

# **Global Flux-Driven Simulations of Ion Temperature-Gradient and Trapped-Electron Modes Driven Turbulence With an Improved Multithreaded Gyrokinetic PIC Code**

**Thèse N° 9575**

Présentée le 6 septembre 2019  
à la Faculté des sciences de base  
SPC - Théorie  
Programme doctoral en physique

pour l'obtention du grade de Docteur ès Sciences

par

**Emmanuel LANTI**

Acceptée sur proposition du jury  
Prof. S. Manley, présidente du jury  
Prof. L. Villard, Dr. S. Brunner, directeurs de thèse  
Dr Y. Idomura, rapporteur  
Dr Y. Sarazin, rapporteur  
Dr S. Coda, rapporteur

2019



# Abstract

The performance of tokamak fusion plasmas is heavily linked to the radial heat and particle transport, which is known to be mainly produced by turbulence driven by micro-instabilities. Understanding such processes is thus of key importance for the design and operation of future fusion reactors such as ITER. One of the most commonly used tools to study turbulent transport in the very hot core of tokamaks is gyrokinetic theory, which aims at reducing the physical complexity by averaging out the fast gyromotion, while keeping its averaged effect. The ORB5 program is a numerical code solving the gyrokinetic Maxwell-Boltzmann system of equations to study various phenomena such as turbulent transport. It is a global particle-in-cell code using a finite element representation with B-spline basis function for the fields and various noise control schemes.

The main objective of this work is to improve the performance and physical model of ORB5 in order to be able to carry out simulations relevant for the Tokamak à Configuration Variable (TCV) at the Swiss Plasma Center. This is done in three major steps.

First, the code undergoes a complete refactoring and optimization process with the aim of increasing its maintainability and performance. Using a simple test bed being a simplification of the PIC method while retaining the key elements, various algorithms and parallelization schemes are developed. Furthermore, it is shown that one of the most critical kernels of the PIC algorithm, the charge deposition, can be accelerated by a factor 2.4 in ORB5.

Second, the hybrid electron model, which is used to simulate certain electron modes at a lower numerical cost than the fully kinetic electron model, needs to be corrected. Indeed, in this model, trapped electrons are kinetic while passing electrons have an adiabatic response. This is problematic because it does not satisfy the ambipolarity condition and produces spurious sources of particles and momentum. A corrected model in which the flux-surface averaged contribution of the passing electrons is treated as kinetic is developed and implemented in ORB5. As a first step, an approximation of the flux-surface average consisting of a Fourier filter retaining only the  $n = m = 0$  mode is tested. Linearly, both the proper flux-surface average and its  $n = m = 0$  approximation lead to an accurate description of the geodesic acoustic mode dynamics. Nonlinearly, the  $n = m = 0$  model exhibits interesting physical features such as sharp radial structures at the mode rational surface  $q = 1$  and a transport regime transition. To conclude this chapter, a first attempt at studying TCV transport stiffness is shown.

Finally, a moment-conserving heat source is implemented in the ORB5 code to allow

---

for flux-driven simulations. The heat source conservation properties are verified and it is shown that conservation of density, parallel momentum, and zonal flows is satisfied up to machine precision. A “mode-switching” approach consisting of using an effective heat source computed from a gradient-driven simulation as an input for the flux-driven run is also tested and consistent results are obtained between gradient-driven, flux-driven, and “mode-switched” simulations. Finally, the heat source is characterized, along with variations, using a 2D phase-space diagnostics. It is shown that in some cases, the use of this fixed heat source can lead to a sampling problem causing an artificial source/sink of moments.

**Keywords:** plasma, fusion, tokamak, gyrokinetic, PIC, micro-instabilities, turbulence, optimization, flux-driven, hybrid electron model

# Résumé

La performance des plasmas de fusion générés dans les tokamaks est largement déterminée par le transport radial de chaleur et de particules. Il est maintenant admis que ce transport provient majoritairement de la turbulence induite par certaines micro-instabilités. Il est donc primordial d'étudier ces phénomènes de transport turbulent, afin d'optimiser la conception et de maîtriser le fonctionnement des futurs réacteurs à fusion nucléaire, comme celui d'ITER. La théorie gyrocinétique est certainement l'outil le plus utilisé pour étudier la dynamique des plasmas peu collisionnels présents dans le cœur des tokamaks. Le but de cette théorie est de réduire la complexité d'un modèle cinétique en moyennant, de manière consistante, le mouvement cyclotronique rapide de Larmor. Le programme ORB5 est un code de simulation numérique utilisé pour résoudre le système d'équations de Maxwell-Boltzmann, afin d'étudier divers phénomènes tels que le transport turbulent dans le cœur des tokamaks. Basé sur la méthode des éléments finis avec une représentation B-spline pour les champs électromagnétiques et la représentation en particules numériques pour les fonctions de distribution, ORB5 est un code global et possède différentes méthodes de contrôle du bruit numérique.

Le but principal de cette thèse est d'améliorer la performance et le modèle physique d'ORB5, afin d'être capable de simuler des plasmas similaires à ceux produits dans le Tokamak à Configuration Variable (TCV) du Swiss Plasma Center. Trois objectifs majeurs sont donc définis.

Dans un premier temps, un réusinage complet du code ainsi qu'une procédure d'optimisation de sa performance sont effectués. Pour ce faire, un "banc d'essai", consistant en un code PIC simplifié, bien que contenant toutes les méthodes essentielles, est développé pour faciliter l'implémentation et le test de nouveaux algorithmes et procédures de parallélisation. Il est démontré, à titre d'exemple, que la déposition de charge, méthode au cœur de l'algorithme PIC, peut être accélérée d'un facteur 2.4 dans ORB5.

Le deuxième objectif, quant à lui, consiste à corriger le modèle à électrons hybrides implémenté dans ORB5. En effet, dans le modèle initial, les électrons passants sont traités de manière adiabatique alors que les électrons piégés sont traités de manière cinétique. La description adiabatique des électrons passants fait que ce modèle ne satisfait pas la condition d'ambi-polarité et produit des sources fictives de particules et de quantité de mouvement. Un modèle corrigé, dans lequel la contribution de la moyenne de surface de flux des électrons passants est traitée de manière cinétique, est développé et implémenté dans ORB5. En première

---

approximation, un second modèle, dans lequel la moyenne de surface de flux est remplacée par un filtre de Fourier ne retenant que le mode  $n = m = 0$ , est également implémenté. Il est montré que ces deux modèles décrivent précisément les propriétés linéaires de la dynamique des GAMs. D'autre part, des simulations non linéaires, présentant des particularités intéressantes telles que des fines structures radiales à la surface de mode rationnelle  $q = 1$ , ainsi qu'une transition de régime de transport, sont également présentées. Pour conclure ce chapitre, une première étude préliminaire de la "rigidité" du transport dans TCV.

Finalement, une source de chaleur avec des propriétés de conservation des moments est implémentée dans ORB5 permettant de faire des simulations à flux imposé. Les propriétés de conservation de la source sont vérifiées jusqu'à la précision machine. De plus, la méthode de "changement de mode", consistant à démarrer une simulation à flux imposé à partir de la source de chaleur effective calculée à l'aide d'une simulation à gradient imposé, est présentée et il est démontré qu'elle donne des résultats consistants avec des simulations à flux et à gradient imposés. Pour conclure, la source de chaleur et quelques variantes sont caractérisées par le truchement d'un diagnostic 2D dans l'espace de phase. Il est notamment montré que, dans certains cas, l'utilisation de cette source de chaleur mène à un problème d'échantillonnage de l'espace de phase, causant la création d'une source artificielle de moments.

**Mots Clés :** plasma, fusion, tokamak, gyrocinétique, PIC, micro-instabilités, turbulence, optimisation, simulation à flux imposé, modèle à électrons hybrides

# Remerciements

Durant ces quatre dernières années, j'ai eu le privilège de travailler dans un environnement passionnant et surtout avec des gens passionnés! Cela m'a permis d'énormément grandir tant sur le plan scientifique que personnel. Je tente ici, de remercier chaque personne qui m'a permis d'en arriver jusque là, même si ces quelques lignes ne seront jamais à la hauteur de leur aide et de leur soutien.

Tout d'abord, je tiens à remercier chaleureusement Laurent et Stephan, mes directeurs de thèse. Merci de m'avoir fait confiance et de m'avoir accueilli dans votre groupe de gyrocinétique. Bien que novice dans le domaine à l'époque, vous avez su me proposer des projets passionnants et surtout me laisser suffisamment libre pour m'épanouir complètement. Merci pour toutes les discussions pas toujours scientifiques et surtout pour votre accueil sans faille. Je sais que vous êtes souvent très occupés par vos tâches respectives, mais je n'ai jamais trouvé votre porte fermée. Vous avez su créer un environnement de travail enrichissant dans lequel j'ai pu m'épanouir et me suis toujours senti écouté. Merci d'avoir relu sans relâche mon manuscrit et autres publications et d'avoir contribué à largement améliorer la qualité de ces derniers.

Je ne peux écrire ces quelques lignes sans une pensée pour Minh. Nous avons collaboré très étroitement pendant ma première année de thèse et je te dois énormément. J'ai eu la chance d'apprendre considérablement de choses grâce à tes connaissances pointues sur la physique numérique et le HPC. Je dois également relever ton optimisme et ta passion pour ton travail qui ont toujours été plus fort que tout.

Je tiens également à remercier chaleureusement Stefano Coda, Yanick Sarazin, et Yasuhiro Idomura qui ont très généreusement accepté de faire partie de mon jury de thèse. Merci pour tous les conseils qui m'ont permis d'améliorer ce manuscrit.

J'ai eu l'occasion de côtoyer plusieurs personnes de l'équipe ORB5 avec qui j'ai toujours eu des discussions passionnantes. Merci à Natalia, Ben, Claudio, Andreas, Alberto, Thomas, Alexey, Alessandro et Ivan.

Merci également à la direction du SPC (anciennement CRPP), Ambrogio, Yves, Paolo, et Quang de m'avoir accueilli pour cette thèse et de m'avoir permis de partager notre passion

## Remerciement

---

lors des nombreuses visites de TCV.

Je ne peux pas oublier de remercier Stefano Alberti, car c'est lui qui m'a transmis l'envie de faire de la physique des plasmas pendant son cours "Plasma I". Merci également à Paolo, Sébastien et Fabio de m'avoir guidé pendant mes premiers pas au CRPP pour mes TPIV et mon projet de Master.

Un énorme merci à la fine équipe: Jérémy, Stéphane et Noé. Vous m'avez accompagné pendant tout mon temps au SPC et vous avez largement contribué à le rendre inoubliable. Merci pour le soutien, les blagues un peu lourdes d'un grand sage que je ne citerai pas, les bons moments passé pendant et hors des heures de travail. Les apéros hebdomadaires pendant lesquels on refaisait le monde.

Je me dois également de penser aux anciens membres du groupe. Tout d'abord à Gabriele qui était en plus mon collègue de bureau. Merci pour ton éternel sourire et ta joie de vivre. Ces quelques temps passés dans ce bureau sont inoubliables! Merci également à Julien, mon mentor ORB5. Tu m'as initié au code et tu m'as motivé à travailler avec bien que j'essayais de repousser ce moment. Tu m'as énormément aidé à la compréhension de ORB5 et je t'en remercie du fond du coeur. Finalement, je tiens à souhaiter bonne chance à notre nouveau post-doc, Peter!

Merci à mes collègues doctorants, post-docs, et autres, de m'avoir aidé et d'avoir rendu mes moments ici très agréables: Matteo, Arsène, Ajay, Rogério, Sam, Hugo, Hugo, Dahye, Oulfa, Antoine, Paola, Hamish, Umar, etc.

Un énorme merci aux personnes de l'ombre: Edith, Tushi, Roxane, et Christine qui, souvent sans la reconnaissance qu'elles méritent, veillent à ce que le SPC tourne rond. Merci de m'avoir aidé et conseillé pour les diverses tâches administratives. Merci à Pierre pour toutes les discussions liées, ou non, à l'informatique, mais surtout pour ta disponibilité sans faille et ton efficacité. Merci à Christian S. d'avoir répondu à mes nombreuses sollicitations concernant les différents OTP.

Même si nous n'avons pas travaillé directement ensemble, on a néanmoins partagé d'excellents moments à la pause et autre repas du labo. Merci Steve, Christan M., Omar, Nelson, Stefano B., Thierry, Damien, et les autres qui me sont malheureusement sorti de l'esprit.

Merci à mes amis de longue date, Raphaël M., Christian, Bruno et Michaël, pour votre soutien et surtout tous les moments qu'on a passés ensemble!

Merci également à Raphaël S., Yannick, Randy, Sonia, et Erwin pour tous les excellents moments passé au sein de Conjonctive. Faire du bruit avec vous m'a toujours aidé à faire le vide et à penser à autre chose. Même si je râlais souvent pour les concerts dans des endroits bizarres, squats, et j'en passe, ce sont finalement que des merveilleux souvenirs.

Merci à la team brioche, Oliver, Olivier, Raphaël, Christian, Chris, Séb, Gino, Corentin, Aline, Jonathan et Pietro. Un énorme merci à toute l'équipe du SDIS Nyon-Dôle avec qui je m'épanouis dans un registre totalement différent.



Finale­ment, le plus im­por­tant. Un énor­me merci à toute ma fa­mille pour m’au­oir per­mis d’en ar­ri­ver là et pour vo­tre sou­tien sans fail­le. C’é­tail pas fa­cile de m’en­ten­dre me plain­dre, sur­tout quand vous n’au­iez au­cune idée de ce que je fa­isais, mais vous au­ez tou­jours é­té là. Mer­ci Pa­pa, Ma­man, Mar­cella et Gas­pard. La vie n’a pas tou­jours é­té fa­cile, mais vous au­ez tous con­tri­bué, à vo­tre ma­nière, à me la ren­dre la plus belle pos­si­ble. Mer­ci à Flo­rian, Dé­bo­rah, Ma­thieu et le pe­tit Ke­ryan. Mer­ci à ma Dé­bo­rah de me sup­por­ter de­puis plus de dix ans. Mê­me si je ne le dis pas as­sez, je n’au­rais rien fait de tout ça sans toi. Tu m’as tou­jours pou­ssé à al­ler plus loin et c’est en gran­de par­tie grâce à toi que j’en suis là.

Mer­ci !

*Lausanne, le 10 juillet 2019*

E. L.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Résumé</b>	<b>v</b>
<b>Remerciement</b>	<b>vii</b>
<b>Contents</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Basic concepts of plasma and tokamak physics . . . . .	6
1.1.1 Particle and fluid drifts . . . . .	6
1.1.2 Trapped particles . . . . .	8
1.1.3 Tokamak magnetic equilibrium . . . . .	9
1.1.4 Microinstabilities . . . . .	12
1.1.5 Physics of zonal flows and geodesic acoustic modes . . . . .	18
1.2 Contribution and outline of the thesis . . . . .	20
<b>2 The ORB5 gyrokinetic model</b>	<b>23</b>
2.1 Magnetic geometry, coordinate system, and normalization . . . . .	23
2.2 Gyrokinetic equations for fields and particles . . . . .	24
2.2.1 Quasineutrality and Ampère equations . . . . .	27
2.2.2 Nonlinear gyrokinetic Vlasov equation . . . . .	28
2.3 Variants of the physical models . . . . .	30
2.3.1 Long wavelength approximation . . . . .	30
2.3.2 Padé approximation . . . . .	30
2.3.3 Adiabatic electron model . . . . .	31
2.3.4 Hybrid electron model . . . . .	32
2.3.5 Summary of the models . . . . .	34
2.4 $\delta f$ method and background distribution functions . . . . .	35
2.5 Strong flows . . . . .	37
2.5.1 Global gyrokinetic equilibria for rotating plasmas . . . . .	38
2.6 Collisions . . . . .	39
	xi

## Contents

---

2.7	Conservation laws and diagnostics . . . . .	40
2.8	Summary of the chapter . . . . .	43
<b>3</b>	<b>ORB5 numerical implementation, refactoring, and optimization</b>	<b>45</b>
3.1	Numerical implementation . . . . .	45
3.1.1	Discretization of $\delta f$ and equations of motion . . . . .	46
3.1.2	Quasineutrality and Ampère equations . . . . .	51
3.1.3	Noise control techniques . . . . .	54
3.1.4	Heating operators . . . . .	61
3.1.5	Parallelization . . . . .	62
3.2	Refactoring and optimization for multicore platforms . . . . .	63
3.2.1	Optimization techniques used to improve the code performance . . . . .	64
3.3	The Pic_Engine: a simple test bed for algorithmic testing . . . . .	69
3.3.1	The Pic_Engine platform . . . . .	69
3.3.2	Performance analysis . . . . .	76
3.4	Application to the ORB5 code: the color scheme . . . . .	88
3.4.1	Effect of sorting on the charge deposition step . . . . .	89
3.4.2	OpenMP performance and the color scheme . . . . .	90
3.5	Summary of the chapter . . . . .	93
<b>4</b>	<b>Improved hybrid electron model</b>	<b>97</b>
4.1	Introduction . . . . .	97
4.2	Validation of the upgraded hybrid electron model . . . . .	99
4.2.1	Linear validation: Rosenbluth-Hinton test . . . . .	99
4.2.2	Nonlinear verification . . . . .	102
4.3	Effect of the $n = m = 0$ hybrid electron model on turbulence properties . . . . .	106
4.4	Core and pedestal TEM turbulent transport . . . . .	114
4.5	Summary of the chapter . . . . .	118
<b>5</b>	<b>Flux driven simulations</b>	<b>121</b>
5.1	Summary of the different heat sources . . . . .	122
5.1.1	Krook operator heat source . . . . .	122
5.1.2	Fixed heat source . . . . .	123
5.1.3	Adaptive Krook heat source . . . . .	125
5.2	Switching from gradient- to flux-driven simulations . . . . .	125
5.2.1	Illustration of the mode-switching approach with ITG and TEM driven simulations . . . . .	127
5.3	Heat source effect on the phase space . . . . .	134
5.4	Summary of the chapter . . . . .	143

