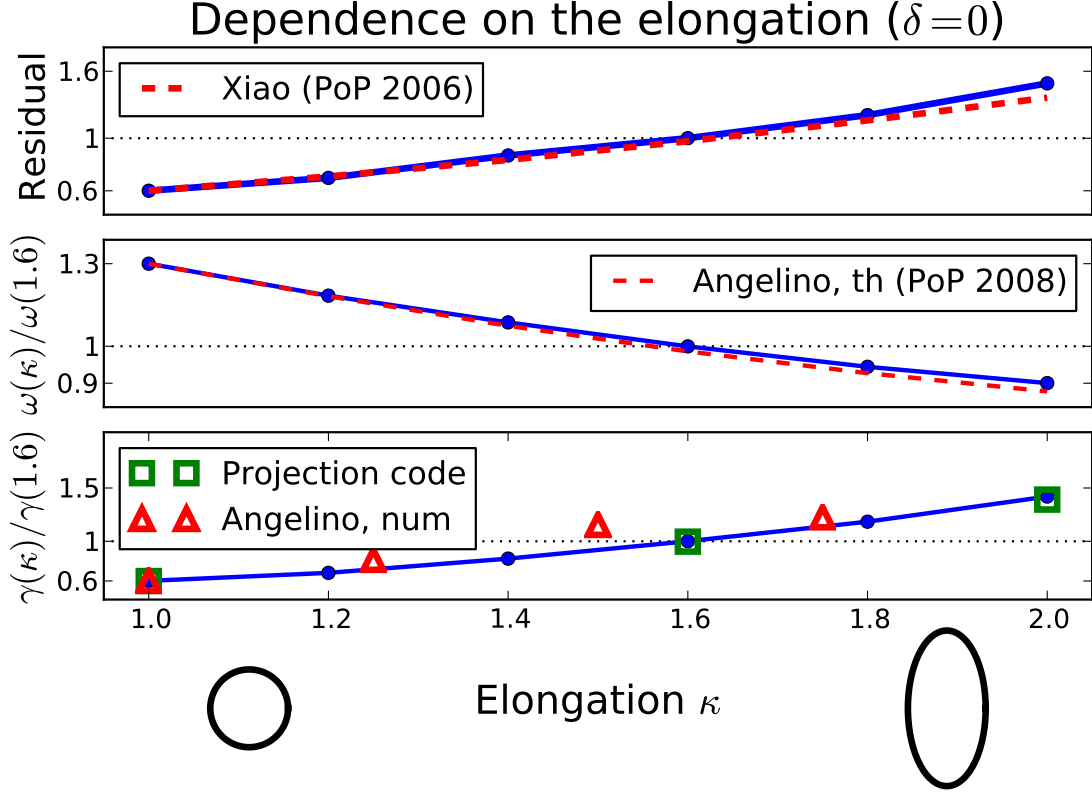


Figure 4.5: Influence of the elongation κ on the residual ZF, oscillation period ω and damping rate γ . The two former are plotted alongside theoretical results from other sources [Xiao and Catto, 2006, Angelino et al., 2008], the latter includes numerical results from another source [Angelino et al., 2008] (red triangles) and a few points from the projection code (green squares). The results on the 3 plots are normalised with respect to the case $\kappa = 1.6$ to simplify comparison with the results on D-shapes presented later.

The residual ZF level is found to increase rapidly as the elongation increases from 1 to 2, nearly tripling, while the oscillation frequency ω decreases by about 30% (see the top two plots in Fig. 4.5). The evolution of these two parameters is in good agreement with the theoretical formulas found respectively in [Xiao and Catto, 2006] and [Angelino et al., 2008].

The damping rate γ also increases rapidly with the elongation and becomes about 2.5 times larger for $\kappa = 2$ compared to the circular case (see bottom plot in Fig. 4.5). As no formula derived from theory is available to evaluate the quality of

our numerical results, our data is this time compared to the numerical results shown in [Angelino et al., 2008] and the ones obtained with our own projection ZF solver. The three sets of numerical results are in good agreement.

4.4.2 Influence of triangularity

The effect of triangularity on GAM damping has been studied even less than the one of elongation and in the references mentioned above, where only the theoretical formula for the ZF residual level still applies. The reason for this scarcity is that the triangularity is often considered to be of very small influence on the dynamics of collisionless GAM damping, compared to elongation, while also making theoretical calculations much more complicated.

We first performed numerical simulations for triangularities δ ranging from -0.3 to $+0.3$ with an elongation of $\kappa = 1.6$ as base case (close to the ITER plasma shape). The results are presented on Fig. 4.6, using the same scaling and normalisation as for elongation in Fig. 4.5 to facilitate the comparison. In the case of the damping rate, the numerical results from our new ZF solver were once again compared with numerical results from our older ZF solver.

While our results do confirm that triangularity is of almost no influence on either the residual ZF level (in good agreement with theory [Xiao and Catto, 2006]) and the oscillation frequency ω (see the top two plots on Fig. 4.6). However the damping rate is found to depend quadratically on the triangularity with an amplitude similar to what was observed for the dependence on the elongation.

This simulation was also performed for various central safety factor q_s , elongation κ and inverse aspect ratio ε to evaluate how the influence of δ on the damping rate changes. Figure 4.7 compares our reference configuration $(\varepsilon, q_s, \kappa) = (0.1, 2, 1.6)$ to 3 alternative configuration in which only one of the 3 parameters has been changed; for each case, the same data is shown twice, either all normalised by the damping rate $\gamma_{ref}(\delta = 0)$ of the reference configuration with zero triangularity (top 3 plots) or each by their own damping rate $\gamma(\delta = 0)$ with zero triangularity (bottom 3 plots). While the former shows that the quadratic profile of $\gamma(\delta)$ remains roughly the same in all cases, the latter highlights small differences in the amplitude of the parabolas. More precisely, we can see that while a lower q_s , higher κ or ε increase the central value $\gamma(\delta = 0)$, within the window of parameters chosen, they also weaken the influence of δ on the damping rate: cases with a high damping rates in the elliptic case gain less from the addition of a triangularity.

In order to investigate further about this tendency, a total of 52 simulations were performed with the following set of initial parameters: $\varepsilon = 0.1, 0.23, 0.36$, $q_s = 1.4, 1.7, 2$, $\kappa = 1.3, 1.6, 1.9$ and $\delta = 0, 0.4$. Fig. 4.8 shows the damping rate for $\delta = 0.4$ as a function of the one for $\delta = 0$, as points which were coloured according to the elongation κ to highlight a clear linear dependence. Based on these