<b>6 Conclusion and Outlook</b>	<b>147</b>
6.1 Summary . . . . .	147
6.2 Outlook . . . . .	149
6.2.1 Refactoring and optimization of ORB5 . . . . .	149
6.2.2 Hybrid electron model . . . . .	150
6.2.3 Flux-driven simulations . . . . .	150
6.2.4 General outlook . . . . .	150
<b>A Parallelization algorithms</b>	<b>151</b>
A.1 Sorting . . . . .	151
A.2 Charge assignment . . . . .	153
<b>Bibliography</b>	<b>157</b>
<b>Curriculum Vitae</b>	<b>172</b>



# 1 Introduction

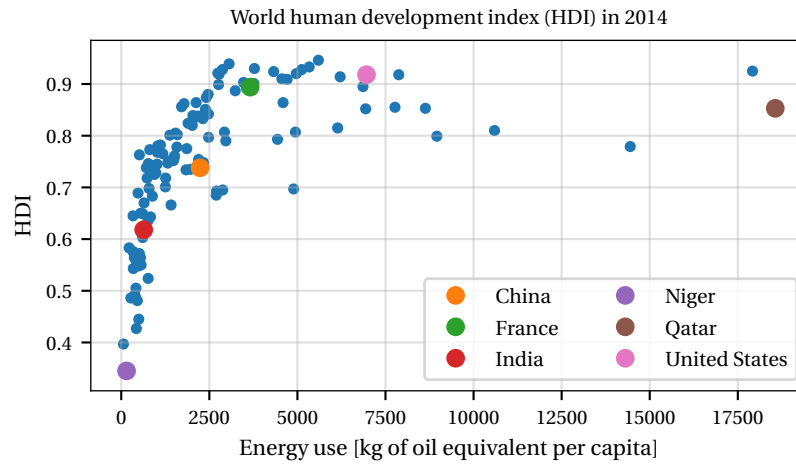
Energy production is for sure one of the main challenges that mankind has to face in the near future. Whether it is used for heating, for mechanical work, or to produce electricity, energy has always been a driver of human progress. In the 19th century, the steam engine—first human-controlled type of energy—played an important role in the Industrial Revolution, which led to a sustained growth of the living standards. Even today, there is a clear correlation between energy consumption and human development, Figure 1.1. Usually, an abundant source of energy provides, directly or indirectly, a higher life expectancy, better education and, as a result, a higher economic growth. However, despite the well recognized importance of energy production in the current society, there is presently no satisfying solution to meet the increasing demand with an economically viable and environmentally-friendly source of energy. On the one hand, the needs grow exponentially due to the higher demands in the industrialized parts of the world combined with the needs of rapidly developing countries such as China and India and, on the other hand, the negative effects of greenhouse gases on the environment put a severe constraint on the energy production.

In 2016, a bit less than 60% of the total electricity production in the Organisation for Economic Cooperation and Development (OECD) came from fossil fuels such as coal, oil and natural gas [3]. These fuels are generally not expensive but the process of converting them into electrical energy is not very efficient. Furthermore, they allow for a continuous and remote production in large power plants each able to supply hundreds of thousands of people. However, burning carbon-based fuels inevitably produces carbon dioxide—and other harmful by-products of the combustion—which is one of the responsible gases for the greenhouse effect.

Alternatives that produce less or no greenhouse gases *during their life cycle*<sup>1</sup> are the renewable energies such as wind, solar and geothermal. The main problem of these sources is that they are intermittent and/or need certain particular conditions to work, which cannot be controlled. For example, geothermal plants and hydro-electric dams can only be built in

---

<sup>1</sup>To be accurate, one should also take into account the production and dismantling energy cost. For example, most of the solar panel production is currently made in China using electricity coming mostly from coal- and gas-fired plants.



**Figure 1.1** : World human development index (HDI) as a function of the energy use. The HDI is a composite index built from the life expectancy, education, and per capita income indicators. It provides a measure of the well-being of a country. Source [1, 2].

very localized geographical areas and solar panels do not produce any electricity at night. Energy storage and transportation would be a solution to these problems but, unfortunately, the current technology does not allow for an efficient and low-cost storage/transport.

Finally, nuclear fission is another alternative to fossil fuels. Heat is extracted from the fission of heavy atoms such as the uranium isotope  $U^{235}$  and transformed into electrical energy. This technology, which is often of major public concern, produces almost no carbon dioxide nor emits harmful particles. Furthermore, because of the high energy density available in the nuclear combustibles, the current fuel supplies are sufficient to provide energy for hundreds of years and even more, if one uses more advanced technologies such as breeder reactors. The main concern about nuclear fission is without any doubt related to safety. Even though there is no statistical evidence that nuclear fission is more dangerous than fossil fuels, there is a large public concern about this technology, especially after the Tchernobyl and Fukushima events in 1986 and 2011, respectively. Countries such as Switzerland and Germany have decided to renounce nuclear fission after the latter event. Still related to safety, the treatment of radioactive waste, the decommissioning of power plants, and the problem of nuclear proliferation are other important issues of nuclear fission.

In summary, the future of energy production should be environment-friendly, safe, and not expensive. This will most certainly consist of a mix of renewable energies and an abundant type of energy serving as a background continuous production to compensate for the intermittent nature of renewable energies.

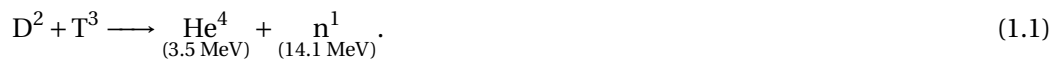
One of the most promising candidates for supplying a large and steady amount of elec-



---

tricity with a minimal impact on the environment is nuclear fusion, a process consisting of fusing light nuclei and harvest the produced energy. For power plant applications, deuterium and tritium—two hydrogen isotopes—are typically used as fuels in the so-called D-T reaction. While deuterium is available in large quantity in sea water, tritium is sparsely found in nature due to its short half-life,  $\sim 12$  years. To overcome this problem, a secondary reaction consisting of producing tritium atoms from lithium will be used in future fusion reactors. The main advantage of this technology is that everything that enters and exits the reactor is safe and non radioactive. Furthermore, there is no risk of uncontrolled chain reactions and nuclear proliferation as it is the case for nuclear fission. The main challenge of nuclear fusion, however, is to overcome the scientific and engineering difficulties inherent to the fusion process in an economically viable way.

The D-T reaction consumes one nucleus each of deuterium and tritium and releases a helium nucleus ( $\alpha$  particle) and a neutron:



The huge amount of energy released by fusion reactions is around four million times higher than the one from burning coal and four times higher than the energy released by nuclear fission reactions at equal mass. This clearly makes nuclear fusion an interesting candidate for mass production of energy. Unfortunately, the fusion process is difficult to achieve because of the Coulomb repulsion between the two positively charged nuclei. Typically, the optimal temperature which maximizes the probability of fusion events is of the order of 10 keV (around  $100'000'000$  K). At such temperature, the D-T gas is fully ionized, i.e. it is in the plasma state. Despite its apparent simplicity, the plasma has a very complicated self-consistent behaviour, because of the presence of electromagnetic interactions. Indeed, the charged particles composing the plasma are affected by the electromagnetic fields, but are also sources of such fields.

In a fusion reactor, the very hot plasma must be confined in such a way as to avoid contact with the vessel to prevent the destruction of the latter. On Earth, this can be achieved using either inertial confinement or magnetic confinement. The basic principle of the latter is to exploit the Larmor motion of charged particles immersed in a magnetic field; the excursion of charged particles is constrained perpendicular to the magnetic field lines. The most advanced concept of magnetic confinement device is the tokamak, a toroidal vessel using helicoidal magnetic field lines to confine the plasma. The twisted field is a superposition of a toroidal magnetic field produced by magnetic coils and a poloidal magnetic field produced by an induced toroidal electric current<sup>2</sup>. An alternative to the tokamak is the so-called stellarator which uses an arrangement of very complex magnetic coils to generate the twisted field magnetic lines. The present work will entirely be dedicated to tokamak plasmas.

---

<sup>2</sup>Other coils may be required for shaping and stabilization

One of the most important criteria that determines the output power of a tokamak is the triple product, also called Lawson criterion [4]:

$$nT\tau_E \geq 10^{21} \text{ keV} \cdot \text{s} \cdot \text{m}^{-3}. \quad (1.2)$$

Basically, this condition states that the product of plasma density  $n$ , temperature  $T$ , and energy confinement time  $\tau_E$  must be greater than a certain value determining the output power gain. In other words, for fusion reactions to take place frequently, there should be enough plasma particles close to each other, with a sufficient energy to overtake the Coulomb repulsion, and for a sufficiently long period of time. Ultimately, most of the research on tokamak confinement comes down to increasing the triple product while avoiding operational limits such as pressure and density limits [5, 6].

Transport theory aims at understanding energy confinement and is therefore one of the most active fields of research in magnetic confinement fusion. An important characteristics of tokamak plasmas is that heat transport is almost always dominated by turbulence-driven processes; this is the so-called anomalous transport. The usual measured levels of heat transport in tokamaks are orders of magnitude higher than the one predicted from collisional transport estimations—either in cylindrical geometry (classical transport) or in toroidal geometry (neo-classical transport). Anomalous transport is governed by highly nonlinear processes, with multiple sources of drive and regulation mechanisms, occurring at different spatio-temporal scales. Due to this inherent complexity, the description of turbulent transport requires advanced mathematical modelling. In the very hot core of tokamaks, a kinetic theory is necessary to correctly describe the different processes taking place. In general, the evolution of the distribution function of each plasma species is computed in the presence of collisions and sources. However, the six-dimensional nature of the phase space (three dimensions for configuration space and three for the velocity) and the wide range of spatial and temporal scales are very challenging obstacles to overcome. Thanks to the anisotropic nature of tokamak turbulence and the scale separation observed in the core of tokamaks, it is possible to make some approximations which allow one to reduce the complexity of the model. For example, gyrokinetic theory consists of averaging out the fast cyclotron motion of plasma particles while taking care to keep the important effects related to it. In this way, the problem dimensionality is reduced from 6D to 5D and the fast cyclotron time scale is removed consistently from the system. Even with this simplification, the complete description of real tokamaks remains a complex task to carry out and further approximations are often used, e.g. adiabatic electrons, collisionless system, electrostatic limit, etc.

In general, solving the gyrokinetic Vlasov-Maxwell (or Boltzman-Maxwell) system of equations is no easy task and cannot be done without the help of numerical codes. In the past decades, a lot of resources have been dedicated to the development of gyrokinetic codes. On the one hand, the physical model is constantly improved, allowing for more realistic descriptions and, on the other hand, numerical algorithms and high performance computers

---

(HPC) are developed to allow for more complex simulations. One of the particularities of magnetized plasma turbulence is that geometric effects due to the background magnetic field play an important role. Because of that, the global approach takes into account the geometry of the whole domain and background profiles are generally left free to evolve. A possible simplification that is typically made is the local, or flux-tube limit. In this approximation, one restricts the simulation domain to a small flux tube following a magnetic field line around the tokamak. The size of the flux-tube is supposed to be small enough so that the background profile variations can be neglected but large enough compared to the radial turbulence correlation length so that radial periodicity can be assumed. These conditions can be difficult to satisfy in some cases where long radial structures such as streamers are excited. Nevertheless, the flux-tube limit is very useful as the computational requirements are much smaller than for a global approach. Furthermore, it was shown that local and global codes results agree in the limit of vanishing  $\rho_*$  [7, 8], where  $\rho_*$  is the ion sound Larmor radius to tokamak minor radius ratio. Another important difference between codes is the heating scheme. In a typical global nonlinear simulation, the marginal stability is quickly reached in the absence of a heat source because of the temperature profile relaxation. While these transient simulations have been used to benchmark codes [9], it is difficult to make statistically-relevant turbulent diagnostics. A heat source is then mandatory to replace the heat transported radially to the outer edge of the domain. Two main different approaches are used. In so-called temperature gradient-driven simulations, the background temperature profile gradients are prescribed in input and a special heat source injects the required amount of heat necessary to maintain the total temperature profile as close as possible to the initial background profile. This approach is very interesting in terms of computational resources because the system remains close to its initial state and sufficient statistics can be obtained over turbulence timescale. However, this method does not reflect the experimental conditions where heat fluxes are typically imposed by various heat sources such as neutral beam injection, electron and ion cyclotron resonance heating, or ohmic heating. To address this issue, the so-called flux-driven approach can be used. It consists of prescribing heat fluxes as input, and letting the temperature profiles evolve consistently.

Currently, several gyrokinetic codes are available; even though very similar in appearance, they are all particular in the approximations and/or the numerical schemes they use. They can be sorted into three groups depending on the numerical scheme they use for solving the gyrokinetic equations. The Lagrangian approach, often called particle-in-cell (PIC) approach, consists in separating particle and field data. The physical particles are represented by numerical particles, also called markers, that sample a portion of the phase space. On the other hand, field data is represented on a discretized grid of configuration space. The main steps of the PIC algorithm are: a deposition method during which the particle charge and current are computed. This involves an interpolation of the quasi-random particle positions to the discrete field grids. Then, the self-consistent electromagnetic fields are computed in the field

solver using the previously computed charge and current. Finally, the particle positions in phase space are evolved according to the gyrokinetic equations and the self-consistent fields. Note that this step also involves an interpolation of field data to particle data. The second numerical method is the Eulerian approach, which consists in discretizing the phase space using fixed grids. Finally, the third approach is the so-called semi-Lagrangian method which uses fixed grid for discretizing the phase space and the values of the distribution function at the next timestep are computed by following backward the characteristics. As a non exhaustive list, we can cite the Lagrangian codes ORB5 [10], GT3D [11], XGC [12], GTC [13], GTS [14, 15], ELMFIRE [16], and PG3EQ [17]; the Eulerian codes GENE [18], GT5D [19], GYRO [20], GS2 [21], GKV [22], and GKW [23]; and the semi-Lagrangian codes GYSELA [24] and NLT [25].

## 1.1 Basic concepts of plasma and tokamak physics

### 1.1.1 Particle and fluid drifts

Plasma is, as a first approximation, a gas composed of positively and negatively charged particles. As such, it is subject to electric and magnetic fields but is also the source of electromagnetic perturbations. This self-consistent problem can become very complex, especially when the magnetic fields possess a complex topology, which is typically the case in magnetic confinement devices. To begin with, let us look at the different forces and drifts that the moving plasma particles may be subject to. Two kind of drifts can be identified: particle and fluid drifts. Particle drifts are caused by a force acting on a single plasma particle while fluid drifts are the consequence of macroscopic effects.

The most basic case is the one of a single charge in an imposed constant and uniform magnetic field. This very simple example defines the Larmor motion of charged particles in a magnetic field. Even though very basic, this case—comparable to the harmonic oscillator—is at the origin of many plasma properties. Let us consider a particle of mass  $m$ , charge  $q$ , and velocity  $\mathbf{v}$  immersed in a constant and uniform magnetic field  $\mathbf{B} = B\hat{e}_z$ , where  $\hat{e}_x$ ,  $\hat{e}_y$ , and  $\hat{e}_z$  are the unit vectors defining an arbitrary Cartesian coordinate system. The Lorentz force  $\mathbf{F}_L$  acting on the charge is given by

$$\mathbf{F}_L = q\mathbf{v} \times \mathbf{B}, \quad (1.3)$$

Due to this force, the charged particle has a characteristic Larmor motion defined by

$$x - x_0 = \rho_L \sin(\Omega_c t), \quad (1.4)$$

$$y - y_0 = \rho_L \cos(\Omega_c t), \quad (1.5)$$

$$z - z_0 = v_{\parallel} t, \quad (1.6)$$

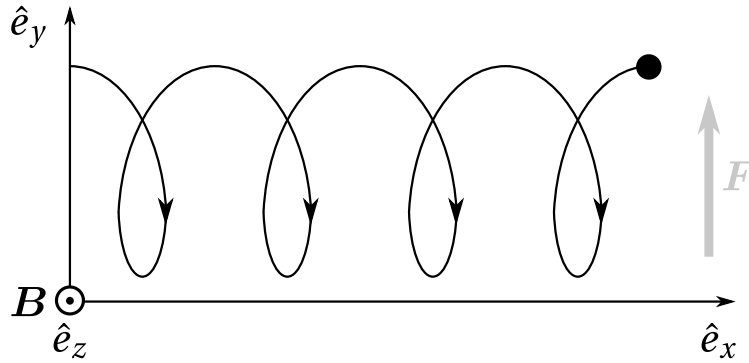
where  $\Omega_c = qB/m$  is the cyclotron frequency,  $\rho_L = v_{\perp}/\Omega_c$  is the Larmor radius, and  $v_{\parallel}$  and  $v_{\perp}$  are respectively the parallel and perpendicular components of the velocity with respect to the

magnetic field. The coordinates  $(x_0, y_0, z_0)$  define the guiding center of the particle.

In general, the Lorentz force is not the only force acting on the plasma particles. If another force is present, the motion of the particle is slightly changed to include a drift of the guiding center perpendicular to the magnetic field and the acting force  $\mathbf{F}$ . This drift has a velocity  $\mathbf{v}_d$  defined by

$$\mathbf{v}_d = \frac{1}{q} \frac{\mathbf{F} \times \mathbf{B}}{B^2}. \quad (1.7)$$

Note that here,  $\mathbf{F}$  is assumed to not vary too much on the scale of  $\rho_L$ , this is the so-called drift approximation. The particle motion can then be decomposed into three components: the very fast Larmor gyration, the motion parallel to the magnetic field, and the slower guiding center drift. An illustration of such a drift is made in Figure 1.2. In this setup, the force  $\mathbf{F}$  is along the  $y$  direction, perpendicular to the magnetic field. Because of  $\mathbf{F}$ , the particle is accelerated in the  $y$  direction which causes the Larmor radius to increase when the particle is moving in the same direction as the force and to decrease when the particle is moving in the direction opposite to the force. This combined motion generates a drift perpendicular to the force and the magnetic field. In principle, the force  $\mathbf{F}$  could be of any nature. However, there are a few relevant cases



**Figure 1.2 :** Illustration of the particle drift generated by the presence of a force  $\mathbf{F}$ .

to the physics of magnetically confined plasmas, e.g. in the presence of an electric field or in an nonuniform magnetic field.