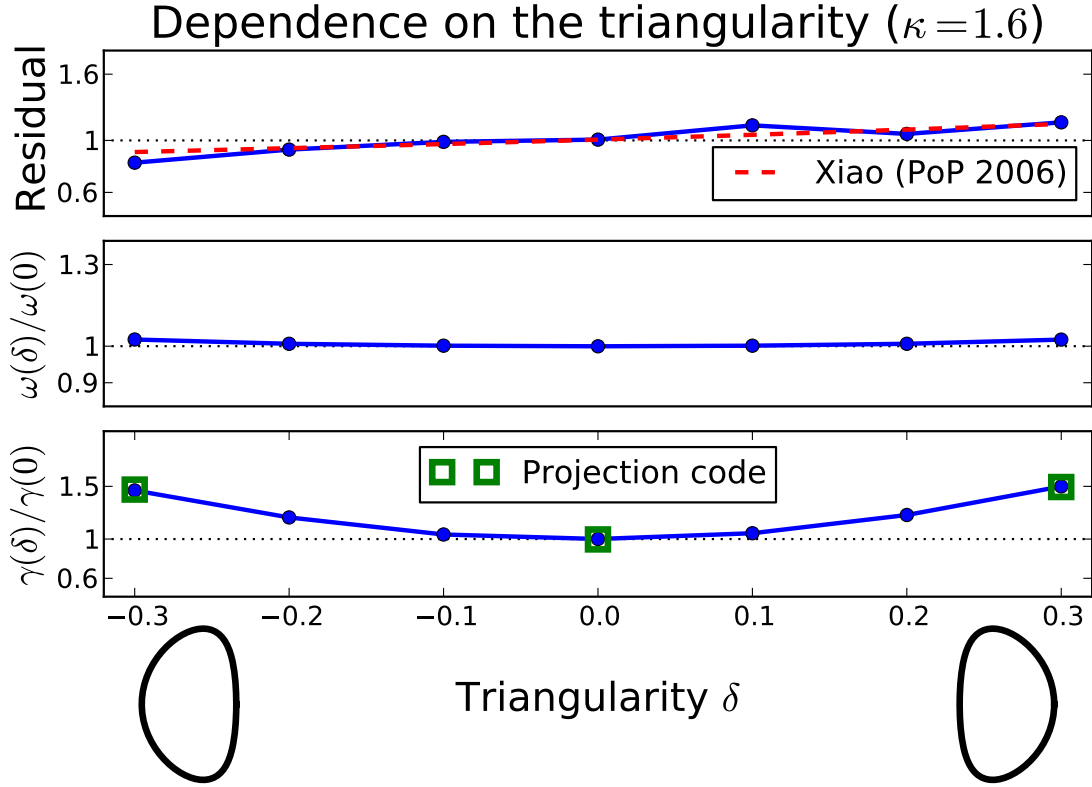


Figure 4.6: Influence of the triangularity δ on the residual ZF, oscillation period ω and damping rate γ . Once again the results on the 3 plots are normalised with respect to the case $\kappa = 1.6$, $\delta = 0$ to simplify comparison with the results of the previous figure, showing the influence of elongation. The results for the damping rate are compared with a few points from the projection code (green squares)

results and continuing to assume a quadratic profile for $\gamma(\delta)$, an observational law can be inferred:

$$\frac{R_0}{v_{th}}\gamma(\delta) = \frac{R_0}{v_{th}}\gamma(\delta = 0) + \left[\frac{1}{3} + \frac{R_0}{v_{th}}\gamma(\delta = 0) \left(-\frac{1}{2} + 4(\kappa - 2)^2 \right) \right] \cdot \delta^2. \quad (4.30)$$

The solid lines of corresponding colours in figure 4.8 correspond to the estimation provided by this formula.

As $R_0\gamma(\delta = 0)/v_{th}$ is usually small, the coefficient in front of δ^2 does not vary much from $1/3$, which explains why the curves on the top 3 plots of Fig. 4.7 were so similar. This numerical law also confirms that for the range of elongations considered so far, an elliptic configuration with a high damping rate $\gamma(\delta = 0)$ will see its damping rate increase less if a triangularity is added (as $-1/2 + 4(\kappa - 2)^2$ is

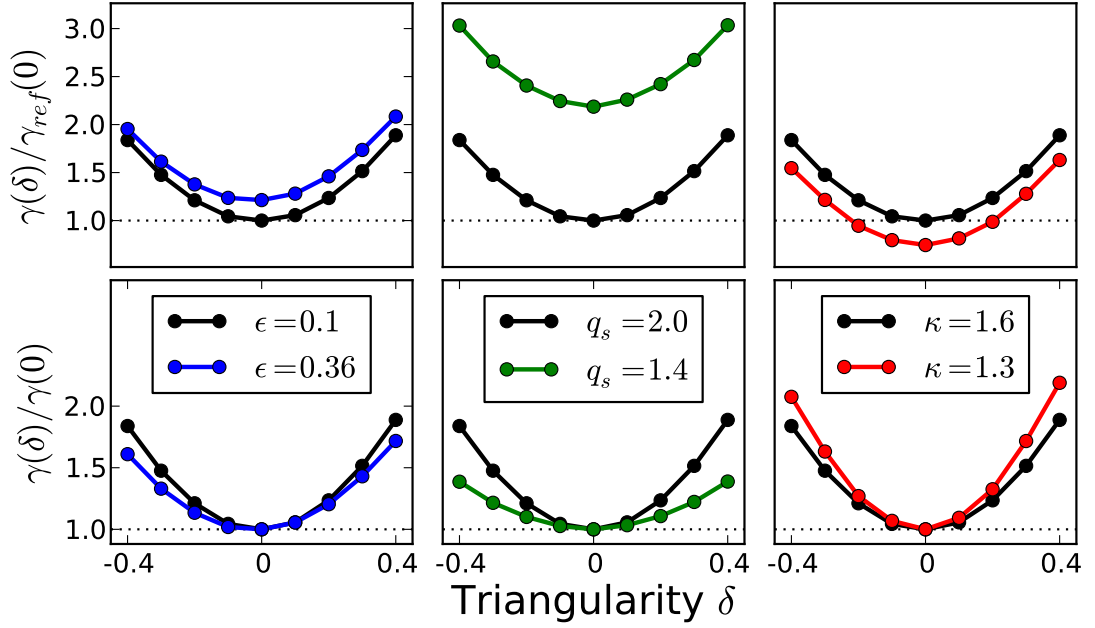


Figure 4.7: Mutual influence on the damping rate γ of the triangularity δ with the safety factor at half radius q_s , the elongation κ and the inverse aspect ratio ϵ on the damping rate. The reference configuration $(\epsilon, q_s, \kappa) = (0.1, 2, 1.6)$ is represented in black on all plots and compared to 3 other configurations where only one parameter is changed. The two plots on each column show the same data, only the normalisation changes: the top 3 plots are normalised by a common $\gamma_{ref}(\delta = 0)$ while the bottom 3 plots are normalised by their own $\gamma(\delta = 0)$ to better show the difference in amplitude.

negative). This tendency is expected to be reversed for $\kappa > 2$. However, for such high elongations, the damping is so fast that no proper oscillations can be observed in the time evolution of the electric potential and the interpolation of a damping rate is impossible.

4.5 Summary

This chapter aimed at understanding better the geometry of the flux surfaces and their inclusion in a global code. After introducing how GKNET represents the flux function and parametrising its iso-surfaces, we exploited this formal framework to study the mathematical properties of the ZF equation itself. Its diagonalisation shed a new light on its intrinsic geometric properties and allowed to design a new