In the presence of an electric field  $\mathbf{E}$ , supposed along the  $y$  direction, the force is simply  $\mathbf{F} = q\mathbf{E}$ . This gives rise to the so-called  $\mathbf{E} \times \mathbf{B}$  drift with velocity

$$\mathbf{v}_{E \times B} = \frac{\mathbf{E} \times \mathbf{B}}{B^2}. \quad (1.8)$$

An interesting point to note is that  $\mathbf{v}_{E \times B}$  is independent of the particle mass and charge; ions and electrons drift at the same velocity.

The second drift occurs in a curved magnetic field. It arises from the centrifugal force due to the curvature of the magnetic field lines combined with a nonuniform magnetic field amplitude. The drift velocity is given by

$$\mathbf{v}_C = \frac{m}{q} \frac{\mathbf{R}_c \times \mathbf{B}}{R_c^2 B^2} \left( v_{\parallel}^2 + \frac{1}{2} v_{\perp}^2 \right), \quad (1.9)$$

where  $\mathbf{R}_c$  is the radius of curvature of the magnetic field lines.

Finally, the fluid drifts are not caused by a force acting on a single particle but rather by the interplay between spatial inhomogeneities of the plasma density and temperature, and the finite Larmor radius in a constant magnetic field. At the microscopic level, the guiding centers do not move, but at the macroscopic level, drifts appear due to the interplay of many particles. The diamagnetic drift is a fluid drift due to the spatial variation of the plasma pressure  $p$ . The related drift velocity is given by

$$\mathbf{v}_{\text{dia}} = -\frac{\nabla p \times \mathbf{B}}{qnB^2}. \quad (1.10)$$

Note that the diamagnetic drift velocity is essentially the drift defined by Eq. (1.7) with a force  $-\nabla p/n$ .

### 1.1.2 Trapped particles

In general, the spatial inhomogeneity of the magnetic field causes a force parallel to the magnetic field lines that affects the parallel motion of the particles:

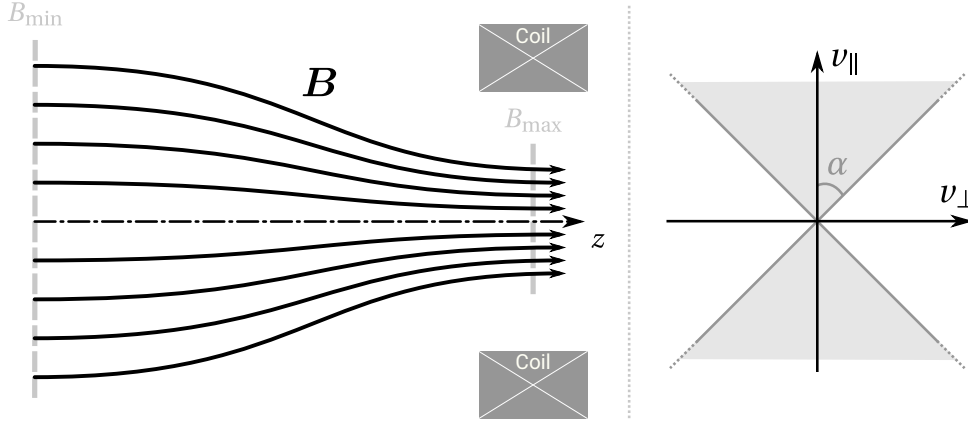
$$F_{\parallel} = -\mu \nabla_{\parallel} B, \quad (1.11)$$

where  $\mu = mv_{\perp}^2/2B$  is the magnetic moment, which is an adiabatic invariant, and  $\nabla_{\parallel}$  is the gradient along the magnetic field lines.

The parallel force of Eq. (1.11) is at the origin of the magnetic mirror. As a plasma particle travels along the magnetic field lines from a low- $B$  region to a high- $B$  region, it sees an increasing magnetic field amplitude. Since the magnetic moment  $\mu$  is constant, the perpendicular velocity  $v_{\perp}$  must increase to compensate the variation of  $B$ . This, in turns, leads to a decrease of the parallel velocity  $v_{\parallel}$  because of the energy conservation. This can be directly seen by writing  $v_{\parallel}$  as

$$v_{\parallel} = \sqrt{2(\mathcal{E} - \mu B)}, \quad (1.12)$$

where  $\mathcal{E}$  is the energy of the particle. If, at some point, the magnetic field is high enough,  $v_{\parallel}$  vanishes and the particle is reflected back. This type of confinement is typically used in linear mirror machines where a set of coils creates a nonuniform magnetic field, Figure 1.3 left. A particle is said to be trapped if its magnetic moment is large enough so that the change in



**Figure 1.3** : Illustration of the magnetic mirror. (left) Due to a non-ideal magnetic coil setup, the magnetic field is not axisymmetric in the  $z$  direction. This causes a force parallel to the magnetic field lines that decreases the parallel velocity of the particle. Under certain conditions, particles can be trapped in this magnetic well. (right) Cone of loss illustrating the part of velocity space where particles are trapped (shaded area) and passing.

magnetic field makes its parallel velocity vanish, i.e.

$$\mu > \mu_{\max} = \frac{\mathcal{E}}{B_{\max}}, \quad (1.13)$$

where  $B_{\max}$  is the maximum amplitude of the magnetic field. Another way to write Eq. (1.13) is

$$\tan \alpha = \frac{v_{\parallel}}{v_{\perp}} < \sqrt{\frac{B_{\max}}{B} - 1}, \quad (1.14)$$

where  $B$  is the magnetic field amplitude at the position of the particle and  $\alpha$  is the angle defined in Figure 1.3 right, which shows an illustration of the so-called loss cone. The figure depicts the velocity space  $(v_{\perp}, v_{\parallel})$  where two regions have been delimited according Eq. (1.14) and which represent the portion of velocity space where particles are trapped (shaded area) and where they are passing. Trapped (passing) particles may become passing (trapped) by two main mechanisms: through a change of their pitch angle, typically caused by collisions or the parallel non-linearity, and through a radial drift which changes the local trapped/passing limit.

### 1.1.3 Tokamak magnetic equilibrium

In tokamaks, the plasma confinement is done using a twisted magnetic field  $\mathbf{B}$  generated by external coils and induced toroidal current. To characterize the static magnetic equilibrium,

## Chapter 1. Introduction

---

the ideal MHD equations in the stationary limit with  $\mathbf{v} = 0$  are given by

$$\mathbf{j} \times \mathbf{B} = \nabla p, \quad (1.15)$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j}, \quad (1.16)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (1.17)$$

where  $\mathbf{j}$  is the current density and  $p$  is the plasma pressure. The static magnetic equilibrium is reached when the plasma pressure force is balanced with the magnetic forces, Eq. (1.15). Even though stationary flows are observed in many fusion experiments, the  $\mathbf{v} = 0$  limit is chosen in this introductory section for the sake of simplicity. A more developed physical model integrating strong flows will be briefly presented in Chapter 2.

By forming the dot product of  $\mathbf{B}$  and  $\mathbf{j}$  with Eq. (1.15), it follows that

$$\mathbf{B} \cdot \nabla p = \mathbf{j} \cdot \nabla p = 0. \quad (1.18)$$

In other words, magnetic field lines and currents lie on surfaces of constant pressure, also called flux surfaces, Figure 1.4. The magnetic equilibrium is thus composed of a set of nested flux surfaces ranging from the magnetic axis, where the surface is a unique toroidal magnetic field line, to the last closed flux surface (LCFS) delimiting the core and scrape-off layer (SOL). Labelling each flux surface with its pressure is not convenient. Usually, one uses the poloidal flux label defined as

$$\psi = \int_S \mathbf{B} \cdot d\mathbf{S}, \quad (1.19)$$

where the integration is made over a surface extending from the magnetic axis to a line on an isobaric surface and  $d\mathbf{S}$  is an infinitesimal surface element normal to the surface and pointing outside of it. In this work, the poloidal flux  $\psi$  is always normalized to its edge value, i.e.  $\psi \equiv \psi/\psi_e$ .

Tokamak magnetic fields need to be twisted, i.e. to have finite rotational transforms in order to counteract the drifts related to the curvature of the magnetic field that would otherwise eject all the plasma to the outer-side walls through a self-generated electric field. Furthermore, at first order approximation, background quantities in tokamaks can be assumed to be axisymmetric<sup>3</sup>. The most general form of divergence-free magnetic field satisfying the aforementioned conditions can be written as

$$\mathbf{B} = F(\psi)\nabla\varphi + \nabla\varphi \times \nabla\psi, \quad (1.20)$$

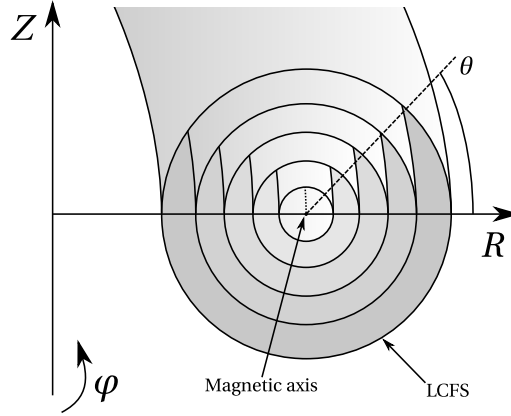
where  $F(\psi)$  is the poloidal current flux<sup>4</sup> and  $(R, Z, \varphi)$  is the cylindrical coordinate system

---

<sup>3</sup>This is of course not the case in experimental conditions because of the non-ideal toroidal coil setup for example

<sup>4</sup>According to Ampère's law:  $\nabla \times \mathbf{B} = \nabla F(\psi) \times \nabla\varphi + \nabla \times (\nabla\varphi \times \nabla\psi)$ . The  $\nabla F(\psi) \times \nabla\varphi$  term thus represent the





**Figure 1.4 :** Illustration of the tokamak nested flux surfaces.

presented in Figure 1.4. The magnetic field defined by Eq. (1.20) is clearly a combination of a purely toroidal magnetic field  $\mathbf{B}_T = F(\psi)\nabla\varphi$  and a purely poloidal magnetic field  $\mathbf{B}_P = \nabla\varphi \times \nabla\psi$ .

Combining Eqs. (1.15) and (1.16), and using the general definition of the magnetic field, Eq. (1.20), leads to the so-called Grad-Shafranov equation:

$$R^2 \nabla \cdot \left( \frac{1}{R^2} \nabla \psi \right) = -F(\psi)F'(\psi) - \mu_0 R^2 p'(\psi). \quad (1.21)$$

Providing a pressure profile  $p(\psi)$  and poloidal current function  $F(\psi)$ , the Grad-Shafranov equation allows one to obtain the corresponding ideal MHD magnetic equilibrium. Codes such as CHEASE[26] solve the Grad-Shafranov equation to find magnetic equilibria.

One important parameter to characterize the magnetic surfaces is the safety factor  $q$  defined by

$$q(\psi) = \frac{1}{2\pi} \oint_0^{2\pi} \frac{\mathbf{B} \cdot \nabla \varphi}{\mathbf{B} \cdot \nabla \theta} d\theta, \quad (1.22)$$

where  $\theta$  is the geometrical poloidal angle, Figure 1.4. The safety factor is a flux surface quantity and its inverse, the rotational transform  $\iota = 1/q$ , is a measure of how twisted the magnetic field lines are.

Plasmas generated in current tokamaks exhibit various sizes and shapes. An analytical axisymmetric toroidal equilibrium expansion [27] can be used to derive the  $(R, Z)$  contours of the flux surfaces. Furthermore, using this expansion, one can define a set of analytical shaping parameters such as elongation and triangularity. The decomposition of the Grad-Shafranov equation, Eq. (1.21), is made using a discrete Fourier transform in the poloidal direction and

---

poloidal current flux.

an expansion in the inverse local aspect ratio  $\varepsilon = r/R_0$  of the cylindrical coordinates  $(R, Z)$ . Here  $r$  is the geometrical radius and  $R_0$  is the tokamak major radius localizing the magnetic axis. The Fourier expansion of  $(R, Z)$  is written:

$$R = R_0 + r \cos(\theta) - \Delta(r) + \sum_{m=2} S_m(r) \cos[(m-1)\theta] + \mathcal{O}(\varepsilon^3), \quad (1.23)$$

$$Z = r \sin(\theta) - \sum_{m=2} S_m(r) \sin[(m-1)\theta] + \mathcal{O}(\varepsilon^3), \quad (1.24)$$

where  $\Delta$  is the Shafranov shift and  $S_m$  are related to the shaping coefficients. For example, the elongation  $\kappa$  and triangularity  $\delta$  are defined by  $S_2$  and  $S_3$  respectively:

$$\kappa = \frac{r - S_2}{r + S_2}, \quad (1.25)$$

$$\delta = \frac{4S_3}{r}. \quad (1.26)$$

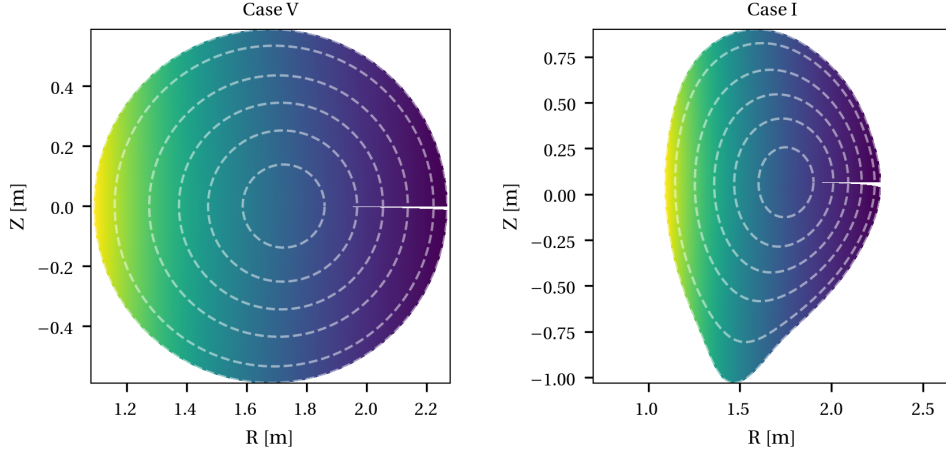
Using the decomposition defined by Eqs. (1.23)–(1.24), the magnetic field amplitude can be written as

$$B = B_0 \left\{ 1 - \varepsilon \cos(\theta) + \varepsilon^2 \left[ \cos^2(\theta) + \frac{1}{2q^2} \right] + \frac{\Delta}{R_0} - \sum_{m=2} \frac{S_m(r)}{R_0} \cos[(m-1)\theta] \right\}. \quad (1.27)$$

The Shafranov shift  $\Delta$  and the shaping coefficients can then be found by solving the Grad-Shafranov equation, Eq. (1.21), using the decomposition, Eqs. (1.23)–(1.24) and the magnetic field expansion, Eq. (1.27). Two magnetic configurations are shown as examples in Figure 1.5. Apart from the MHD stability point of view, different experiments have shown that plasma shape also influences transport properties. For example, different experiments in TCV and DIII-D [30, 31, 32] showed that plasmas with a negative triangularity exhibit a significantly reduced level of core electron heat transport compared to similar shots with positive triangularity. This was also confirmed, at least qualitatively, by gyrokinetic simulations [33].

#### 1.1.4 Microinstabilities

Tokamak plasmas exhibit a vast number of microinstabilities spanning over multiple spatial and temporal scales. These modes provide a mechanism for the generation of plasma turbulence, which in turns leads to an important fraction of the particle and heat transport. Although fluid models allow one to derive basic properties for these microinstabilities, important effects such as dissipation due to the Landau damping are not described. In general, a rigorous description of these modes generally requires to solve the nonlinear Vlasov equation. However, a linear analysis of the mode characteristics is already insightful and allows one to identify possible turbulence drive mechanisms. A general description, even linear, including the complex magnetic geometry of a tokamak requires numerical codes to solve the complex set of equations of the physical model. Various approximations are often used to simplify



**Figure 1.5** : Illustration of two different magnetic geometries taken from [28]. The amplitude of the magnetic field is shown as well as some flux surfaces (white dashed lines). The left figure represents the so-called “Case V” which is a circular equilibrium and the right figure shows the so-called “Case I” with enhanced shaping and which is very close to a real DIII-D discharge [29].

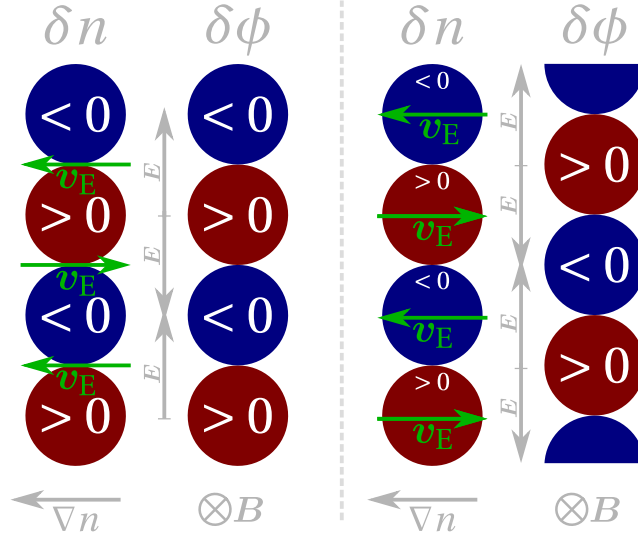
the model, e.g. adiabatic electrons or cold ion limit, thus allowing one to derive analytical predictions such as dispersion relations.

Various types of microinstabilities exist with different drives. For example, one distinguishes between electrostatic and electromagnetic instabilities depending on the dominant fluctuating field. They can be destabilized by density and temperature gradients, magnetic field curvature, or even interaction with fast particles. In this work, we mainly focus on electrostatic-type instabilities: the toroidal ion (electron) temperature gradient (ITG/ETG) instability and the trapped electron modes (TEM). An emphasis is also made on the zonal flows, which act as a saturation mechanism for some type of turbulence, and their oscillatory counterpart, the so-called geodesic acoustic modes (GAM).

### 1.1.4.1 Drift waves

Drift waves are electrostatic modes that develop in an inhomogeneous plasma. They are driven by a density gradient and propagate essentially perpendicular to the magnetic field at a phase velocity of the order of the diamagnetic drift velocity, Eq. (1.10). Drift waves usually lead to fluctuations in density, temperature and electrostatic potential and are identified as the main mechanism for anomalous transport, along with the ITG/ETG/TEM discussed in the following sections. The drift wave is essentially a deformed sound wave, i.e. in the absence of density gradient, the drift wave dispersion relation reduces to the ion sound wave dispersion relation. Using a kinetic approach, one can show that drift waves are linearly stable in the limit

of adiabatic electrons and destabilized by resonant interactions between particles and the wave.



**Figure 1.6 :** Illustration of the basic mechanisms leading to the drift wave propagation. (left) In the limit of adiabatic electrons, both density and electrostatic potential perturbations are in phase thus leading to a linearly stable waves. (right) On the other hand, if there is a finite phase between the density and electrostatic potential perturbations, the drift wave becomes linearly unstable.

The basic mechanisms leading to the development of a drift wave are illustrated in Figure 1.6. The setup is the following. A plasma with a background density profile having a gradient  $\nabla n$  is immersed into a fixed magnetic field  $\mathbf{B}$  such that  $\mathbf{B} \cdot \nabla n = 0$ . At some arbitrary position, there is a small density perturbation  $\delta n$  consisting of regions where the perturbation is positive (red dots) and regions where the perturbation is negative (blue dots). Similarly, perturbations of the electrostatic potential  $\delta\phi$  are created self-consistently to the density perturbations. Note that, in the sketch, they are placed next to each other for the sake of visibility. In reality, both perturbations are superimposed. First, let us look at the case where electrons respond adiabatically, i.e. there is no phase between the density and electrostatic potential perturbations, left figure. Because of the density perturbation, a fluctuating electrostatic potential is generated self-consistently to which an electric field  $E$  is associated. This electric field in turns makes the plasma drift due to the  $E \times B$  drift. Since there is a density gradient, regions of lesser density plasma are advected to the positive perturbation, i.e. when both density gradient and  $E \times B$  velocity are in the same direction, and regions of higher density plasma are advected to the negative perturbation, i.e. when  $\nabla n$  and  $v_E$  are in the opposite direction. This causes the perturbation to drift essentially perpendicularly to the density gradient and the magnetic field. Note that since both perturbations are in phase, the amplitude of the drift wave does

not change in time: it is linearly stable. On the other hand, if there is a finite phase between  $\delta n$  and  $\delta\phi$  (right figure), the drift waves becomes now linearly unstable. Indeed, when the perturbations are in phase, a minimum of  $\delta n$  corresponds to a null  $E \times B$  drift. This causes the drift to “fill” (“dig”) the density perturbation holes (bumps). On the other hand, when there is a phase difference between  $\delta n$  and  $\delta\phi$ , the  $E \times B$  drift is non null at the position of minimum/maximum density perturbation thus amplifying it.

### 1.1.4.2 Ion temperature gradient modes

Ion temperature gradient modes have been extensively studied over the last decades because they are identified as being mainly responsible for the anomalous ion heat transport in tokamaks. The ITGs are a drift-wave type instability driven by a temperature gradient, with a perpendicular phase velocity related to the ion diamagnetic drift motion, and are characterized by relatively long perpendicular wavelengths,  $k_\theta \rho_{si} \sim 0.1-1$ , where  $\rho_{si}$  is the ion acoustic gyroradius.

An insightful description of the ITG can be derived using the approximation of a slab plasma, i.e. without any magnetic field curvature; this is the so-called slab-ITG. It basically consists of an ion acoustic wave destabilized by a temperature gradient. If one bends the magnetic field, forces due to the curvature and  $\nabla B$  act on the plasma and the ITG develop an interchange like behaviour. This is the so-called toroidal-ITG. To derive its dispersion relation, we follow the procedure presented in [34]. The system is perturbed by a small electrostatic fluctuation  $\delta\phi$  of the form

$$\delta\phi = \widehat{\delta\phi} \exp [i(k_\theta \theta + k_\parallel \varphi - \omega t)], \quad (1.28)$$

where  $\mathbf{k} = (0, k_\theta, k_\parallel)$  is the wave vector and  $\omega$  is the mode complex frequency. Assuming adiabatic electrons and quasineutrality, we solve the dielectric function  $\epsilon(\mathbf{k}, \omega) = 0$  with

$$\epsilon(\mathbf{k}, \omega) = \frac{1}{(k\lambda_{De})^2} + \frac{1}{(k\lambda_{De})^2} \left[ 1 + (\omega - \omega'_{Ti}) \int d\mathbf{v} \frac{f_{0i}}{n_{0i}} \frac{J_0^2\left(\frac{k_\theta v_\perp}{\Omega_{ci}}\right)}{k_\parallel v_\parallel + \omega_{di} - \omega} \right]. \quad (1.29)$$

Here,  $\lambda_{D\sigma}$  is the Debye length of species  $\sigma$ ,  $f_{0i}$  and  $n_{0i}$  are respectively the ion background distribution function and density,  $J_0$  is a Bessel function of the first kind describing finite Larmor effects,  $\omega'_{Ti} = (k_\theta T_i \nabla T_i / qB) \partial / \partial T_i$  is an operator defining the temperature drift frequency, and  $\omega_{di} = \mathbf{k} \cdot (\mathbf{v}_{\nabla B} + \mathbf{v}_C)$  is the drift frequency due to the  $\nabla B$  and curvature drifts.

To solve  $\epsilon(\mathbf{k}, \omega) = 0$ , we use the so-called fluid-like limit, i.e.  $|\omega / (k_\parallel v_{th,i})| \ll 1$ , where  $v_{th,i}$  is the ion thermal speed. Furthermore, finite Larmor effects are neglected, i.e.  $J_0 \sim 1$  and the mode frequency is supposed to be small as compared to the diamagnetic drift frequency, i.e.  $|\omega \ll \omega_{di}|$ . With the aforementioned approximations,  $\epsilon(\mathbf{k}, \omega) = 0$  reduces to

$$1 - \left(1 - \frac{\omega_T}{\omega}\right) \left[ \frac{k_\parallel c_s}{\omega} + \frac{T_e}{T_i} \frac{\langle \omega_{di} \rangle}{\omega} \right] = 0, \quad (1.30)$$

where we have carried out the temperature derivative in  $\omega'_{Ti}$  and  $\langle \omega_{di} \rangle = 2 T_i k_\theta |\nabla_\perp \ln B| / (eB)$ . Note that, in this derivation, the density gradient has been neglected to emphasize that the ITG drive is the temperature gradient.

It is interesting to note that the ITG instability has a finite growth rate even in the limit  $k_\parallel \rightarrow 0$ . Indeed, further assuming that  $|\omega| \ll |\omega_T|$  leads to

$$1 + \frac{T_e}{T_i} \frac{\omega_T \langle \omega_{di} \rangle}{\omega^2} = 0 \quad (1.31)$$

which admits the following two solutions

$$\omega = \pm \left[ -2 \frac{T_e}{T_i} \left( \frac{k_\theta}{eB} \right)^2 \nabla \ln T_i \cdot \nabla \ln B \right]^{1/2}. \quad (1.32)$$

In this case, ITG modes are destabilized if the gradients of the temperature and magnetic fields are in the same direction, i.e. in the so-called unfavorable curvature region, and are stable in the inverse case, i.e. in the so-called favorable curvature region. They are then ballooned in the low field side of the tokamak.

Note that at sufficiently small perpendicular wavelength, the effect of perturbations on ions is averaged out due to the finite Larmor radius effects. One can thus assume that ions have an adiabatic-like behaviour. In this case, the role of ions and electron is exchanged and the corresponding mode is the so-called ETG, which was predicted to drive significant electron heat transport [35].

### 1.1.4.3 Trapped electron modes

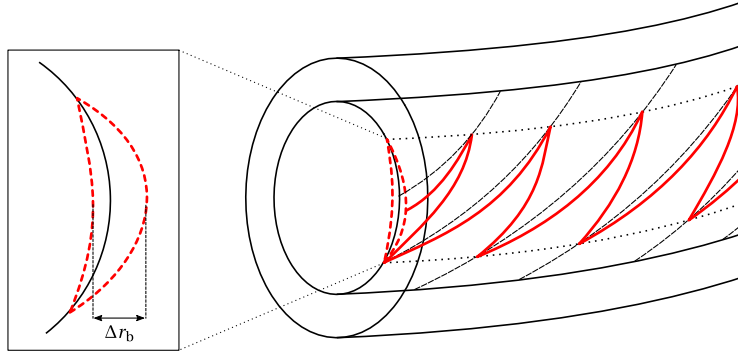
Another important mode of interest for understanding tokamak transport is the trapped electron mode (TEM). First identified in [36], it is now recognized as the main player for the anomalous electron heat transport and has been extensively investigated using gyrokinetic simulations [37, 38, 39, 40, 41].

Trapped electrons bounce back and forth between regions of high and low magnetic field, Figure 1.7. Due to the magnetic field curvature, they experience drifts that slightly displace them radially. If one projects their trajectory on a poloidal plane, this periodic motion takes the particular shape of a banana with a width and a bounce frequency that can respectively be estimated by

$$\Delta r_b \sim \frac{q}{\sqrt{\epsilon}} \rho_L, \quad (1.33)$$

$$\omega_b \sim \sqrt{\epsilon} \frac{v_{th}}{qR_0}. \quad (1.34)$$

Because of their particular shape, trapped particle trajectories are called banana orbits. Furthermore, trapped particles also experience a finite toroidal precession with velocity  $\langle \dot{\phi} \rangle_b$  and



**Figure 1.7 :** Cartoon illustrating the trapped particle trajectory. Trapped particles are reflected back and forth at the turning points, where their parallel velocity becomes zero. Due to the magnetic field curvature they have a finite toroidal precession (red plain line). Projected on a poloidal plane, the trapped particle trajectory takes the particular shape of a banana, hence the name of banana orbits.

can resonate with an electrostatic perturbation thus developing an instability; the TEM in the case of trapped electrons (trapped ions have their counterpart called trapped ion mode).

In the following, we give a short summary of the procedure leading to a dispersion relation for the TEM. The complete derivation, from which this summary is based on, can be found in [42]. Due to their small mass, electrons have, in general, a Larmor radius much smaller than the ions<sup>5</sup>. This justifies the use of the drift-kinetic limit where one assumes that electrons have no Larmor radius, i.e. particles coincide with their guiding centers. Furthermore, the frequency  $\omega$  of the modes of interest is typically much smaller than the bounce frequency of trapped electrons, Eq. (1.34). In this limit, the drift-kinetic Vlasov equation can be further reduced by taking its bounce average:

$$\left( \frac{\partial}{\partial t} + \langle \dot{\phi} \rangle_b \frac{\partial}{\partial \phi} \right) \tilde{G}_e = - \left( \frac{e f_{0e}}{T_e} \frac{\partial}{\partial t} - \frac{1}{B_p} \frac{\partial f_{0e}}{\partial x} \frac{\partial}{\partial \phi} \right) \langle \phi \rangle_b, \quad (1.35)$$

where  $\langle \dot{\phi} \rangle_b$  is the toroidal precession velocity and  $B_p$  is the poloidal component of the magnetic field and  $x$  is a flux surface label. The electron background distribution function is supposed to be a Maxwellian of temperature  $T_e$  and  $\tilde{G}_e$  is the non-adiabatic fluctuating part of the total distribution function:

$$\tilde{G}_e = f_e - \frac{e\phi}{T_e} f_{0e}. \quad (1.36)$$

<sup>5</sup>Considering electrons with the same temperature as the ions

Finally, the operator  $\langle A \rangle_b$  is the bounce average of the quantity  $A$  defined by

$$\langle A \rangle_b = \frac{\omega_b}{2\pi} \int_{\text{orbit}} A \frac{dl}{v_{\parallel}}. \quad (1.37)$$

Considering perturbations of the form Eq. (1.28), Eq. (1.35) can be solved for the non-adiabatic part of the distribution function:

$$\tilde{G}_e = -\frac{e f_{0e}}{T_e} \frac{\omega - \omega_e^*}{\omega - n \langle \dot{\phi} \rangle_b} \langle \phi \rangle_b, \quad (1.38)$$

where the drift frequency of the species  $\sigma$  is given by

$$\omega_{\sigma}^* = \omega_{n\sigma} + \omega'_{T\sigma} = \frac{k_{\theta} T_{\sigma}}{q_{\sigma} B} \left( \frac{d \ln n_{\sigma}}{dx} + \nabla T_{\sigma} \frac{\partial}{\partial T_{\sigma}} \right). \quad (1.39)$$

Finally, using the quasineutrality equation, the dispersion relation is found, assuming  $\langle \phi \rangle_b \approx \phi$ :

$$\frac{T_i}{T_e} + \frac{T_i}{T_e} \frac{2\alpha_t}{\omega_{\phi e}} \left\{ \left[ \omega - \omega_{ne} \left( 1 - \frac{3}{2} \eta_e \right) \right] W(z_{be}) - \omega_{ne} \eta_e \left( \frac{\omega}{\omega_{\phi e}} W(z_{be}) + \frac{1}{2} \right) \right\} + \quad (1.40)$$

$$+ 1 - \int d\nu \frac{f_{0i}}{n_{0i}} \frac{\omega - \omega_i^*}{\omega - k_{\parallel} v_{\parallel} - \omega_{di}} = 0, \quad (1.41)$$

where  $\alpha_t$  is the fraction of trapped electrons,  $\eta_{\sigma} = d \ln N_{\sigma} / d \ln T_{\sigma}$ ,  $\omega_{\phi} = n \langle \dot{\phi} \rangle_b$ , the plasma dispersion function  $W(z) = (2\pi)^{-1/2} \int_{-\infty}^{\infty} x / (x - z) \exp(-x^2/2) dx$ , and  $z_b = \text{sign}(\omega_{\phi}) \sqrt{2\omega/\omega_{\phi}}$ . The dispersion relation defined by Eqs. (1.40)–(1.41) is, in fact, not restricted to the pure TEM case. One can recover the ITG dispersion relation by removing the TEM drive, i.e. putting  $\alpha_t = 0$ . Note that contrarily to the ITG dispersion relation presented in Eq. (1.29) which only considered a temperature gradient, here one also retains the effect of a density gradient. On the other hand, if the ITG drive is neglected, one finds the dispersion relation for a pure TEM:

$$\frac{T_i}{T_e} + \frac{T_i}{T_e} \frac{2\alpha_t}{\omega_{\phi e}} \left\{ \left[ \omega - \omega_{ne} \left( 1 - \frac{3}{2} \eta_e \right) \right] W(z_{be}) - \omega_{ne} \eta_e \left( \frac{\omega}{\omega_{\phi e}} W(z_{be}) + \frac{1}{2} \right) \right\} + \frac{\omega_{ni}}{\omega} = 0, \quad (1.42)$$

Finally, one can estimate a typical frequency and growth rate of the TEM in the limit of isothermal species with  $|\omega_{\phi e}| \ll \omega$ :

$$\omega \simeq \omega_{ne}, \quad (1.43)$$

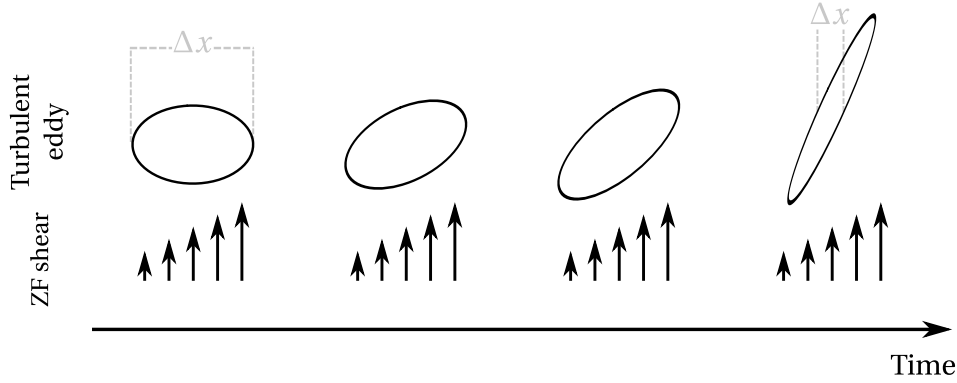
$$\gamma \simeq \sqrt{\frac{3}{2} \alpha_t \omega_{\phi e} \omega_{ne} (1 + \eta_e)}. \quad (1.44)$$

### 1.1.5 Physics of zonal flows and geodesic acoustic modes

Zonal flows (ZFs) are axisymmetric band-like shear flows that are present in many situations going from the Jovian atmosphere to tokamak plasmas. In the latter case, which is the one



of interest in this work, ZFs are  $n = m = 0$  electrostatic fluctuations, where  $n$  and  $m$  are respectively the toroidal and poloidal Fourier mode number, with a finite radial wavenumber and can be viewed as a superposition of a geodesic acoustic mode (GAM) with frequency  $\omega_{\text{GAM}}$  and a Landau damping rate  $\gamma_{\text{GAM}}$ , and a zero-frequency residual. As such, ZFs cannot tap free energy stored in, e.g. density and temperature gradients and are thus driven by nonlinear interactions—even though they are linearly stable. ZFs are generated through nonlinear coupling with drift-wave-type instabilities and also act as a saturation mechanism of the turbulence generated by the primary instability. An example of this saturation mechanism was shown in [29]: ZFs quench the growing ITG instability thus increasing the critical temperature gradient at which ITG modes can grow; this is the so-called Dimits shift. A pictorial explanation of the saturation mechanism by ZFs is shown in Figure 1.8. The turbulent eddies are sheared by the ZFs until they eventually break up and couple to small scale dissipation. Zonal flows being modes with  $n = 0$  and  $k_{\parallel} = 0$ , they are not subject to Landau damping. This implies that the only dissipation is made asymptotically through collisions.