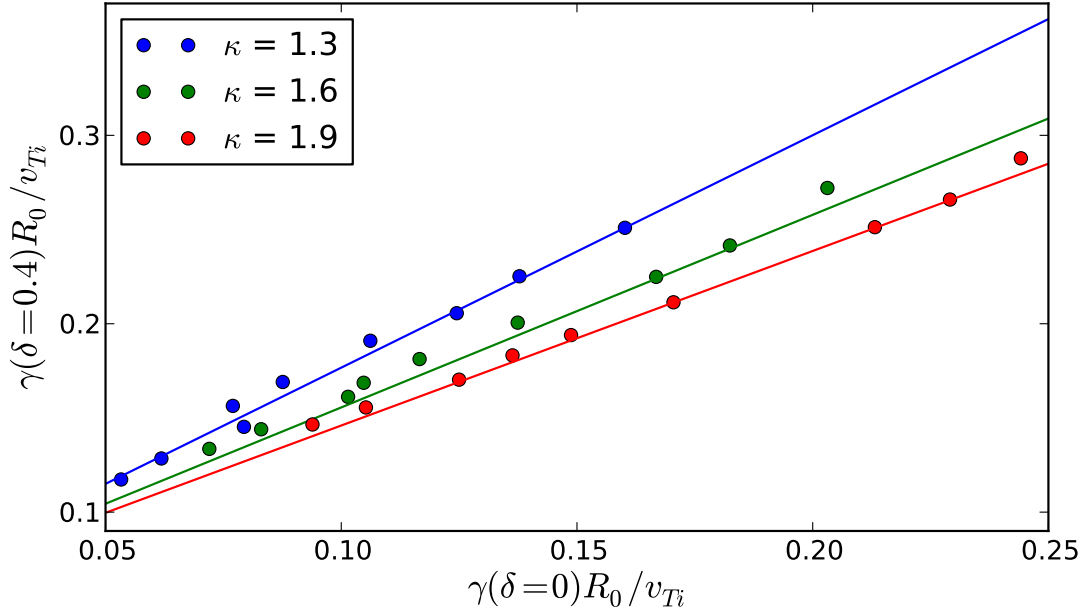


Figure 4.8: Damping rate for a triangularity of $\delta = 0.4$ as a function of the damping rate for its elliptic counter part. The points correspond to numerical results for various initial conditions and have been coloured according to their elongation κ to highlight linear dependencies. The solid lines were obtained with the formula Eq. (4.30).

method to solve it efficiently, even on numerical grids whose shape does not allow an accurate representation of the flux surfaces.

After recovering the already existing results, stating that higher elongations cause a faster damping of ZF towards a higher residual, additional new results were introduced about the influence of triangularity. We thus showed that while its effects are small on both the residual ZF and oscillation frequency, as is often assumed, triangularity has a strong influence on the damping rate. Indeed, both higher elongations and higher triangularities in absolute value increase the damping rate in comparable orders of magnitude. This influence was furthermore quantified with the introduction a new empirical law.

However, ZF benchmark tests, such as the one presented in this study focuses on a local property, as it is what most codes are limited to. To go further and exploit better the global full-f capacities of GKNET, a study of the ITG instability is presented in the next chapter.

Chapter 5

Full-f study of magnetic shaping effects on the Ion Temperature Gradient instability

Contents

5.1	The Ion Temperature Gradient Instability	90
5.1.1	Creation of ballooning structures	90
5.1.2	From linear ITG modes growth to the saturated phase	91
5.2	Difference between δf and full-f in circular configuration	92
5.3	Influence of elongation	93
5.3.1	Linear ITG growth rates	93
5.3.2	Ballooning structures	95
5.3.3	Combining results	95
5.4	Influence of triangularity	97
5.5	Summary	98

5.1 The Ion Temperature Gradient Instability

5.1.1 Creation of ballooning structures

The ITG instability arises when 2 neighbouring regions of the plasma are at different temperatures and are additionally traversed by magnetic fields of different strength. As higher temperatures means more kinetic energy, hotter particles move further under ∇B -drift. This difference in drift velocity creates a charge gradient which in turns creates an electric field $\mathbf{E}$. In areas where ∇B is aligned with ∇T , the subsequent $\mathbf{E} \times \mathbf{B}$ -drift reinforces the ∇B -drift in an unstable positive feedback loop.

A Fourier transform of the electrostatic potential Φ in toroidal direction, allows observing these areas where the charge difference has creates an electric field. Figure 5.1 shows the poloidal cross-section of a few toroidal modes when an ITG instability is present, as obtained with a δf simulation code. While the temperature profiles are essentially radial, those so-called ballooning structures are not since ∇B varies roughly like $1/R$. On the outer side, ∇B and ∇T are aligned, creating the actual instability; on the inner side, the opposite causes the $\mathbf{E} \times \mathbf{B}$ -drift to suppress the perturbation.

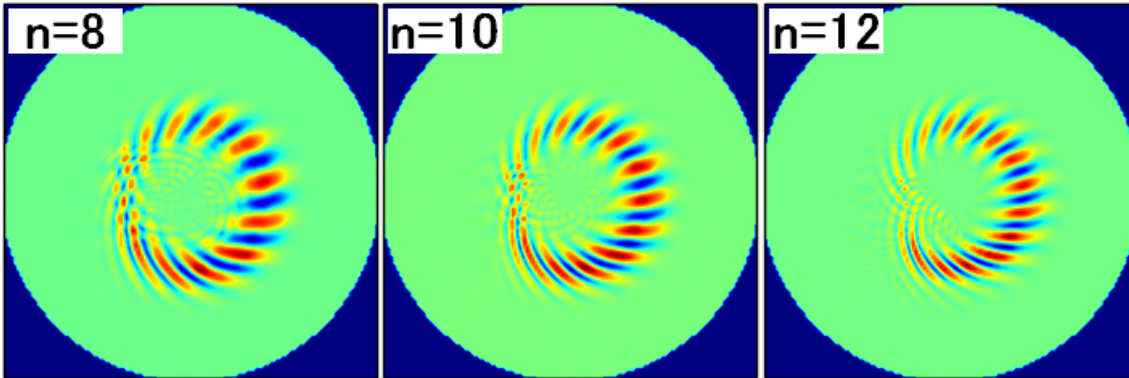


Figure 5.1: Example of typical ballooning structures in the circular case (using δf models).

A similar mechanism can occur for electrons, although their ∇B -drifts happen in opposite direction. This leads to the Electron Temperature Gradient (ETG) instability, which is beyond the scope of the present study and requires the inclusion of kinetic electrons in the simulation code.

5.1.2 From linear ITG modes growth to the saturated phase

In the linear theory, the growth of the ITG instability can be observed through the exponential growth of the toroidal modes of electrostatic potential Φ . Not all modes grow at the same speed and the growth rate of the dominant mode as well as its mode number are characteristic numbers for a given configuration. The 2 plots on the left of Fig. 5.2 show an example of such growth of the modes' energy and the growth rate profile.

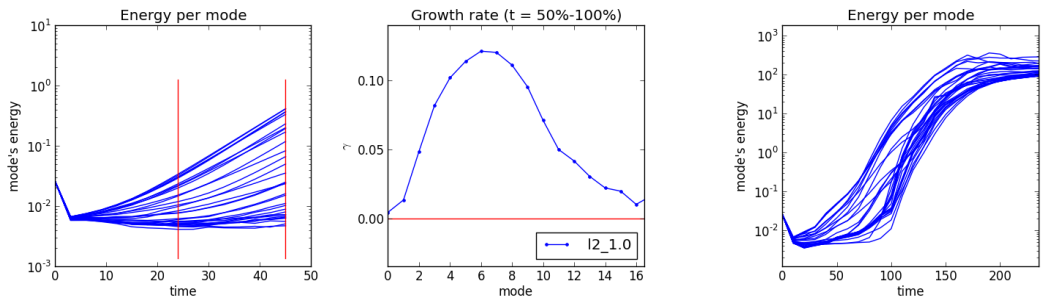


Figure 5.2: Example of growth of the electrostatic potential's toroidal modes. Left: exponential growth of the energy with time. Centre: profile of the growth rate as a function of their mode number. Right: transition from linear to saturated phase.