**Figure 1.8 :** Cartoon illustrating the turbulence saturation mechanism through ZF shearing of the turbulent eddies. Because of the ZF shearing the typical turbulent eddy radial width is reduced thus decreasing the diffusive-type transport coefficient  $D \sim \Delta x^2 / \Delta t$ .

The finite frequency ZFs are the geodesic acoustic modes (GAMs). They are perturbations of the  $n = m = 0$  electrostatic potential coupled with the  $n = 0$ ,  $m = 1$  sidebands of the density perturbation because of the magnetic field toroidicity. Various analytical works have aimed at computing the GAM frequency and growth rate as well as the residual level, e.g. [43, 44, 45, 46]. Among them, the calculations of Rosenbluth and Hinton (RH) in [43] have given birth to the eponymous test, which is nowadays commonly used to test the ZF properties of gyrokinetic codes and cross-code benchmarks. The typical RH test setup is to initialize an axisymmetric perturbation with a finite radial wave number and letting the system evolve free of any drive. After a first phase of GAM oscillation of the electrostatic potential, the system converges to a

(non) vanishing ZF residual for collisional (collisionless) systems.

### 1.2 Contribution and outline of the thesis

The main goal of this work is to enable the ORB5 code to perform flux-driven simulations and study their application to comparison with experiments carried out in the TCV tokamak [47]. To this end, three major lines of work have been defined.

Flux-driven simulations are very demanding in terms of computational resources because the runs have to be long enough for the plasma profiles to relax toward a quasi-steady equilibrium consistent with the imposed heat fluxes. This relaxation process takes place over transport time scales that are much longer than the turbulence time scales. Indeed, it is proportional to the squared of the normalized machine size, which makes flux-driven simulations of a large machine such as ITER [48] impractical. A solution to mitigate this problem is to ensure that the code makes an optimal use of the numerical resources. To this end, the first main objective of this thesis consists of the refactoring and optimization of the ORB5 code. After almost 20 years of continuous development, the code became impractical to work with. Furthermore, the optimization process requires the logic of the code to be as simple as possible, which further motivated the need of a refactoring. To carry out the optimization process, which essentially consists of implementing multi-threading using OpenMP, we first implemented a test bed retaining only the key elements of the PIC algorithm that allowed us to easily test various algorithm optimization and parallelization strategies, the best of which were then implemented into the full production ORB5 code.

One of the extensions of this work is to study numerically the problem of profile stiffness observed in the TCV tokamak [49]. The prerequisites to carry out such TCV-relevant simulations is to work with kinetic electrons since TCV shots often exhibit regimes dominated by electron modes. However, resolving the physics at the electrons space and time scales is constrained by the fast electron velocities and small scales. An alternative to the kinetic electrons is to use a hybrid model in which passing electrons have an adiabatic response while trapped electrons are kinetic. This model is able to simulate the trapped electron modes which are dominant in the aforementioned TCV experiments. However, this model is wrong in the nonlinear regime as it does not ensure the ambipolar condition and creates spurious sources of particle and momentum. The second main objective of this thesis is thus to develop a corrected hybrid electron model which addresses the problems of the current model.

The final objective of this work is to enable the flux-driven capabilities of ORB5. To this end, a heat source prescribing a fixed heat flux while conserving other moments such as density and parallel momentum is implemented in ORB5. We also tested a “mode switching” approach consisting of pursuing gradient-driven simulations in flux-driven mode. This is done to reach more quickly a quasi-steady state and thus limit the numerical resources needed. This approach is tested on production-relevant cases and compared against both pure gradient-driven and pure flux-driven runs to ensure the consistency of the method. Finally, the heat

source is characterized using a 2D phase-space diagnostics.

This thesis is organized as follows. The gyrokinetic model of ORB5 and in particular the upgraded hybrid electron model that corrects the former hybrid model are presented in Chapter 2. In Chapter 3, the numerical implementation of ORB5 as well as the refactoring and optimization processes are described with a short explanation of the different optimization techniques used during this process. The linear and nonlinear validation of the upgraded hybrid electron model is presented in Chapter 4. This chapter also includes a short discussion about radial structures appearing at lowest-order mode rational surfaces and a TCV simulation using the upgraded hybrid model is presented as well. In Chapter 5, the methodology used to perform flux-driven simulation with ORB5 is presented with some illustrative “proof-of-principle” nonlinear simulations. Furthermore, the different problems encountered with various heat sources/sinks are documented. Finally, Chapter 6 summarizes this work and exposes the conclusions and future perspectives.



## 2 The ORB5 gyrokinetic model

In this chapter, the ORB5 gyrokinetic model is presented including its variants. First, the magnetic geometry, the coordinate system and the normalizations are presented. Then, using a variational approach, the gyrokinetic Vlasov-Maxwell system is derived in a consistent manner which ensures, at least theoretically, energy conservation. Different variants of the physical model are then presented. They include approximations for the polarization density in the quasineutrality equation as well as various electron models. Finally, the strong flows, collisions, and conservation laws are presented. Note that, even though the whole ORB5 model is presented here for completeness, only simulations using the electrostatic limit without collisions and strong flows have been carried out in this work.

### 2.1 Magnetic geometry, coordinate system, and normalization

The background fields of a tokamak are usually approximated as axisymmetric. A general axisymmetric magnetic field in the nested-flux-surface region may be expressed as

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

where  $F(\psi)$  is the poloidal current flux function,  $\psi$  is the poloidal magnetic flux and  $\varphi$  is the toroidal angle. The ORB5 code uses ideal-MHD equilibria, solutions of the Grad-Shafranov equation, that are provided by the CHEASE code [26]. It can also use an analytical *ad-hoc* approximated magnetic equilibrium comprising circular concentric magnetic surfaces.

A straight-field-line coordinate system is used in ORB5. The magnetic surfaces are labeled by  $s = \sqrt{\psi/\psi_{\text{edge}}}$  where  $\psi_{\text{edge}}$  is the value of  $\psi$  at the radial edge, the toroidal angle is  $\varphi$ , and the straight-field-line poloidal angle is defined by

$$\theta^* = \frac{1}{q(s)} \int_0^\theta \frac{\mathbf{B} \cdot \nabla\varphi}{\mathbf{B} \cdot \nabla\theta'} d\theta', \quad (2.2)$$

where  $q(s)$  is the safety factor profile and  $\theta$  is the geometric poloidal angle.

All the physical quantities in ORB5 are normalized according to four reference parameters;

these normalizations are used internally and in the code output. The reference quantities are the ion mass  $m_i$ , the ion charge  $q_i = eZ_i$  with  $e$  being the elementary charge and  $Z_i$  the ion atomic number, the magnetic field amplitude at the magnetic axis  $B_0$ , and the electron temperature  $T_e(s_0)$  at a reference magnetic surface  $s_0$ . Note that for simulations with multiple ion species, the user must define a reference ion species for the normalization. Derived units are then defined with respect to these four parameters: time is in units of the inverse of the ion cyclotron frequency  $\Omega_{ci} = q_i B_0 / m_i c$  with  $c$  the speed of light in vacuum (CGS units are used in the following), velocities are normalized to the ion sound velocity  $c_s = \sqrt{e T_e(s_0) / m_i}$ , lengths are given in units of the ion sound Larmor radius  $\rho_s = c_s / \Omega_{ci}$ , and the densities are normalized to the volume averaged density  $\bar{n}$ . These reference quantities are then used to construct normalizations for other quantities in the code such as the electrostatic potential, various fluxes, etc.

### 2.2 Gyrokinetic equations for fields and particles

The gyrokinetic Vlasov-Maxwell model implemented in ORB5 is derived from a variational principle [50, 51] which has some advantages compared to models implemented in other gyrokinetic codes which are derived outside such a variational framework. The first advantage is the possibility to include all necessary approximations into the expression of the action before deriving the equations of motion. The second advantage lies in the possibility to consistently derive exactly conserved quantities, corresponding to the model, such as the energy. In the ORB5 code these quantities are then used for diagnostics and the verification of the quality of the simulations. Finally, the variational formulation directly provides the weak form of gyrokinetic Poisson and Ampère equations suitable for a finite element discretization.

The choice of the ordering plays a crucial role in defining the complexity of the gyrokinetic model, and in particular the nonlinear terms which are taken into account. The gyrokinetic variational principle corresponding to the ORB5 model is established according to the specific gyrokinetic ordering achievable for numerical implementation. In particular, it means that all the geometrical effects due to the non-uniformity of the background magnetic field are considered one order smaller than the relative fluctuations of the electromagnetic fields. To quantify that statement, we define the magnetic field geometry-related small parameter  $\epsilon_B = \rho_{th} / L_B$ , where  $\rho_{th}$  is the thermal Larmor radius of the particle and  $L_B = |\nabla B / B|^{-1}$  sets up the length scale of the background magnetic field variation. The electromagnetic-fluctuations-related small parameter is defined by  $\epsilon_\delta \sim |\mathbf{B}_1| / B \sim c |\mathbf{E}_{1\perp}| / (B v_{th}) \sim (k_\perp \rho_{th}) e \phi_1 / T_i \equiv \epsilon_\perp e \phi_1 / T_i$ , where  $\mathbf{E}$  is the electric field,  $v_{th}$  is the thermal velocity,  $k_\perp$  is the wave number perpendicular to the magnetic field,  $B$  is the amplitude of the background magnetic field,  $\phi_1$  is the perturbed electrostatic potential,  $T_i$  is the ion temperature, the subscript 1 refers to the fluctuating part of the corresponding fields and the subscript  $\perp$  represents the component perpendicular to the magnetic field line. The parameter  $\epsilon_\perp$  allows the distinction between the gyrokinetic theory with  $\epsilon_\perp \sim \mathcal{O}(1)$  and the drift-kinetic theory with  $\epsilon_\perp \ll 1$ . Both models are implemented in the

## 2.2. Gyrokinetic equations for fields and particles

code ORB5.

As shown in [51, 52] the ordering  $\epsilon_B = \epsilon_\delta^2$  corresponds to gyrokinetic models implemented in most global codes. In particular, it has been demonstrated that the ORB5 equations can be derived via variational calculation from the field-particle Lagrangian accurate to second order with respect to the parameter  $\epsilon_\delta$ . Below we present the variational framework and summarize the main gyrokinetic equations resulting from the variational derivation.

The expression of the action functional leading to the ORB5 code gyrokinetic Maxwell-Vlasov equations containing first order geometric corrections, i.e.  $\mathcal{O}(\epsilon_B)$  terms, and the electromagnetic corrections up to the second order, i.e.  $\mathcal{O}(\epsilon_\delta^2)$  terms, is given by:

$$\begin{aligned} \mathcal{A} = \int_{t_0}^{t_1} dt \mathcal{L} &= \sum_{\sigma} \int dt d\Omega \left( \frac{q_{\sigma}}{c} \mathbf{A}^* \cdot \dot{\mathbf{X}} + \frac{m_{\sigma} c}{q_{\sigma}} \mu \dot{\Theta} - H_0 \right) f_{\sigma} \\ &- \epsilon_{\delta} \sum_{\sigma \neq e} \int dt d\Omega H_1 f_{\sigma} - \epsilon_{\delta} \int dt d\Omega H_1^{\text{dk}} f_e \\ &- \epsilon_{\delta}^2 \sum_{\sigma \neq e} \int dt d\Omega H_2 f_{\text{eq},\sigma} - \alpha \epsilon_{\delta}^2 \int dt d\Omega H_2^{\text{dk}} f_{\text{eq},e} - \alpha \epsilon_{\delta}^2 \int dt dV \frac{|\nabla_{\perp} A_{1\parallel}|^2}{8\pi}, \end{aligned} \quad (2.3)$$

where  $\alpha = 0$  corresponds to the electrostatic model and  $\alpha = 1$  to the electromagnetic model,  $d\Omega = dV dW$  with  $dV = d^3\mathbf{X}$  and  $dW = B_{\parallel}^* d\mu dp_z$  represents the infinitesimal volume of the reduced (gyrocenter) phase space,  $B_{\parallel}^*$  is defined as the parallel component of the symplectic magnetic field  $\mathbf{B}^* = \nabla \times \mathbf{A}^*$  with  $\mathbf{A}^* = \mathbf{A} + (c/q_{\sigma}) p_z \hat{\mathbf{b}}$  being the symplectic magnetic potential and  $\hat{\mathbf{b}}$  being the unit vector parallel to the unperturbed magnetic field line. The action is derived using the  $p_z$  formulation in which we define the reduced gyrocenter position  $\mathbf{X}$ , the canonical gyrocenter momentum  $p_z = m_{\sigma} v_{\parallel} + \alpha \epsilon_{\delta} (q_{\sigma}/c) A_{1\parallel}$  with the parallel velocity  $v_{\parallel}$ , the magnetic moment  $\mu$  and the fast gyro angle  $\Theta$ . The sums are made over all the species  $\sigma$  except for the second and third sums where the electrons are excluded because they are treated as drift-kinetic, i.e. their Larmor radius is neglected and they are identified at their guiding center. The first five terms of the gyrokinetic action are gyrocenter contributions and the last term is a contribution from the perturbed magnetic field.

Before presenting the equations of motion implemented in ORB5, we discuss all necessary approximations included in the gyrokinetic action given by Eq. (2.3). The first three terms of the action involves the full distribution functions  $f_{\sigma}$ , while the fourth and fifth terms, involving the nonlinear Hamiltonian  $H_2$ , involve equilibrium distribution functions  $f_{\text{eq},\sigma}$ , which are by definition invariant under the unperturbed Hamiltonian dynamics, i.e. they satisfy the

condition  $\{f_{\text{eq},\sigma}, H_0\} = 0$ , where  $\{F, G\}$  is the Poisson bracket defined by [53]:

$$\{F, G\} = \frac{e}{mc} \left( \frac{\partial F}{\partial \Theta} \frac{\partial G}{\partial \mu} - \frac{\partial F}{\partial \mu} \frac{\partial G}{\partial \Theta} \right) + \frac{\mathbf{B}^*}{B_{\parallel}^*} \cdot \left( \nabla F \frac{\partial G}{\partial p_z} - \frac{\partial F}{\partial p_z} \nabla G \right) \quad (2.4)$$

$$- \frac{c\hat{\mathbf{b}}}{eB_{\parallel}^*} \cdot (\nabla F \times \nabla G) - \frac{\partial F}{\partial w} \frac{\partial G}{\partial t} + \frac{\partial F}{\partial t} \frac{\partial G}{\partial w}, \quad (2.5)$$

where  $w$  is the canonically conjugate variable to  $t$ . This approximation brings several simplifications in the model. First, it results in the linearization of the gyrokinetic Poisson and Ampère equations. Second, it simplifies the gyrokinetic Vlasov equation by excluding some nonlinear terms from the gyrocenter characteristics associated with the Hamiltonian  $H_2$ .

The gyrocenter model is fixed via the Hamiltonian terms  $H_0$  (non-perturbed dynamics),  $H_1$  (linear gyrocenter dynamics),  $H_1^{\text{dk}}$  (linear drift-kinetic dynamics for electrons), and  $H_2$  (nonlinear second order gyrocenter dynamics). The choice of the linear  $H_1$ ,  $H_1^{\text{dk}}$  and nonlinear Hamiltonians  $H_2$  determines the expressions for the gyrokinetic charge and current in the reduced Poisson and Ampère equations. In this section we present the general electromagnetic model of the ORB5 code. For further options and approximations implemented on the level of the reduced particle dynamics, see the sections below.

Concerning the field part of the gyrokinetic action, three approximations have been made. First of all, the quasi-neutrality approximation, which allows one to neglect the perturbed electric field energy  $-\epsilon_{\delta}^2 \int dt dV |E_1|^2 / 8\pi$ . The second approximation consists of neglecting the magnetic compressibility of perturbations, i.e. the parallel component  $B_{1\parallel} = \epsilon_{\delta} |\mathbf{B}_{1\perp}|$  of the perturbed magnetic field is neglected and only the perpendicular part of the perturbed magnetic field  $\mathbf{B}_{1\perp} = \hat{\mathbf{b}} \times \nabla A_{1\parallel}$ , associated with  $A_{1\parallel}$ , is implemented. Finally, due to the chosen ordering, the background component of the magnetic field can be excluded from the Maxwell part of the gyrokinetic action.

The background Hamiltonian contains information about the kinetic energy of a charged particle moving in the unperturbed magnetic field with amplitude  $B$ :

$$H_0 = \frac{p_z^2}{2m_{\sigma}} + \mu B. \quad (2.6)$$

The linearized Hamiltonian model for ions is given by the gyroaveraged linear electromagnetic potential:

$$H_1 = q_{\sigma} \left\langle \phi_1 - \alpha A_{1\parallel} \frac{p_z}{m_{\sigma} c} \right\rangle, \quad (2.7)$$

where  $\langle \dots \rangle$  is the gyroaveraging operator. The gyroaveraging is absent in the linear Hamiltonian



model for the electrons which are considered as drift-kinetic:

$$H_1^{\text{dk}} = -e \left( \phi_1(\mathbf{X}) - \alpha A_{1\parallel}(\mathbf{X}) \frac{p_z}{m_e c} \right). \quad (2.8)$$

The nonlinear Hamiltonian model which contains all orders in finite Larmor radius (FLR) in its electrostatic part and up to second order FLR terms in its electromagnetic part is considered for ions only:

$$\begin{aligned} H_2 = & -\frac{q_\sigma^2}{2B} \frac{\partial}{\partial \mu} \left\langle \tilde{\phi}_1(\mathbf{X} + \boldsymbol{\rho}_0)^2 \right\rangle \\ & + \alpha \frac{q_\sigma^2}{2m_\sigma c^2} \left[ A_{1\parallel}(\mathbf{X})^2 + m_\sigma \left( \frac{c}{q_\sigma} \right)^2 \frac{\mu}{B} A_{1\parallel}(\mathbf{X}) \nabla_\perp^2 A_{1\parallel}(\mathbf{X}) \right], \end{aligned} \quad (2.9)$$

where  $\tilde{\phi}_1 = \phi_1 - \langle \phi \rangle$  represents the fluctuating part of a perturbed electrostatic potential and  $\boldsymbol{\rho}_0$  is the lowest order guiding-center displacement. Finally the second order Hamiltonian for the electrons contains the first FLR correction to the electromagnetic potential only:

$$H_2^{\text{dk}} = \alpha \frac{e^2}{2m_e c^2} A_{1\parallel}(\mathbf{X})^2.$$

### 2.2.1 Quasineutrality and Ampère equations

The corresponding quasineutrality equation in a weak form is derived using a functional derivative of the gyrokinetic action, Eq. (2.3):

$$\sum_{\sigma \neq e} \mathcal{Q}_\sigma^{\text{gyr}} + \mathcal{Q}_e^{\text{dk}} = \sum_{\sigma \neq e} \mathcal{Q}_\sigma^{\text{pol}}, \quad (2.10)$$

$$\mathcal{Q}_\sigma^{\text{gyr}} = \int d\Omega f_\sigma q_\sigma \langle \hat{\phi}_1 \rangle, \quad (2.11)$$

$$\mathcal{Q}_e^{\text{dk}} = - \int d\Omega f_e e \hat{\phi}_1(\mathbf{X}), \quad (2.12)$$

$$\mathcal{Q}_\sigma^{\text{pol}} = \epsilon_\delta \int d\Omega f_{e,q,\sigma} \frac{q_\sigma^2}{B} \frac{\partial}{\partial \mu} \left( \langle \phi_1 \hat{\phi}_1 \rangle - \langle \phi_1 \rangle \langle \hat{\phi}_1 \rangle \right), \quad (2.13)$$

where  $\hat{\phi}_1$  represents an arbitrary test function, which in the case of ORB5 is a B-spline of a required order for the finite element discretization. On the left-hand side of the equation,  $\mathcal{Q}_\sigma^{\text{gyr}}$  represents the gyro-charge density of the ions,  $\mathcal{Q}_e^{\text{dk}}$  the drift-kinetic charge of the electrons and on the right-hand side,  $\mathcal{Q}_\sigma^{\text{pol}}$  represents the linear ion polarization charge. Note that due to the drift-kinetic approximation used for the electrons, there is no contribution to the polarization density from the electron species.

Similarly, the Ampère equation derived from the variational principle is given by

$$\begin{aligned}
 0 = & - \epsilon_\delta \int \frac{dV}{4\pi} \nabla_\perp A_{1\parallel} \cdot \nabla_\perp \hat{A}_{1\parallel} + \sum_{\sigma \neq e} \int d\Omega f_\sigma \frac{q_\sigma p_z}{m_\sigma c} \langle \hat{A}_{1\parallel} \rangle - \int d\Omega f_e \frac{e p_z}{m_e c} \hat{A}_{1\parallel} \quad (2.14) \\
 & - \epsilon_\delta \int d\Omega f_{eq,e} \left( \frac{e^2}{m_e c^2} A_{1\parallel} \hat{A}_{1\parallel} \right) \\
 & - \sum_{\sigma \neq e} \epsilon_\delta \int d\Omega f_{eq,\sigma} \left[ \frac{q_\sigma^2}{m_\sigma c^2} A_{1\parallel} \hat{A}_{1\parallel} + \frac{\mu}{2B} (A_{1\parallel} \nabla_\perp^2 \hat{A}_{1\parallel} + \hat{A}_{1\parallel} \nabla_\perp^2 A_{1\parallel}) \right],
 \end{aligned}$$

for any test function  $\hat{A}_{1\parallel}$ .

## 2.2.2 Nonlinear gyrokinetic Vlasov equation

The gyrokinetic Vlasov equation for the distribution function  $f_\sigma$  of each species  $\sigma$  is reconstructed from the linearized gyrocenter characteristics according to the approximations performed on the action functional given by Eq. (2.3):

$$0 = \frac{df_\sigma}{dt} = \frac{\partial f_\sigma}{\partial t} + \dot{\mathbf{X}} \cdot \nabla f_\sigma + \dot{p}_z \frac{\partial f_\sigma}{\partial p_z}, \quad (2.15)$$

where the gyrocenter characteristics are derived from the Euler-Lagrange equation and depend on the linearized Hamiltonian model:

$$\dot{\mathbf{X}} = \frac{c\hat{\mathbf{b}}}{q_\sigma B_\parallel^*} \times \nabla H + \frac{\partial H}{\partial p_z} \frac{\mathbf{B}^*}{B_\parallel^*}, \quad (2.16)$$

$$\dot{p}_z = -\frac{\mathbf{B}^*}{B_\parallel^*} \cdot \nabla H, \quad (2.17)$$

with  $H = H_0 + \epsilon_\delta H_1$ , where  $H_0$  is a Hamiltonian corresponding to the non perturbed guiding-center dynamics given by Eq. (2.6) and  $H_1$  corresponds to the first order gyrocenter contributions given by Eq. (2.7).

For the ordering considered above, the characteristics become:

$$\dot{\mathbf{X}} = \frac{c\hat{\mathbf{b}}}{q_\sigma B_\parallel^*} \times \nabla \left[ \mu B + \epsilon_\delta q_\sigma \left( \langle \phi_1 \rangle - \alpha \frac{p_z}{m_\sigma} \langle A_{1\parallel} \rangle \right) \right] + \frac{\mathbf{B}^*}{B_\parallel^*} \left( \frac{p_z}{m_\sigma} - \epsilon_\delta \alpha \frac{q_\sigma}{m_\sigma} \langle A_{1\parallel} \rangle \right), \quad (2.18)$$

$$\dot{p}_z = -\frac{\mathbf{B}^*}{B_\parallel^*} \cdot \nabla \left[ \mu B + \epsilon_\delta q_\sigma \left( \langle \phi_1 \rangle - \alpha \frac{p_z}{m_\sigma} \langle A_{1\parallel} \rangle \right) \right], \quad (2.19)$$

## 2.2. Gyrokinetic equations for fields and particles

which can be written in a different form to make the usual drift velocities appear:

$$\dot{\mathbf{X}} = \frac{p_z}{m_\sigma} \hat{\mathbf{b}} - \frac{c p_z^2}{q_\sigma m_\sigma B_\parallel^*} \left[ \hat{\mathbf{b}} \times \left( \hat{\mathbf{b}} \times \frac{\nabla \times \mathbf{B}}{B} \right) \right] + \frac{c}{q_\sigma B_\parallel^*} \left( \mu B + \frac{p_z^2}{m_\sigma} \right) \hat{\mathbf{b}} \times \frac{\nabla B}{B} \quad (2.20)$$

$$+ \epsilon_\delta \frac{c}{B_\parallel^*} \hat{\mathbf{b}} \times \nabla \left( \langle \phi_1 \rangle - \alpha \frac{p_z}{m_\sigma} \langle A_{1\parallel} \rangle \right) + \epsilon_\delta \alpha \frac{c p_z}{m_\sigma B_\parallel^*} \langle A_{1\parallel} \rangle \boldsymbol{\kappa} - \epsilon_\delta \alpha \frac{q_\sigma}{m_\sigma} \langle A_{1\parallel} \rangle \hat{\mathbf{b}} \\ \equiv \mathbf{v}_\parallel + \mathbf{v}_D + \mathbf{v}_{\nabla B} + \mathbf{v}_C + \mathbf{v}_{E \times B} + \mathbf{v}_{A_\parallel}, \quad (2.21)$$

where  $\boldsymbol{\kappa}$  is the curvature vector

$$\boldsymbol{\kappa} = \hat{\mathbf{b}} \times \left[ \hat{\mathbf{b}} \times \frac{\nabla \times \mathbf{B}}{B} \right] + \frac{\nabla B \times \hat{\mathbf{b}}}{B}. \quad (2.22)$$

The first term of the equation is the parallel velocity  $\mathbf{v}_\parallel$  (plus a first order correction term for the electromagnetic case), the second is the diamagnetic drift  $\mathbf{v}_D$ , the third term can be separated in the  $\nabla B$  drift  $\mathbf{v}_{\nabla B}$  and curvature drift  $\mathbf{v}_C$ , the fourth is the  $E \times B$  drift  $\mathbf{v}_{E \times B}$ , and the last two terms are labeled as  $\mathbf{v}_{A_\parallel}$ . Similarly, the same procedure can be applied to the  $p_z$  characteristic:

$$\dot{p}_z = \mu B \nabla \cdot \mathbf{B} - \frac{c p_z \mu}{q_\sigma B_\parallel^*} \left[ \hat{\mathbf{b}} \times \left( \hat{\mathbf{b}} \times \frac{\nabla \times \mathbf{B}}{B} \right) \right] \cdot \nabla B \quad (2.23)$$

$$- \epsilon_\delta \nabla \left( \langle \phi_1 \rangle - \alpha \frac{p_z}{m_\sigma} \langle A_{1\parallel} \rangle \right) \cdot \left( q_\sigma \hat{\mathbf{b}} + \frac{c p_z}{B_\parallel^*} \boldsymbol{\kappa} \right) \quad (2.24)$$

$$\equiv - \frac{m_\sigma}{p_z} (\mathbf{v}_\parallel + \mathbf{v}_D + \mathbf{v}_C) \cdot \nabla \left( \mu B + \epsilon_\delta q_\sigma \langle \phi_1 \rangle - \epsilon_\delta \alpha \frac{p_z}{m_\sigma} \langle A_{1\parallel} \rangle \right). \quad (2.25)$$

In the ORB5 gyrokinetic model, different additional approximations can be made on the total time derivative operator introduced in Eq. (2.15): the linear and/or neoclassical limits. To this end, the characteristic equations (2.18) and (2.19) are slightly modified. In the linear limit, all the perturbed terms, proportional to  $\epsilon_\delta$ , are neglected leading to:

$$\dot{\mathbf{X}}^{\text{lin}} = \mathbf{v}_\parallel + \mathbf{v}_D + \mathbf{v}_{\nabla B} + \mathbf{v}_C, \quad (2.26)$$

$$\dot{p}_z^{\text{lin}} = -\mu \frac{m_\sigma}{p_z} (\mathbf{v}_\parallel + \mathbf{v}_D + \mathbf{v}_C) \cdot \nabla B. \quad (2.27)$$

The neoclassical limit is made neglecting the fluctuating electromagnetic fields and assuming small banana widths compared to the characteristic lengths of the system which leads to neglecting all drift velocities compared to the parallel drift velocity:

$$\dot{\mathbf{X}}^{\text{neo}} = \mathbf{v}_\parallel, \quad (2.28)$$

$$\dot{p}_z^{\text{neo}} = -\mu \frac{m_\sigma}{p_z} \mathbf{v}_\parallel \cdot \nabla B. \quad (2.29)$$

## 2.3 Variants of the physical models

In this section, we present the different variants of the physical model presented above that are available in the ORB5 code. Usually, each variant can be obtained in the framework of the variational formulation by changing the  $H_0$ ,  $H_1$  and  $H_2$  Hamiltonians according to the corresponding approximations. This is the case for the long-wavelength approximated electromagnetic model as well as the electrostatic models with a Padé approximation and a strong background flow. For the adiabatic electron model, an external coupling of the gyrokinetic equations with a fluid polarization density of the electrons is assumed. Including this model into the general framework requires some additional approximations on the field term of the field-particles Lagrangian given by Eq. (2.3). Note that these models are not necessarily mutually exclusive and a summary of the different possible combinations will be presented at the end of the section.

### 2.3.1 Long wavelength approximation

This approximation is obtained by replacing the second order nonlinear Hamiltonian  $H_2$  given by Eq. (2.9) in the gyrokinetic Lagrangian, Eq. (2.3), by the nonlinear Hamiltonian model [52] containing FLR expansions up to the second order for both its electrostatic and electromagnetic parts:

$$H_2^{\text{FLR}} = -\frac{m_\sigma c^2}{2B^2} |\nabla_\perp \phi_1(\mathbf{X})|^2 + \alpha \frac{q_\sigma^2}{2m_\sigma c^2} \left[ A_{1\parallel}(\mathbf{X})^2 + m_\sigma \left( \frac{c}{q_\sigma} \right)^2 \frac{\mu}{B} A_{1\parallel} \nabla_\perp^2 A_{1\parallel}(\mathbf{X}) \right]. \quad (2.30)$$

This changes only the term associated with the polarization charge of the quasineutrality equation, Eq. (2.10), so that Eq. (2.13) is replaced with

$$\mathcal{Q}_{\sigma, \text{LWA}}^{\text{pol}} = \epsilon_\delta \int d\Omega f_{\text{eq}, \sigma} \frac{m_\sigma c^2}{B^2} \nabla_\perp \phi_1 \cdot \nabla_\perp \hat{\phi}_1, \quad (2.31)$$

for any test function  $\hat{\phi}_1$ . The subscript LWA stands for long wavelength approximation. Since the magnetic terms in Eq. (2.30) remain unchanged compared to the Hamiltonian  $H_2$  given by Eq. (2.9), for which the long wavelength approximation had already been considered, the corresponding Ampère equation remains the same as given by Eq. (2.14). The gyrokinetic Vlasov equation is unchanged as well, since the background  $H_0$  and linear  $H_1$  Hamiltonians are not affected by the approximation and no contributions from the second order Hamiltonian appear in the characteristics given by Eqs. (2.18)–(2.19).

### 2.3.2 Padé approximation

In addition to the long wavelength approximation, a Padé-approximated quasineutrality model for the ion species is available in ORB5 [54, 55]. In practice, the Padé approximation is currently only implemented for one ion species ( $\sigma = \text{i}$ ). In order to include this approximation inside the common variational principle, the linear Hamiltonian model has to be slightly

modified with respect to Eq. (2.7) for both ions:

$$H_{1,\text{Padé}} = (1 - \nabla_{\perp} \cdot \rho_i^2 \nabla_{\perp}) H_1, \quad (2.32)$$

and electrons:

$$H_{1,\text{Padé}}^{\text{dk}} = (1 - \nabla_{\perp} \cdot \rho_i^2 \nabla_{\perp}) H_1^{\text{dk}}. \quad (2.33)$$

The nonlinear Hamiltonian model in that case is given by the FLR second-order truncated Hamiltonian  $H_2^{\text{FLR}}$ , Eq. (2.30). The quasineutrality equation in weak form is written in a different way by multiplying it by the operator  $[1 - \nabla_{\perp} \cdot \rho_i^2 \nabla_{\perp}]$  to cancel the  $[1 - \nabla_{\perp} \cdot \rho_i^2 \nabla_{\perp}]^{-1}$  term in the polarization density. This is done for computational reasons: the inverse of the block banded matrix coming from the discretization of the  $[1 - \nabla_{\perp} \cdot \rho_i^2 \nabla_{\perp}]$  operator is a full matrix. For example, with drift-kinetic electrons, this leads to:

$$\begin{aligned} 0 &= q_i \int d\Omega f_i (1 - \nabla_{\perp} \cdot \rho_i^2 \nabla_{\perp}) \langle \hat{\phi}_1 \rangle - e \int d\Omega f_e (1 - \nabla_{\perp} \cdot \rho_i^2 \nabla_{\perp}) \hat{\phi}_1 \\ &+ \epsilon_{\delta} \int d\Omega f_{\text{eq},i} \frac{m_i c^2}{B^2} \nabla_{\perp} \phi_1 \cdot \nabla_{\perp} \hat{\phi}_1. \end{aligned} \quad (2.34)$$

### 2.3.3 Adiabatic electron model

In order to include a model with adiabatic electrons inside the variational formulation, we need to include a fluid approximation for the electron dynamics inside the field-particle Lagrangian. Compared to the main field-particle Lagrangian, Eq. (2.3), here the sum over the species in the first term is over the ion species only and the field term is modified by a purely electrostatic contribution from the electrons. The action principle for a model with adiabatic electrons is then given by

$$\begin{aligned} \mathcal{A}_{\text{adiab}} &= \int dt \mathcal{L}_{\text{adiab}} = \sum_{\sigma \neq e} \int dt d\Omega \left[ \frac{q_{\sigma}}{c} \mathbf{A}^* \cdot \dot{\mathbf{X}} + \frac{m_{\sigma} c}{q_{\sigma}} \mu \dot{\Theta} - (H_0 + \epsilon_{\delta} H_1) \right] f_{\sigma} \\ &+ \epsilon_{\delta} \int dt dV \left[ n_{e0} \phi_1 + \epsilon_{\delta} \frac{e}{2T_e} n_{e0} (\phi_1 - \bar{\phi}_1)^2 \right] \\ &- \epsilon_{\delta}^2 \sum_{\sigma \neq e} \int dt d\Omega H_2 f_{\text{eq},\sigma}, \end{aligned} \quad (2.35)$$

where  $\bar{\phi}_1$  represents the flux-surface-averaged electric potential given by

$$\bar{\phi}_1 \equiv \frac{\int \phi_1 J_{s\theta^* \varphi}(s, \theta^*) d\theta^* d\varphi}{\int J_{s\theta^* \varphi}(s, \theta^*) d\theta^* d\varphi}, \quad (2.36)$$

where  $J_{s\theta^* \varphi}(s, \theta^*) = \nabla s \cdot (\nabla \theta^* \times \nabla \varphi)$  is the Jacobian of the magnetic coordinate transformation and  $n_{e0}$  is the equilibrium electron density. Since the adiabatic electron model is only valid

## Chapter 2. The ORB5 gyrokinetic model

---

in the electrostatic limit,  $p_z = m_\sigma v_\parallel$  and the velocity part of the phase space volume reduces to  $dW = B_\parallel^* m_\sigma dv_\parallel d\mu$  and  $\mathbf{B}^* = \nabla \times \mathbf{A}^*$  with  $\mathbf{A}^* = \mathbf{A} + (c/q_\sigma) m_\sigma v_\parallel \hat{\mathbf{b}}$ , while the spatial part  $dV = d^3\mathbf{X}$  remains unchanged with respect to the electromagnetic case. The Hamiltonian models are now defined for a simplified electrostatic case as

$$H_0 = \frac{m_\sigma v_\parallel^2}{2} + \mu B, \quad (2.37)$$

$$H_1 = q_\sigma \langle \phi_1 \rangle. \quad (2.38)$$

The nonlinear ion dynamics is defined by the electrostatic part of either the full FLR, the Padé-approximated, or the second order FLR long-wavelength-approximated nonlinear Hamiltonian.