Of course, the modes cannot gain energy forever and after growing by several orders of magnitude, a saturation occurs. The exponential growth is the consequence of the domination of the linear terms over the nonlinear ones (quadratic) and it continues until the situation is reversed, which is known as linear saturation. This transition can be seen on the right plot of Fig. 5.2, after a growth for about 3 times longer than what was shown before. The initial phase is referred to as the linear phase, while the latter is either called the nonlinear or the saturated phase.

A characteristic object of study in the saturated phase is the ion heat diffusivity χ_i which characterises neoclassical transport and hopefully anomalous transport. While the ballooning structures play a major role in the linear phase, when the interaction between the modes can be neglected, they are destroyed in the saturated phase. However, the mixing length theory predicts that the ion heat diffusivity χ_i is roughly proportional to γ/k_r^2 where γ is the growth rate and k_r^2 the radial mode's frequency, which is inversely proportional to the ballooning structure's width.

5.2 Difference between δf and full-f in circular configuration

The simulations presented in this study rely on 2 operating modes of GKNET: δf and full-f. The magnetic flux surface average term $\bar{\Phi}$ in the field equation eq. (2.64), which is commonly omitted for linear benchmark, is found to lead to significant differences, even early within the so-called linear phase.

To highlight those differences, some simulation results will be first given in a circular configuration similar to the CBC presented above, except with $q(r_s) = 2$ and $\hat{s}(r_s) = 1$ as this alternative case appears in some of the references cited later on. Figure 5.3 shows the linear ITG growth rates and ballooning mode structures for the dominant modes in the circular case for several variations of the model. The growth rates are shown for a δf scheme (A), a full-f scheme where $\bar{\Phi}$ was omitted (B) (denoted as "mixed") and a proper full-f scheme (C), with its $\bar{\Phi}$, while the mode structures are only given for the δf and proper full-f cases. Indeed, (A) and (B) exhibit similar growth rate profiles, their modes structures are nearly identical, justifying for instance the omission of $\bar{\Phi}$ in the linear benchmark presented in [Grandgirard et al., 2006]. However model (C) leads to lower growth rates (despite quite similar overall shape of the profile) and more strongly tilted V-shaped ballooning modes. It is to be noted however that the mode growth in the full-f case are not perfect exponential growth and can be harder to measure accurately.

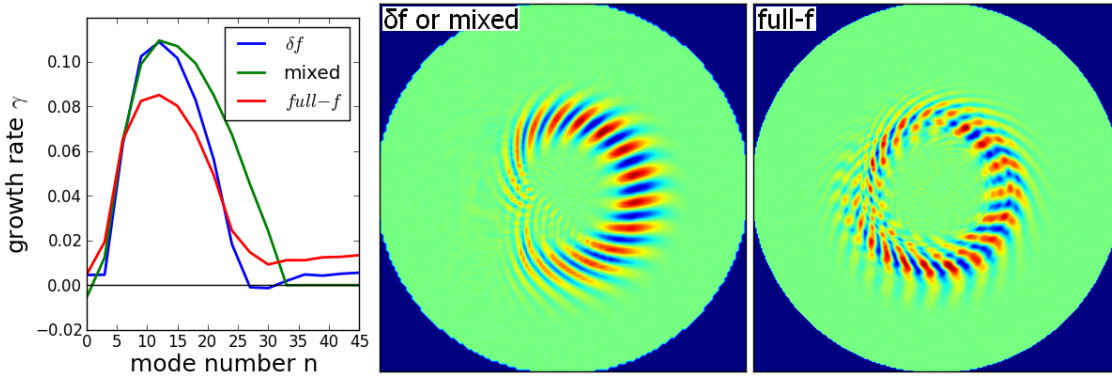


Figure 5.3: Left: growth rate profiles as obtained with the schemes (A) (blue), (B) (green) and (C) (red). Center: ballooning mode structures for the δf scheme. Right: same for the full-f scheme.

The specificity of the results obtained with scheme (C) can be checked further on figure 5.4 which shows the structures for several modes using δf or full-f models that include or not $\bar{\Phi}$. As can be expected, the presence or not of $\bar{\Phi}$ in the field equation of a δf scheme does not have any influence on the results, as the mode are not

expected to interact within a linear phase. The only line that really differs from the others is the last one, corresponding to scheme (C). The close similarity of scheme (B) with the linear cases has been for instance exploited in [Grandgirard et al., 2016] to match linear benchmark results with a full-f code.

5.3 Influence of elongation

While plasma shaping and its influence on drift wave linear stability and MHD stability have been studied in details, such studies are less common in gyrokinetics, and in particular none has been performed using a full-f simulation code. This section explores the influence of elongation κ on ballooning structures, after presenting some simpler results on linear ITG growth rates, which are somehow often overlooked.

5.3.1 Linear ITG growth rates

We first explore the effects of magnetic shaping on the linear gyrokinetic stability of ITG modes. While some other sources have presented a study at either a fixed value $k_\theta \rho_i$ [Kinsey et al., 2007] or considering only the dominant mode [Belli et al., 2008], we find it interesting to instead look at the complete growth rate profile. Figure 5.5 shows the growth rate profiles of the toroidal linear ITG modes for 5 different elongations with both the linear and full-f schemes. In both cases, the general evolution of the profile is found to be richer than a simple change in amplitude as they also make the dominant mode shifts towards higher mode numbers. After a stagnation or slight increase of the amplitude growth rate profile, both schemes see the growth rates decrease roughly linearly.

The shift towards higher $k_\theta \rho_i$ is mentioned in some sources based on simple models [Anderson et al., 2000] but does not seem to have been explored numerically. It implies in particular that numerical codes of low order in terms of $k_\theta \rho_i$, might miss some of the dynamics when varying the shape of their simulation domain. In particular, since the width of our simulation domain is kept constant and its height increased with the elongation κ , the domain and the associated equilibrium configuration f_0 as well as temperature profiles *etc* are being stretched vertically, while no distance has been compressed.

Paradoxically, such a shift can be prevented if the domain is deformed differently as the elongation increases. We mentioned above the alternative approach in which the surface area of the cross section is kept constant, decreasing the domain's width to compensate for the additional height. In doing so, if the temperature profile is defined not in term of a/L_T as we did, but of R_0/L_T , effectively staying constant as the elongation is varied, we find that the $k_\theta \rho_i$ shift does not occur. However, despite

5. FULL-F STUDY OF MAGNETIC SHAPING EFFECTS ON THE ION TEMPERATURE GRADIENT INSTABILITY

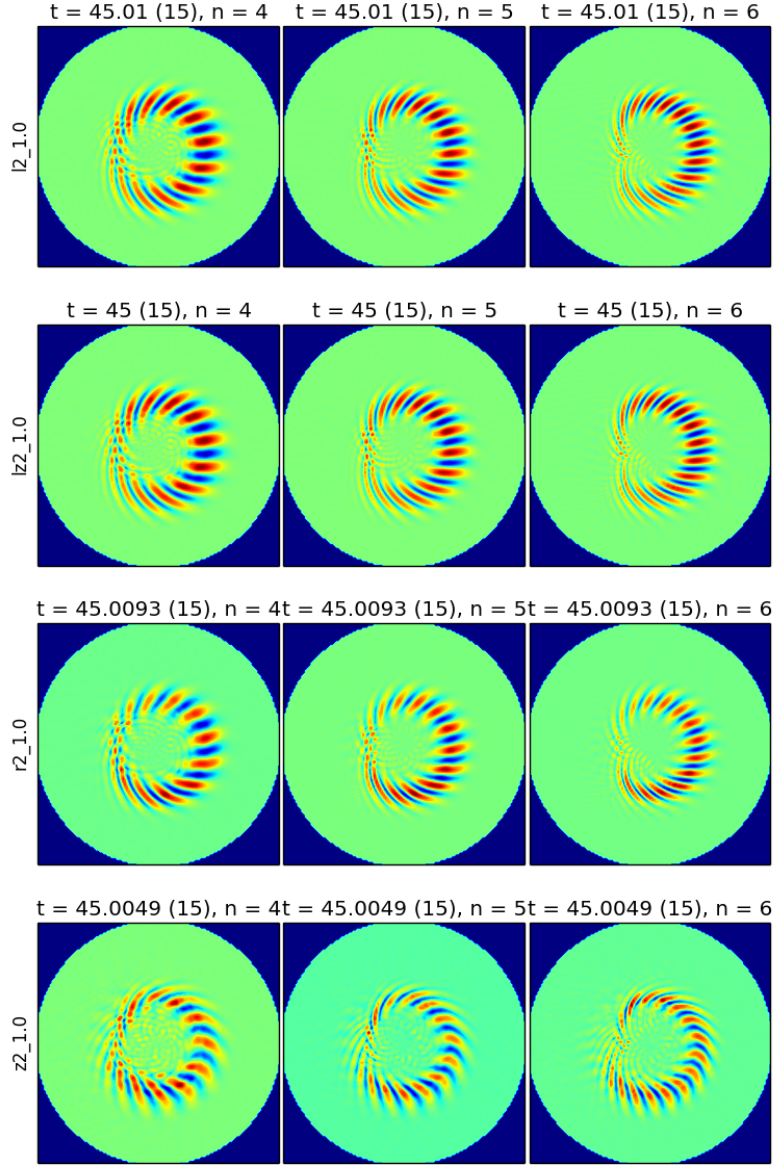


Figure 5.4: Structures of modes $n=8$ (left column), $n=10$ (central one) and $n=12$ (right column) for the δf scheme without mean flow term (first line), with mean flow term (second line), full-f scheme without mean flow term (third line) and with (bottom line).

this interesting property and the simplifications it allows in the study the influence of κ on growth rates, the rest of this study does not utilise such an approach.

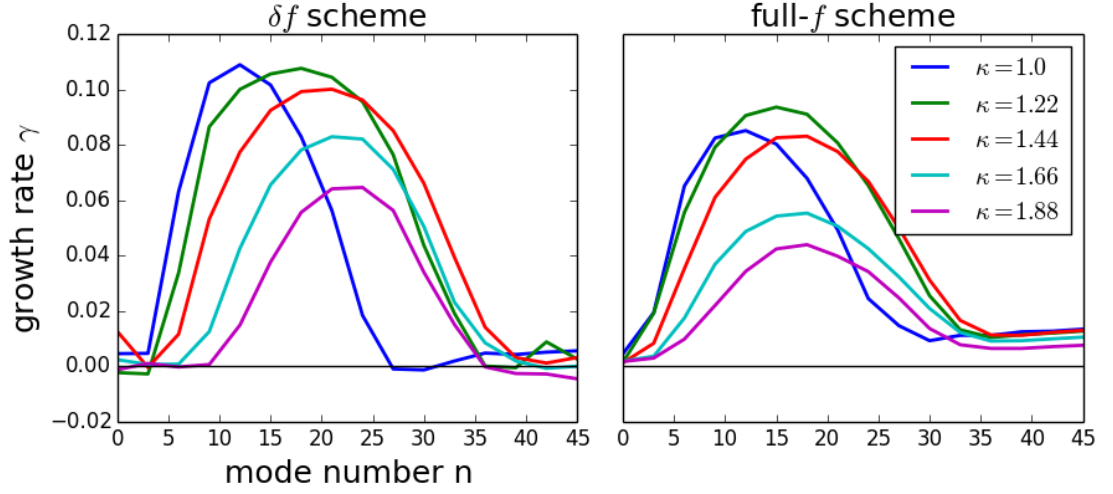


Figure 5.5: Growth rate profiles for various elongations with both the δf scheme (left) and the full-f one (right).

5.3.2 Ballooning structures

The influence of elongation on mode structures can be observed through their width as well as overall shape. Figure 5.6 shows the dominant mode's structure for the elongated domains $\kappa = 1, 1.22, 1.44, 1.66$ cases with the full-f scheme. A common property with the δf case (not shown here) is that the radial width of the modes decreases quickly with elongation. However, the full-f scheme also has the interesting property that the V-shaped mode structures straighten as their tilting angle changes of sign. This tendency remains when looking at more modes and more elongations, although only a few are presented here by lack of space.

To evaluate the influence of elongation more precisely, the width of the associated ballooning structures is estimated numerically by projecting a portion of the weak field half of the cross-section onto the radial line $R > R_0, Z = 0$. After searching for a maximum of $|\Phi|^2$, ballooning mode's width is taken to be the length of the interval where $|\Phi|^2$ is at least half that maximum. Figure 5.7 shows the width of several modes, drawing a line between the dominant ones for each for each value of κ , in a roughly linear decrease.

5.3.3 Combining results

The results presented above can be combined using the mixing length formula to give an estimate of the dependence of χ_i on the elongation κ . The formula can be expressed as $\chi_i \sim \gamma w^2$, where w is the ballooning mode structure. Since there

5. FULL-F STUDY OF MAGNETIC SHAPING EFFECTS ON THE ION TEMPERATURE GRADIENT INSTABILITY

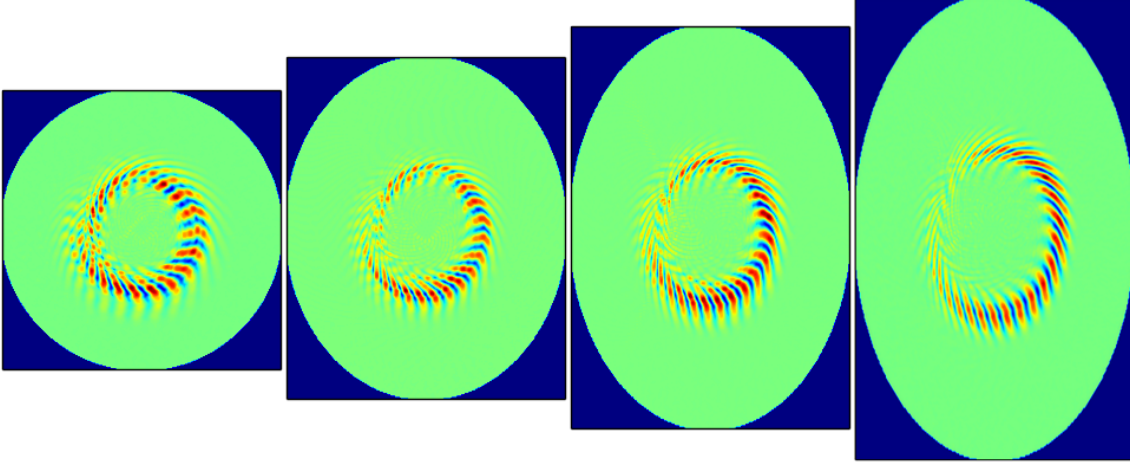


Figure 5.6: Structure of the dominant mode for elongations $\kappa = 1, 1.22, 1.44, 1.66$.