The corresponding Vlasov equation does not contain any contribution from the electron species, so we have for ions ( $\sigma = i$ )

$$0 = \frac{df_i}{dt} = \frac{\partial f_i}{\partial t} + \dot{\mathbf{X}} \cdot \nabla f_i + \dot{v}_\parallel \frac{\partial f_i}{\partial v_\parallel}, \quad (2.39)$$

with the characteristics corresponding to the electrostatic limit ( $\alpha = 0$ ) of Eqs. (2.16) and (2.17):

$$\begin{aligned} \dot{\mathbf{X}} &= \frac{c\hat{\mathbf{b}}}{q_i B_\parallel^*} \times \nabla (\mu B + \epsilon_\delta q_i \langle \phi_1 \rangle) + \frac{\mathbf{B}^*}{B_\parallel^*} v_\parallel, \\ \dot{v}_\parallel &= -\frac{\mathbf{B}^*}{B_\parallel^*} \cdot \nabla (\mu B + \epsilon_\delta q_i \langle \phi_1 \rangle). \end{aligned} \quad (2.40)$$

For the quasineutrality equation, only the electron contribution to the gyro-charge density term is modified, leading to

$$\mathcal{Q}_{e,\text{adiab}}^{\text{dk}} = \epsilon_\delta \int dV \frac{en_{e0}}{T_e} (\phi_1 - \bar{\phi}_1) \hat{\phi}_1 + \int dV n_{e0} \hat{\phi}_1. \quad (2.41)$$

### 2.3.4 Hybrid electron model

There is also the possibility to include a hybrid electron model inside the variational formulation. In that case the fraction of passing electrons designated with a coefficient  $\alpha_P$  is treated as an adiabatic species, while the fraction of trapped electrons is treated as a drift-kinetic species. At the same time, the ions are treated as kinetic species. The corresponding action functional

is given by

$$\begin{aligned} \mathcal{A}_{\text{hybrid}} = \int dt \mathcal{L}_{\text{hybrid}} = \sum_{\sigma \neq e} \int dt d\Omega \left[ \frac{q_\sigma}{c} \mathbf{A}^* \cdot \dot{\mathbf{X}} + \frac{m_\sigma c}{q_\sigma} \mu \dot{\Theta} - (H_0 + \epsilon_\delta H_1) \right] f_\sigma \\ - \epsilon_\delta^2 \sum_{\sigma \neq e} \int dt d\Omega H_2 f_{\text{eq},\sigma} \\ + \int dt dV \int_{\text{trapped}} dW \left[ \frac{e}{c} \mathbf{A}^* \cdot \dot{\mathbf{X}} + \frac{m_e c}{e} \mu \dot{\Theta} - (H_0 + \epsilon_\delta H_1^{\text{dk}}) \right] f_e \\ + \epsilon_\delta \int dt dV \alpha_P \left[ n_{e0} \phi_1 + \epsilon_\delta \frac{e}{2T_e} n_{e0} (\phi_1 - \bar{\phi}_1)^2 \right], \end{aligned}$$

where the integral over the fraction of trapped electrons in the velocity phase space is assumed with  $\int_{\text{trapped}} dW$ . The trapped/passing boundary is defined with respect to equilibrium fields and neglecting radial drifts, Eq. (1.14). Note that for electrons, the banana width is neglected and the amplitude of the magnetic field is taken locally at the electron position. The phase space configuration is the same as in the case of an adiabatic electron model. The gyrocenter model used for modelling the ion species dynamics is identical to the one presented for the adiabatic electron model discussed in the previous section, i.e. the Hamiltonians  $H_0$  and  $H_1$  are given by Eqs. (2.37)–(2.38) and the nonlinear Hamiltonian is coming from either the full FLR, the Padé approximation or the long wavelength approximation. Concerning the gyrocenter models used for modeling the hybrid electron dynamics, the equilibrium dynamics is defined with  $H_0$  given by Eq. (2.37). The linear part of the trapped electron dynamics is defined by the drift-kinetic model defined by Eq. (2.8) with  $\alpha = 0$ . The quasineutrality equation is only affected through the electron contribution to the gyro-charge term that reads

$$\mathcal{Q}_{\text{e,hyb}}^{\text{dk}} = \epsilon_\delta \int dV \alpha_P n_{e0} \frac{e}{T_e} (\phi_1 - \bar{\phi}_1) \hat{\phi}_1 + \int dV n_{e0} \hat{\phi}_1 + \int_{\text{trapped}} d\Omega f_e \hat{\phi}_1. \quad (2.42)$$

The ion characteristics are reconstructed identically to the case with adiabatic electrons, accordingly to Eq. (2.40). The characteristics for the electrons are defined by the simplified drift-kinetic equations corresponding to the dynamics of  $H = H_0 + H_1^{\text{dk}}$ :

$$\begin{aligned} \dot{\mathbf{X}} &= -\frac{c\hat{\mathbf{b}}}{eB_\parallel^*} \times \nabla (\mu B + \epsilon_\delta e \phi_1) + \frac{\mathbf{B}^*}{B_\parallel^*} v_\parallel, \\ \dot{v}_\parallel &= -\frac{\mathbf{B}^*}{B_\parallel^*} \cdot \nabla (\mu B + \epsilon_\delta e \phi_1). \end{aligned} \quad (2.43)$$

Note that all the electrons, passing and trapped, are evolved according to the above drift-kinetic characteristics.

The hybrid electron model presented above was originally implemented to simulate linear electron modes such as TEM, allowing for a larger timestep than with fully drift-kinetic electrons. However, in nonlinear regime, it does not ensure the ambipolar condition—which also

impacts the conservation of the toroidal angular momentum—as no flux-surface-averaged passing-electron density is accounted for. Spurious sources of e.g. density and momentum are added in this model because particles that are becoming trapped (detrapped) are suddenly (not) accounted for kinetically in the quasi-neutrality equation. The trapping/detrapping processes are the collisions and parallel nonlinearity that affect the particle pitch angle and radial drifts that change the value of the local background magnetic field. To correct this trapped/passing hybrid model, a part of this thesis was dedicated to implementing an upgraded hybrid electron model in ORB5 as an improvement of the model presented in [56]. The main difference between the former ORB5 model and this upgrade is that the flux-surface average of the passing electrons is also treated kinetically leading to the following weak form of the electron contribution to the quasi-neutrality equation:

$$\begin{aligned} \mathcal{Q}_{e,hyb}^{dk} = & \epsilon_\delta \int dV \alpha_P n_{e0} \frac{e}{T_e} (\phi_1 - \bar{\phi}_1) \hat{\phi}_1 + \int dV n_{e0} \hat{\phi}_1 \\ & + \int_{\text{trapped}} d\Omega f_e \hat{\phi}_1 + \int_{\text{passing}} d\Omega f_e \bar{\hat{\phi}}_1, \end{aligned} \quad (2.44)$$

where  $\bar{\hat{\phi}}_1$  is the flux-surface average of the arbitrary test function  $\hat{\phi}_1$ . As a first approximation, the flux-surface average is computed by filtering out all the  $m \neq 0$  kinetic contributions of the passing electrons for the  $n = 0$  mode while keeping an adiabatic response for the other passing components. This way, the quasineutrality equation, Eq. (2.44), is slightly changed as follows

$$\begin{aligned} \mathcal{Q}_{e,hyb}^{dk} = & \epsilon_\delta \int dV \alpha_P n_{e0} \frac{e}{T_e} (\phi_1 - \bar{\phi}_1) \hat{\phi}_1 + \int dV n_{e0} \hat{\phi}_1 \\ & + \int_{\text{trapped}} d\Omega f_e \hat{\phi}_1 + \int_{\text{passing}} d\Omega f_e \hat{\phi}_1^{00}, \end{aligned} \quad (2.45)$$

where  $\hat{\phi}_1^{00}$  is the  $n = m = 0$  component of the arbitrary test function  $\hat{\phi}_1$ . A more detailed description can be found in Chapter 4.

### 2.3.5 Summary of the models

All the variants of the particle models presented in the previous sections are summarized here. The main changes brought by the different models mainly come through the quasineutrality



equation which can be written

$$\sum_{\sigma \neq e} \mathcal{Q}_{\sigma}^{\text{gyr}} + \mathcal{Q}_e^{\text{dk}} = \sum_{\sigma \neq e} \mathcal{Q}_{\sigma}^{\text{pol}}, \quad (2.46)$$

$$\mathcal{Q}_{\sigma}^{\text{gyr}} = \int d\Omega f_{\sigma} q_{\sigma} \langle \hat{\phi}_1 \rangle, \quad (2.47)$$

$$\mathcal{Q}_e^{\text{dk}} = - \int d\Omega f_e e \hat{\phi}_1(\mathbf{X}), \quad (2.48)$$

$$\mathcal{Q}_{\sigma}^{\text{pol}} = \epsilon_{\delta} \int d\Omega f_{\text{eq},\sigma} \frac{q_{\sigma}^2}{B} \frac{\partial}{\partial \mu} \left( \langle \phi_1 \hat{\phi}_1 \rangle - \langle \phi_1 \rangle \langle \hat{\phi}_1 \rangle \right), \quad (2.49)$$

where  $\mathcal{Q}_{\sigma}^{\text{gyr}}$  is the term corresponding to the ion gyrodensity contribution,  $\mathcal{Q}_e^{\text{dk}}$  is the term corresponding to the electron drift kinetic density contribution and  $\mathcal{Q}_{\sigma}^{\text{pol}}$  represents the polarization density contribution from the ions. For the ions, only  $\mathcal{Q}_{\sigma}^{\text{pol}}$  is affected by the different models:

$$\mathcal{Q}_{\sigma, \text{LWA}}^{\text{pol}} = \epsilon_{\delta} \int d\Omega f_{\text{eq},\sigma} \frac{m_{\sigma} c^2}{B^2} \nabla_{\perp} \phi_1 \cdot \nabla_{\perp} \hat{\phi}_1, \quad (2.50)$$

$$\mathcal{Q}_{i, \text{Padé}}^{\text{pol}} = \epsilon_{\delta} \int d\Omega f_{\text{eq},i} \frac{m_i c^2}{B^2} \left[ 1 - \nabla_{\perp} \cdot \rho_i^2 \nabla_{\perp} \right]^{-1} \left[ \nabla_{\perp} \phi_1 \cdot \nabla_{\perp} \hat{\phi}_1 \right]. \quad (2.51)$$

Note that in the case of the Padé approximation, all the quasineutrality equation is multiplied by  $\left[ 1 - \nabla_{\perp} \cdot \rho_i^2 \nabla_{\perp} \right]$  to avoid having to invert it.

For the electrons, only  $\mathcal{Q}_e^{\text{dk}}$  is changed by the different fluid and hybrid approximations:

$$\mathcal{Q}_{e, \text{adiab}}^{\text{dk}} = \epsilon_{\delta} \int dV \frac{e n_{e0}}{T_e} (\phi_1 - \bar{\phi}_1) \hat{\phi}_1 - \int dV n_{e0} \hat{\phi}_1, \quad (2.52)$$

$$\begin{aligned} \mathcal{Q}_{e, \text{hyb}}^{\text{dk}} = & \alpha_P \epsilon_{\delta} \int dV \frac{e n_{e0}}{T_e} (\phi_1 - \bar{\phi}_1) \hat{\phi}_1 - \int dV n_{e0} \hat{\phi}_1 \\ & - \int_{\text{trapped}} d\Omega f_e \hat{\phi}_1 + \int_{\text{passing}} d\Omega f_e \bar{\hat{\phi}}_1. \end{aligned} \quad (2.53)$$

In ORB5, the previous approximations are not mutually exclusive, i.e. each model for the polarization density can be combined with any electron model.

## 2.4 $\delta f$ method and background distribution functions

The ORB5 code uses a  $\delta f$  control-variates approach to reduce the numerical noise due to the finite phase-space sampling [57, 58]. The rationale of this method is to separate the total distribution function into two parts: a time-independent part  $f_0$  and a time-dependent part  $\delta f$ . The first function,  $f_0$ , is supposed to be known and easily computable. Only the  $\delta f$  part is represented with a sample of “numerical particles” or “markers”. The statistical sampling error will thus be reduced, compared to a full- $f$  method, if  $|\delta f| \ll f_0$ .

In the collisionless limit and in the absence of sources, the total distribution function is

conserved along the trajectories. Using the  $\delta f$  separation, we obtain

$$\frac{d\delta f}{dt} = -\frac{df_0}{dt} = -\left.\frac{df_0}{dt}\right|_0 - \left.\frac{df_0}{dt}\right|_1, \quad (2.54)$$

where the time-derivative operator has been split into two parts labeled by 0 and 1, which respectively represent the unperturbed dynamics, i.e. without the fluctuating fields, and the perturbed dynamics. In the *standard*  $\delta f$  method, we choose  $f_0 \equiv f_{\text{eq}}$  to be an equilibrium distribution, solution of the unperturbed collisionless equations of motion and thus, to satisfy  $\{f_{\text{eq}}, H_0\} = 0$  reducing Eq. (2.54) to

$$\frac{d\delta f}{dt} = -\left.\frac{df_{\text{eq}}}{dt}\right|_1. \quad (2.55)$$

In ORB5, different choices for the initial distribution function are available. The plasma can be assumed to be in a local thermodynamic equilibrium described by a *local Maxwellian*  $f_L(\psi, \epsilon, \mu)$  defined by

$$f_L(\psi, \epsilon, \mu) = \frac{n_\sigma(\psi)}{(2\pi T_\sigma(\psi)/m_\sigma)^{\frac{3}{2}}} \exp\left[-\frac{\frac{1}{2}m_\sigma(v_\parallel^2 + 2\mu B)}{T_\sigma(\psi)}\right]. \quad (2.56)$$

Both the particle energy  $\epsilon$  and the magnetic moment  $\mu$  are constants of motion but the poloidal magnetic flux  $\psi$  is not. Note that in ORB5, the magnetic moment is defined as  $\mu = v_\perp^2/(2B)$ . The local Maxwellian is therefore not invariant under the unperturbed dynamics and Eq. (2.54) must be used. The inclusion of the  $df_L/dt|_0$  term leads to the drive of a spurious zonal flow discussed in [11], which appears already in the linear phase of a simulation, even though zonal flows are linearly stable and excited through nonlinear coupling [59]. As done in many PIC codes, the above-mentioned term responsible for this zonal flow drive can be neglected but it is not consistent with the perturbative ordering used here.

The alternative approach is to use a distribution function that is a true equilibrium, implying that  $f_0$  is a function of constants of motion only. This is verified by the so-called *canonical* Maxwellian  $f_C(\psi_0, \epsilon, \mu)$  defined by

$$f_C(\psi_0, \epsilon, \mu) = \frac{n_\sigma(\psi_0)}{(2\pi T_\sigma(\psi_0)/m_\sigma)^{\frac{3}{2}}} \exp\left[-\frac{\frac{1}{2}m_\sigma(v_\parallel^2 + 2\mu B)}{T_\sigma(\psi_0)}\right], \quad (2.57)$$

where  $\psi_0 = \psi + (m_\sigma c/q_\sigma)(F(\psi)/B)v_\parallel$  is the toroidal momentum which is conserved in an axisymmetric toroidal system. However, it is easily shown that the effective density  $n_\sigma(\psi)$  and temperature  $T_\sigma(\psi)$  computed from  $f_C$  are different from the ones,  $n_\sigma(\psi_0)$  and  $T_\sigma(\psi_0)$ , given as input and function of  $\psi_0$ . The use of a canonical Maxwellian can lead to large, unrealistic values of parallel flows preventing any instability to develop, especially for small system size

and large  $n_0$  and  $v_{\parallel}$  gradients [60]. To address this issue, a *corrected canonical Maxwellian*  $f_{CC}(\hat{\psi}, \epsilon, \mu)$  is used. A correction term is added to the toroidal momentum to minimize the gap between the local and canonical Maxwellians while still being an equilibrium distribution. The corrected toroidal momentum reads

$$\hat{\psi} = \psi_0 + \psi_{0,\text{corr}} = \psi_0 - \text{sign}(v_{\parallel}) \frac{m_{\sigma} c}{q_{\sigma}} R_0 \sqrt{2(\epsilon - \mu B_0)} \mathcal{H}(\epsilon - \mu B_0), \quad (2.58)$$

where  $R_0$  is the major radius and  $\mathcal{H}$  is the Heaviside function. The correction term is zero for trapped particles and of opposite sign for forward and backward passing particles. The corrected toroidal momentum being built only with constants of motion,  $f_{CC}(\hat{\psi}, \epsilon, \mu)$  satisfies  $\{f_{CC}, H_0\} = 0$ .