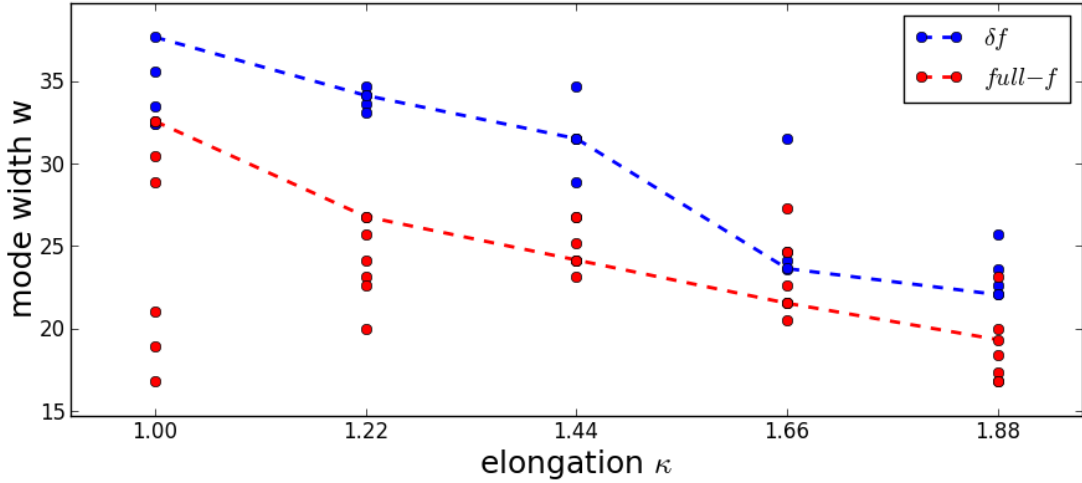


Figure 5.7: Radial width of various modes' structures with the δf and full-f schemes. The dashed line runs through the dominant mode for each value of κ .

is more than one mode and the formula only mentions one, we will here use the dominant mode for both the growth rate and mode width.

Given the particular behaviour of growth rate profiles between $\kappa = 1$ and 1.22, and since we only want tendency, we can only consider the growth rates profiles above $\kappa = 1.22$ where both the δf and full-f growth rates decrease in amplitude roughly linearly. The case of mode width has already been shown on Figure 5.7 and also shows a roughly linear decrease in both cases. Combining those 2 results, it can

be estimated that $\chi_i \sim \kappa^{-3}$, suggesting a strongly stabilising influence of elongation (as higher elongations reduce the heat losses).

This result can be compared qualitatively and quantitatively to the 2 sources: [Kinsey et al., 2007] and [Belli et al., 2008], which also give power laws. While our estimates do agree in terms of tendency with those 2, both find much weaker influences of respectively κ^{-1} and between $\kappa^{-1.5}$ and κ^{-2} . However it is worth noting that the latter, [Belli et al., 2008], mentions experimental observations which suggest much stronger dependencies, especially as the elongation increases.

5.4 Influence of triangularity

Figure 5.8 shows the changes in the growth rate profile for elongations $\kappa = 1.22$ and $\kappa = 1.88$ as the triangularity is varied from -0.3 to $+0.3$. Unlike elongation before, the tendencies seemed to remain the same in both δf and full-f, however this time it is elongation that leads to a disjunction of cases. For lowly elongation domains, triangularity is to shift the overall profile towards higher values of $k_\theta \rho_i$ while significantly increasing the amplitude of the growth rate profile. However, this influence diminishes with the elongation and the case $\kappa = 1.88$ on the figure does not show any significant difference between the various triangularities.

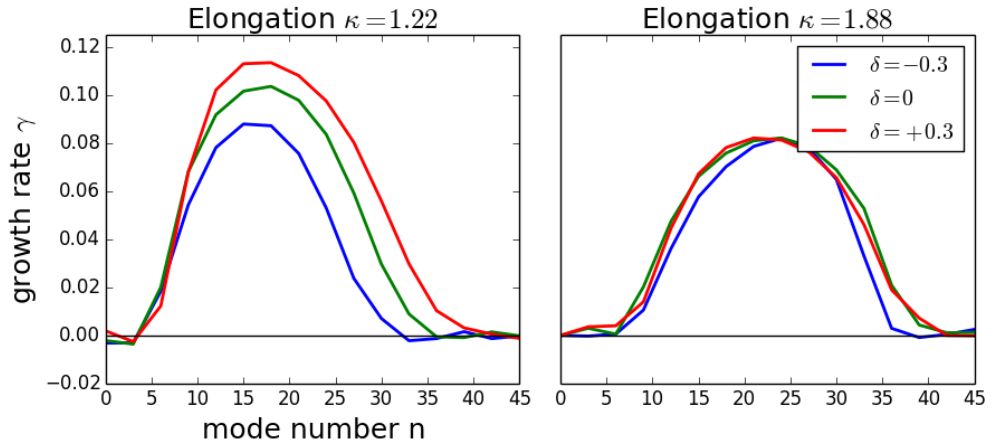


Figure 5.8: Linear growth rate profiles for elongations $\kappa = 1.22$ and $\kappa = 1.88$ with triangularities running from -0.3 to $+0.3$.

Continuing as was done with the elongation, the influence of triangularity on mode structures can be investigated. Figure 5.9 shows the mode structures for the dominant modes at elongations of $\kappa = 1.22$ and $\kappa = 1.88$ and triangularities of -0.3 and $+0.3$. Some small variations of mode width can be measured but they are too

5. FULL-F STUDY OF MAGNETIC SHAPING EFFECTS ON THE ION TEMPERATURE GRADIENT INSTABILITY

small to be relevant. This lack of influence of triangularity on mode with combined with vanishing effect on growth rates suggests a slightly destabilising influence of triangularity, which disappears for higher elongations.

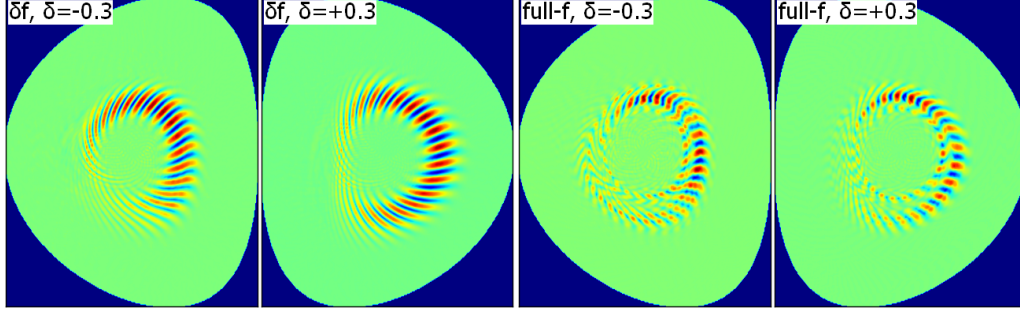


Figure 5.9: Mode structures for $\kappa = 1.22$ and triangularities of ± 0.3 with both schemes.

5.5 Summary

This chapter revolved around linear simulations of ITG instability, combining a study of the differences between δf or full-f models as well as magnetic shaping effects, in continuation of the previous chapter. Initial tests in the circular case showed that full-f simulations yield much smaller growth rates than in the δf case, and mode structures which are V-shaped instead of straight. This difference suggests that the so-called "linear" phase as it is usually understood does not exist: the modes do interact non-linearly with each other.

In elongated domains, both growth rates and mode width are found to decrease roughly linearly for both δf or full-f models. This tendency would correspond to a stabilising influence of elongation as it would reduce heat diffusivity with κ^{-3} . Higher elongation also tends to flatten the V-shape of the mode structures while reversing the sign of the tilting angle. The influence of triangularity was found to be much weaker and transition from slightly destabilising at low elongations having no real impact for more elongated cases.

Chapter 6

Conclusion

Contents

6.1	Summary of the results	100
6.2	The meaning of this all	101
6.3	To go beyond	102

6.1 Summary of the results

This thesis started with a very general introduction to what nuclear fusion is, where it is found in nature, and how we could hope on day to harness it. Although various types of experimental nuclear fusion reactors already exist, none has yet reached the maturity to enable this large scale, abundant and environmental friendly source of energy at an affordable cost. The brief introduction to possible modelisations and approaches to simulate plasmas naturally connected to chapter 2, which presented a rigorous derivation of the fundamental equations of gyrokinetics (GK), the model used throughout this study. In this theory, a gyro-averaged Vlasov equation is coupled to the GK quasi-neutrality condition to simulate the time evolution of the ion distribution and the electrostatic field it generates. As gyrokinetic simulations can take many forms, several possible combinations were presented along with a non exhaustive list of some of the major codes.

Having given an overview of the possible approaches GK offers for the simulation of plasmas, chapter 3 presented the design and main properties of GKNET, simulation code that was developed over the years, as part of this PhD. GKNET was developed on the basis of a slab geometry code, GKMUST, which was greatly modified into the full-f code in general geometry it is now. One of its rare features is its computation of gyro-averaged terms in real space, allowing for an accurate treatment of the dynamics at the scale of the ion Larmor radius as well as the fixed-point method used to solve the field equation, based on those real space gyro-averagings. It is also parallelised along its 5 dimensions and relies on advanced techniques to optimise this parallelisation.

GKNET was written with in mind the study of magnetic shaping effects in D-shaped domains. This quest started with the study of collisionless Zonal Flow (ZF) damping, which is presented in chapter 4, after a proper definition of the geometry that would be used. In addition to the numerical study, a new theoretical approach to the problem was developed with the formal diagonalisation of the ZF equation, followed by the design and implementation of a ZF solver based on it. With this new ZF solver, the numerical study of shaping effects was carried it out, first using existing results on the influence of elongation as a benchmark: we could verify with good accuracy that a higher elongation led to a larger residual ZF level while the oscillation frequency decreased and the damping rate increased. Despite the usual assumption that triangularity was of weak or no effect, our study proved that higher triangularity in absolute value increases the damping rate in proportions quite similar to the one of elongation, an influence which was quantified with the introduction of an empirical law.

Finally, GKNET was applied to the more complex phenomenon known as the ITG instability. The study included not only the influence of magnetic shaping but also of the use of a proper full-f model versus the much more common δf ones. Our

simulations suggest that, despite its name, the "linear phase" is not accurately represented by a linearised δf scheme as the interaction of modes leads to smaller linear mode growth rates and V-shaped ballooning structures rather straight. In elongated domains, both δf and full-f simulations exhibited a decrease of mode structure width and a shift of the growth rate profiles towards larger toroidal mode numbers, accompanied with a drop of overall amplitude from $\kappa = 1.4$. The combination of such tendencies suggests a stabilising influence of elongation, in agreement with other sources based on direct measurements of heat diffusivities in local codes. A similar method found a slight destabilising influence of triangularity which vanished as elongation increased.

6.2 The meaning of this all

This thesis revolved around the study of shaping effect, and their potential connection to anomalous transport. Many steps were necessary before on the way, with the development of a whole simulation code for this purpose. Our simulation code, GKNET, is not the first gyrokinetic simulation code, and neither can it be said to be better than all the others. However, of all of the other simulation codes mentioned in Sec. 2.5.3, only 2 are full-f (namely, GYSELA and GT5D), of which only one can simulate non-circular magnetic configurations (GT5D), making it the only one which would have been suitable for our needs in this endeavour. While all those codes have the common aim of understanding anomalous transport, the core difference between the full-f codes and the others resides in where they expect to find the answer.

From the "classical" models proposed by MHD, the introduction of neo-classical transport has been a jump towards the much smaller scales of space and time that correspond to the dynamics of ions. Figure 2.6 gave an illustration of the scales involved. To understand anomalous transport, the most common approach seems to favour the idea of taking another step towards the even smaller scales of the dynamics of electrons. The modern δf codes therefore focus on a smaller portion of the whole plasma so they can study it at the finest of scales, beyond the ion gyroradius. On the other hand, the opposite approach, which we have followed throughout this study, instead tries to fill the gap left between the macroscopic scale of MHD and the scale of ion dynamics.

While studies on magnetic shaping effects have been performed using local codes, they can only "perceive" the shape of the magnetic field in the specific area of the plasma they chose to study. The results obtained around the ITG instability in the previous chapter of this thesis clearly shows that δf and full-f codes can lead to very different results, even if in the so-called linear phase and even when they are simulating the same whole domain. If this result is far from meaning that

anomalous transport arises solely from an interaction between macroscopic MHD and the microscopic dynamics at the ion gyroradius scale, it does show that some of the physics is still to be understood in this area.

6.3 To go beyond

As a last note, I would like to highlight some limitations of the work conducted and the possible future studies that could be undertaken in its continuity.

During the 5 years it took this project to come to fruition, a lot of time was spent writing a functional simulation code, leading to a rush by the end. In particular, while nonlinear ITG simulations were working, their inclusion in the study on magnetic shaping effects of the previous chapter would have required the proper usage of heat sources and sink so that heat diffusivities could be measured numerically. Unfortunately, some issues still remain with the heat diffusivities measured, leaving them unsuitable for a publication or the inclusion in this thesis. Fixing this lack would be of great interest and could confirm the strong dependence of heat diffusivities on elongation that the mixing length formula suggested, much stronger than in the other paper, close to experimental measurements.

More generally, the previous section mentioned the difference of philosophy between local codes, focusing on extension of neo-classical transport towards smaller scales, and full-f codes, rather trying to connect dynamics at the ion gyroradius with macroscopic scales. However, this does not mean that researchers developing full-f codes see the dynamics at smaller scales as having no influence, to the contrary! Some work has already started to implement kinetic electrons in this version GKNET, with the aim of studying TEM or ETG instabilities, and more generally to capture the dynamics at the scale of the electron gyroradius.

Bibliography

- [Anderson et al., 2000] Anderson, J., Nordman, H., and Weiland, J. (2000). Effects of non-circular tokamak geometry on ion-temperature-gradient driven modes. *Plasma physics and controlled fusion*, 42(5):545.
- [Angelino et al., 2008] Angelino, P., Garbet, X., Villard, L., Bottino, A., Joliet, S., Ghendrih, P., Grandgirard, V., McMillan, B., Sarazin, Y., Dif-Pradalier, G., et al. (2008). The role of plasma elongation on the linear damping of zonal flows. *Physics of Plasmas*, 15(6):062306.
- [Belli et al., 2008] Belli, E., Hammett, G., and Dorland, W. (2008). Effects of plasma shaping on nonlinear gyrokinetic turbulence. *Physics of Plasmas*, 15(9):092303.
- [Brizard and Hahm, 2007] Brizard, A. and Hahm, T. (2007). Foundations of nonlinear gyrokinetic theory. *Reviews of modern physics*, 79(2):421.
- [Candy and Waltz, 2003] Candy, J. and Waltz, R. (2003). An eulerian gyrokinetic-maxwell solver. *Journal of Computational Physics*, 186(2):545–581.
- [Cary and Littlejohn, 1983] Cary, J. R. and Littlejohn, R. G. (1983). Noncanonical hamiltonian mechanics and its application to magnetic field line flow. *Annals of Physics*, 151(1):1–34.
- [Chang et al., 2009] Chang, C., Ku, S., Diamond, P., Lin, Z., Parker, S., Hahm, T., and Samatova, N. (2009). Compressed ion temperature gradient turbulence in diverted tokamak edge a. *Physics of Plasmas*, 16(5):056108.
- [Dif-Pradalier et al., 2008] Dif-Pradalier, G., Grandgirard, V., Sarazin, Y., Garbet, X., and Ghendrih, P. (2008). Defining an equilibrium state in global full-f gyrokinetic models. *Communications in Nonlinear Science and Numerical Simulation*, 13(1):65–71.
- [Dorland et al., 2000] Dorland, W., Jenko, F., Kotschenreuther, M., and Rogers, B. (2000). Electron temperature gradient turbulence. *Physical Review Letters*, 85(26):5579.

- [Grandgirard et al., 2016] Grandgirard, V., Abiteboul, J., Bigot, J., Cartier-Michaud, T., Crouseilles, N., Dif-Pradalier, G., Ehrlacher, C., Esteve, D., Garbet, X., Ghendrih, P., et al. (2016). A 5d gyrokinetic full-f global semi-lagrangian code for flux-driven ion turbulence simulations. *Computer Physics Communications*, 207:35–68.
- [Grandgirard et al., 2006] Grandgirard, V., Sarazin, Y., Garbet, X., Dif-Pradalier, G., Ghendrih, P., Crouseilles, N., Latu, G., Sonnendrücker, E., Besse, N., and Bertrand, P. (2006). Gysela, a full-f global gyrokinetic semi-lagrangian code for itg turbulence simulations. In *Aip conference proceedings*, volume 871, pages 100–111. AIP.
- [Hilscher, 2013] Hilscher, P. P. (2013). *Study of multi-scale interaction and dissipation based on gyro-kinetic model in fusion plasmas*. PhD thesis, Kyoto University.
- [Idomura, 2016] Idomura, Y. (2016). A new hybrid kinetic electron model for full-f gyrokinetic simulations. *Journal of Computational Physics*, 313:511–531.
- [Idomura et al., 2008] Idomura, Y., Ida, M., Kano, T., Aiba, N., and Tokuda, S. (2008). Conservative global gyrokinetic toroidal full-f five-dimensional vlasov simulation. *Computer Physics Communications*, 179(6):391–403.
- [Idomura et al., 2003] Idomura, Y., Tokuda, S., and Kishimoto, Y. (2003). Global gyrokinetic simulation of ion temperature gradient driven turbulence in plasmas using a canonical maxwellian distribution. *Nuclear Fusion*, 43(4):234.
- [Jenko, 2000] Jenko, F. (2000). Massively parallel vlasov simulation of electromagnetic drift-wave turbulence. *Computer physics communications*, 125(1-3):196–209.
- [Jenko and Kendl, 2002] Jenko, F. and Kendl, A. (2002). Stellarator turbulence at electron gyroradius scales. *New Journal of Physics*, 4(1):35.
- [Jolliet et al., 2007] Jolliet, S., Bottino, A., Angelino, P., Hatzky, R., Tran, T., McMillan, B., Sauter, O., Appert, K., Idomura, Y., and Villard, L. (2007). A global collisionless pic code in magnetic coordinates. *Computer Physics Communications*, 177(5):409–425.
- [Kinsey et al., 2007] Kinsey, J., Waltz, R., and Candy, J. (2007). The effect of plasma shaping on turbulent transport and $\mathbf{e} \times \mathbf{b}$ shear quenching in nonlinear gyrokinetic simulations. *Physics of Plasmas*, 14(10):102306.
- [Lao et al., 1985] Lao, L., John, H. S., Stambaugh, R., Kellman, A., and Pfeiffer, W. (1985). Reconstruction of current profile parameters and plasma shapes in tokamaks. *Nuclear fusion*, 25(11):1611.

-
- [Lee, 1983] Lee, W. (1983). Gyrokinetic approach in particle simulation. *The Physics of Fluids*, 26(2):556–562.
- [Lin et al., 1998] Lin, Z., Hahm, T. S., Lee, W., Tang, W. M., and White, R. B. (1998). Turbulent transport reduction by zonal flows: Massively parallel simulations. *Science*, 281(5384):1835–1837.
- [Lin and Lee, 1995] Lin, Z. and Lee, W. (1995). Method for solving the gyrokinetic poisson equation in general geometry. *Physical Review E*, 52(5):5646.
- [Miller et al., 1998] Miller, R., Chu, M., Greene, J., Lin-Liu, Y., and Waltz, R. (1998). Noncircular, finite aspect ratio, local equilibrium model. *Physics of Plasmas*, 5(4):973–978.
- [Obrejan et al., 2017] Obrejan, K., Imadera, K., Li, J., and Kishimoto, Y. (2017). Development of a new zonal flow equation solver by diagonalisation and its application in non-circular cross-section tokamak plasmas. *Computer Physics Communications*, 216:8–17.
- [Obrejan et al., 2015] Obrejan, K., Imadera, K., Li, J.-Q., and Kishimoto, Y. (2015). Development of a global toroidal gyrokinetic vlasov code with new real space field solver. *Pasma Fusion Res.*, 10:42.
- [Peeters et al., 2009] Peeters, A., Camenen, Y., Casson, F. J., Hornsby, W., Snodin, A., Strintzi, D., and Szepesi, G. (2009). The nonlinear gyro-kinetic flux tube code gkw. *Computer Physics Communications*, 180(12):2650–2672.
- [Rosenbluth and Hinton, 1998] Rosenbluth, M. and Hinton, F. (1998). Poloidal flow driven by ion-temperature-gradient turbulence in tokamaks. *Physical review letters*, 80(4):724.
- [Sugama and Watanabe, 2006] Sugama, H. and Watanabe, T.-H. (2006). Collisionless damping of zonal flows in helical systems. *Physics of Plasmas*, 13(1):012501.
- [Szepesi, 2012] Szepesi, G. (2012). *Derivation of the fully electro-magnetic, non-linear, gyrokinetic Vlasov–Maxwell equations in a rotating frame of reference for GKW with Lie transform perturbation method.*
- [Takeda and Tokuda, 1991] Takeda, T. and Tokuda, S. (1991). Computation of mhd equilibrium of tokamak plasma. *Journal of computational physics*, 93(1):1–107.
- [Taylor and Hastie, 1968] Taylor, J. and Hastie, R. (1968). Stability of general plasma equilibria-i formal theory. *Plasma Physics*, 10(5):479.

BIBLIOGRAPHY

- [Weiland, 1999] Weiland, J. (1999). *Collective Modes in Inhomogeneous Plasmas: Kinetic and Advanced Fluid Theory*. CRC Press.
- [Wettengel, 2017] Wettengel, J. (2017). German carbon emissions rise in 2016 despite coal use drop. <https://www.cleanenergywire.org/news/german-carbon-emissions-rise-2016-despite-coal-use-drop>. Accessed: 2017-06-20.
- [Xiao and Catto, 2006] Xiao, Y. and Catto, P. J. (2006). Plasma shaping effects on the collisionless residual zonal flow level. *Physics of plasmas*, 13(8):082307.