## 2.5 Strong flows

The strong flow gyrokinetic ordering allows for  $u_E / v_{\text{th},i} \sim 1$ , with  $\mathbf{u}_E = c(\hat{\mathbf{b}} \times \nabla \Phi / B)$  the background  $E \times B$  velocity, where  $\Phi$  represents the background electric potential, and  $v_{\text{th},i} = \sqrt{T_i / m_i}$  is the ion thermal velocity [61]. Implementing this ordering in ORB5 enables the treatment of plasmas rotating toroidally at close to the Mach velocity. More details of this formalism have been published earlier [62]. In this case, a further approximation is performed on the background distribution function. While a local Maxwellian is used for the polarization density in the quasineutrality equation, the canonical Maxwellian is implemented for the derivation of the gyrokinetic Vlasov equation.

In order to include the model containing a background electrostatic potential  $\Phi$  within the general field-gyrocenter action given by Eq. (2.3), the background Hamiltonian  $H_0$ , as well as the symplectic magnetic potential  $\mathbf{A}^*$  have to be consistently modified:

$$H_0^{\text{flow}} = q_{\sigma} \Phi + \mu B + \frac{p_z^2 + (m_{\sigma} \mathbf{u}_E)^2}{2m_{\sigma}}, \quad (2.59)$$

and  $\mathbf{A}^* = \mathbf{A} + (c / q_{\sigma}) p_z \hat{\mathbf{b}} + (m_{\sigma} c / q_{\sigma}) \mathbf{u}_E$ .

In ORB5, when considering strong flows, only the electrostatic limit is considered. This corresponds to setting  $\alpha = 0$  in Eq. (2.3). Remark that including the background  $E \times B$  velocity does not affect the quasineutrality equation, since no corrections due to the presence of a strong flow are included into the linear and nonlinear Hamiltonian models given by Eqs. (2.7)-(2.9). The gyrokinetic Vlasov equation is modified according to the change in the background dynamics from  $H_0$  given by Eq. (2.6) to  $H_0^{\text{flow}}$  given by Eq. (2.59). The corresponding  $\delta f$  gyrokinetic Vlasov equation is reconstructed from the modified characteristics. Since the

perturbed magnetic field is not considered,  $p_z = m_\sigma v_\parallel$  is a purely kinetic momentum:

$$\begin{aligned}\dot{\mathbf{X}} &= \frac{c\hat{\mathbf{b}}}{q_\sigma B_\parallel^*} \times \nabla \left( q_\sigma \Phi + \mu B + \frac{m_\sigma}{2} |\mathbf{u}_E|^2 + \epsilon_\delta q_\sigma \langle \phi_1 \rangle \right) + \frac{\mathbf{B}^*}{B_\parallel^*} v_\parallel, \\ \dot{v}_\parallel &= -\frac{\mathbf{B}^*}{B_\parallel^*} \cdot \nabla \left( q_\sigma \Phi + \mu B + \frac{m_\sigma}{2} |\mathbf{u}_E|^2 + \epsilon_\delta q_\sigma \langle \phi_1 \rangle \right).\end{aligned}\quad (2.60)$$

For strongly rotating plasmas, with Mach number around one, dynamic pressure resulting from the flow is comparable to the thermal pressure, and a modified Grad-Shafranov equation should be used to accurately compute the magnetic equilibrium. To self-consistently include these effects, we have used the MHD code FLOW [63] which can solve the MHD force balance equation in the presence of a background flow. FLOW reads the equilibrium via the standard EQDSK format [26]. We have considered only toroidally rotating MHD equilibria, with the temperature being a flux surface function, as this allows collisionless kinetic and MHD equilibria to be consistent in the large-system size limit.

### 2.5.1 Global gyrokinetic equilibria for rotating plasmas

The constants of motion are the magnetic moment  $\mu$ , the unperturbed energy of the particle  $\varepsilon = H_0^{\text{flow}}$ , the sign of the parallel velocity (for passing particles), and finally the toroidal canonical momentum  $\psi_0$ , which is conserved in a tokamak due to axisymmetry. The strong-flow canonical momentum  $\psi_C$  is an extension of the canonical momentum discussed in Section 2.4:

$$\psi_C^{\text{flow}} = \psi + \frac{m_\sigma c}{q_\sigma} \frac{F}{B} v_\parallel + \frac{m_\sigma c}{q_\sigma} u_\varphi, \quad (2.61)$$

where  $u_\varphi$  is the toroidal component of the background  $E \times B$  velocity.

In the presence of toroidal rotation, the canonical Maxwellian, which is corrected so that the flux surface averaged density remains close to  $n_{0\sigma}$  when rotation is introduced, is given by

$$f_C = \left( \frac{m_\sigma}{2\pi T_{0\sigma}(\psi_C)} \right)^{3/2} n_{0\sigma}(\psi_C) \exp \left\{ -\frac{1}{T_{0\sigma}(\psi_C)} \left[ H_0^{\text{flow}} + \frac{m_\sigma R_0(\psi_C)^2}{2} \left( \frac{\partial \bar{\Phi}}{\partial \psi} \right)^2 \right] \right\}, \quad (2.62)$$

where  $\bar{\Phi}$  is the flux surface average of  $\Phi$ . In the local limit, this choice leads to an in-out density variation

$$n_\sigma = n_{0\sigma}(\psi) \exp \left[ \frac{m_\sigma (R^2 - R_0^2) \Omega^2}{2} \right], \quad (2.63)$$

where the plasma rotation frequency  $\Omega$  may be expressed as  $\Omega = \partial \Phi / \partial \psi$ .

## 2.6 Collisions

The inclusion of collisions in a gyrokinetic code like ORB5 is important to assess the right level of transport. Indeed, collisions are required to model the neoclassical physics, which is a key player in the transport of certain classes of particles, e.g. heavy impurities. Furthermore, collisions are known to impact turbulence. For example, ITG driven turbulence increases when collisions are taken into account due to the collisional damping of the zonal flows [64, 65]. On the other hand, TEM turbulence is reduced by collisions via the collisional detrapping of electrons.

ORB5 currently includes ion-ion intra-species and electron-ion collisions [66]. For the collisional dynamics, FLR and polarization effects are neglected. In ORB5, collisions are represented by a linearized Landau collision operator. The linearization procedure is done with respect to a local Maxwellian background  $f_L$ . The full Landau operator describing collisions of species  $a$  on species  $b$  is bilinear with respect to the distributions  $f_a$  and  $f_b$  and may thus be decomposed into four terms:  $C_{ab}[f_b, f_a] = C_{ab}[f_{b,L}, f_{a,L}] + C_{ab}[f_{b,L}, \delta f_a] + C_{ab}[\delta f_b, f_{a,L}] + C_{ab}[\delta f_b, \delta f_a]$ , where  $\delta f_s$  is the perturbed part of the distribution of species  $s$ . Note that in our notation,  $C_{ab}[f_a, f_b]$  refers to the effect of collisions of species  $a$  on species  $b$ . Note that, for the whole collision part, the species background distribution function is converted to a local Maxwellian if it is not already the case. After the collision dynamics has been treated, the background Maxwellian is converted back to its original form if needed,  $f = f_0 + \delta f = f_{0L} + \delta f_L$ . The perturbed part  $\delta f$  is modified accordingly such that the total distribution remains unchanged in this “background switching” process. For Maxwellian distributions with identical parallel velocities and temperatures, the first term on the right-hand side is zero. Assuming the perturbation is small, the final, nonlinear term, is also neglected leaving two terms called the “test particle” term  $C_{ab}[f_{b,L}, \delta f_a]$  and the “background reaction” term  $C_{ab}[\delta f_b, f_{a,L}]$ .

For the self-collisions, the “test particle” term can be readily evaluated using the exact Landau operator in its drag-diffusion form:

$$C[f_L, \delta f] = \frac{\partial}{\partial \mathbf{v}} \cdot [\mathbf{\Gamma}(f_L) \delta f] - \frac{\partial^2}{\partial \mathbf{v} \mathbf{v}} : [\overline{\overline{D}}(f_L) \delta f], \quad (2.64)$$

where the drag vector and the diffusion tensor are respectively given by

$$\mathbf{\Gamma} = -\bar{\nu} H(x) \mathbf{v}, \quad \overline{\overline{D}} = \frac{\bar{\nu} v_{th}^2}{4} \left[ K(x) \left( \overline{\overline{I}} - \frac{\mathbf{v} : \mathbf{v}}{v^2} \right) + 2H(x) \frac{\mathbf{v} : \mathbf{v}}{v^2} \right], \quad (2.65)$$

where the collision frequency is defined as  $\bar{\nu} = 8\pi n q^4 \ln \Lambda / m^2 v_{th}^3$ ,  $x = v / \sqrt{2} v_{th}$  is the normalized velocity with  $v_{th} = \sqrt{T/m}$  the thermal velocity of the species, and  $\overline{\overline{I}}$  is the identity tensor. The Coulomb logarithm  $\ln \Lambda$  is assumed constant across the plasma, and typically has a value of 10–15. The functions  $K(x)$  and  $H(x)$  are resulting from the analytical evaluation of the

Rosenbluth potentials in the case of a Maxwellian background distribution:

$$H(x) = \frac{1}{2\sqrt{2}x^3} \left[ \text{erf}(x) - x \frac{d}{dx} \text{erf}(x) \right], \quad (2.66)$$

$$K(x) = \frac{1}{\sqrt{2}x} \phi(x) - H(x). \quad (2.67)$$

where erf stands for the error function.

Evaluating the background reaction term exactly would in particular require the reconstruction of the  $\delta f$  distribution function and taking its velocity derivative. Such a direct approach is too expensive and includes steps subject to significant noise in a PIC code. Instead, ORB5 uses an approximation first suggested by [67]:  $C[\delta f, f_L] \simeq f_L \beta(\delta f)$ , with

$$\beta(\delta f) = \frac{1}{n} \left[ 6\sqrt{\pi} H(x) \frac{\delta P_{\parallel} v_{\parallel}}{v_{\text{th}}^2} + \sqrt{\pi} G(x) \frac{\delta E}{v_{\text{th}}^2} \right], \quad (2.68)$$

where  $G(v) = (4x^2 - 1)H(x) - K(x)$ . The two terms  $\delta P_{\parallel}$  and  $\delta E$  represent respectively the parallel momentum and energy transferred to the distribution by the “test particle” operator. This approximation can be shown to satisfy the desirable properties of a collision operator [67, 68]. Indeed, it conserves the mass and, when combined with its counterpart  $C[f_L, \delta f]$ , conserves also the momentum and energy. Furthermore, the combined linear operator is self-adjoint and satisfies the H-theorem. The operator is zero if the perturbation is a shifted linearized Maxwellian, i.e. such distributions are stationary states.

The only interspecies collisions which are currently taken into account in ORB5 are the electron-ion collisions. The “test particle” part of the electron-ion collisions in ORB5 is represented by a Lorentz operator, which assumes a large mass ratio between ions and electrons. In this limit, electrons experience only pitch-angle scattering. This Lorentz operator can simply be written:

$$C_{\text{ei}}[f_i, \delta f_e] = -\nu_{\text{ei}}(v) \frac{\partial}{\partial \xi} \left[ (1 - \xi^2) \frac{\partial \delta f_e}{\partial \xi} \right], \quad (2.69)$$

where the electron-ion collision frequency is given by  $\nu_{\text{ei}}(v) = (\bar{\nu}_{\text{ei}}/4)(v_{\text{th,e}}/v)^3$ , with  $\bar{\nu}_{\text{ei}} = 8\pi n_i Z^2 e^4 \ln \Lambda / m_e^2 v_{\text{th,e}}^3$  and  $\xi = v_{\parallel} / v_{\perp}$  is the pitch angle variable. The “test particle” Lorentz operator conserves the mass and energy. The “background reaction” of the Lorentz operator is neglected in ORB5. Therefore, momentum conservation is not ensured by the reduced electron-ion collision operator.

## 2.7 Conservation laws and diagnostics

In this section we present the conserved quantities associated with the field-particle Lagrangian, which are implemented in ORB5 as diagnostic tools. These quantities can be obtained from a direct application of the Noether method, details of the derivation can be found

in [51]. We start by presenting the energy invariant corresponding to each model. This invariant is used for constructing the so-called power balance diagnostic, which allows one to verify the quality of numerical simulations.

First, we provide a generic expression for the energy density corresponding to the most complete electromagnetic model, which can also be obtained from a direct application of the Noether method, see e.g. [51].

$$\begin{aligned} \mathcal{E}^{\text{EM}} = & \sum_{\sigma} \int d\Omega H_0 f_{\sigma} + \epsilon_{\delta} \sum_{\sigma \neq e} \int d\Omega H_1 f_{\sigma} + \epsilon_{\delta} \int d\Omega H_1^{\text{dk}} f_e \\ & + \epsilon_{\delta}^2 \sum_{\sigma \neq e} \int d\Omega H_2 f_{\text{eq},\sigma} + \alpha \epsilon_{\delta}^2 \int d\Omega H_2^{\text{dk}} f_{\text{eq},e} + \alpha \epsilon_{\delta}^2 \int dV \frac{|\nabla_{\perp} A_{1\parallel}|^2}{8\pi}. \end{aligned} \quad (2.70)$$

This expression can be simplified and rewritten in the form of code diagnostics by direct substitution of the expression for the Hamiltonians  $H_0$ ,  $H_1$  given by Eqs. (2.6) and (2.7), while  $H_2$  is given by Eq. (2.9) in the case of the all-orders in FLR polarization density model and by Eq. (2.30) in the case of the long-wavelength approximation. All the terms in the expression for energy except the first one are then rewritten using the corresponding quasineutrality and Ampère equations in their weak form. Here we choose a particular test function  $\hat{\phi}_1 = \phi_1$  and we substitute it in Eqs. (2.10)–(2.13) or, for the case of the long-wavelength approximation, in Eqs. (2.10), (2.11), (2.12), and (2.31). Similarly, the test function  $\hat{A}_{1\parallel} = A_{1\parallel}$  is substituted to the corresponding Ampère equation given by Eq. (2.14). In PIC codes, particles and fields are evaluated in two different ways: particles are advanced continuously along their characteristics while fields are evaluated with finite elements on a fixed grid. To test the quality of the simulation, the contributions to the energy from the particles and from the fields should be computed independently. This is why we are considering the power balance equation, also called the  $E \times B$  transfer equation. The code diagnostics are implemented to verify the following balance equation for  $\mathcal{E}^{\text{EM}} = \mathcal{E}_{\text{F}} + \mathcal{E}_{\text{kin}}$ :

$$0 = \frac{d\mathcal{E}^{\text{EM}}}{dt} \Rightarrow \frac{d\mathcal{E}_{\text{kin}}}{dt} = -\frac{d\mathcal{E}_{\text{F}}}{dt}, \quad (2.71)$$

where the time derivative of the left hand side of this relation can be evaluated through the particle characteristics and the right hand side from the field contributions evaluated on the grid.

From Eq. (2.70), the first term on the r.h.s. is defined as the “kinetic energy”  $\mathcal{E}_{\text{kin}}$ :

$$\mathcal{E}_{\text{kin}} = \sum_{\sigma} \int d\Omega H_0 f_{\sigma} = \sum_{\sigma} \int d\Omega \left( \frac{p_z^2}{2m_{\sigma}} + \mu B \right) f_{\sigma}, \quad (2.72)$$

which depends only on the unperturbed Hamiltonian  $H_0$  and therefore, its time derivative can be evaluated considering only the unperturbed characteristics. The other terms are defined as the “field energy”  $\mathcal{E}_{\text{F}}$ , which can be written, for the case of the  $H_2$  Hamiltonian written in the

LWA, Eq. (2.30), as:

$$\begin{aligned} \mathcal{E}_F = & \epsilon_\delta \sum_{\sigma \neq e} \int d\Omega q_\sigma \left\langle \phi_1 - \alpha A_{1\parallel} \frac{p_z}{m_\sigma} \right\rangle f_\sigma - \epsilon_\delta \int d\Omega e \left( \phi_1 - \alpha A_{1\parallel} \frac{p_z}{m_e} \right) f_e \\ & + \epsilon_\delta^2 \sum_{\sigma \neq e} \int d\Omega f_{\text{eq},\sigma} \left\{ -\frac{m_\sigma c^2}{2B^2} |\nabla_\perp \phi_1|^2 + \alpha \frac{q_\sigma^2}{2m_\sigma} \left[ A_{1\parallel}^2 + \left( \frac{m_\sigma}{q_\sigma} \right)^2 \frac{\mu}{B} A_{1\parallel} \nabla_\perp^2 A_{1\parallel} \right] \right\} \\ & + \epsilon_\delta^2 \alpha \int d\Omega f_{\text{eq},e} \frac{e^2}{2m_e} A_{1\parallel}^2 + \epsilon_\delta^2 \alpha \int dV \frac{|\nabla_\perp A_{1\parallel}|^2}{8\pi}. \end{aligned} \quad (2.73)$$

Using the quasineutrality equation Eqs. (2.10)–(2.12) with the polarization term in the LWA, Eq. (2.31), Ampère equation, Eq. (2.14), and setting  $\hat{\phi}_1 = \phi_1$  and  $\hat{A}_{1\parallel} = A_{1\parallel}$ , we obtain two equivalent expressions for the field energy:

$$\mathcal{E}_F = \epsilon_\delta \frac{1}{2} \sum_{\sigma \neq e} q_\sigma \int d\Omega \left( \langle \phi_1 \rangle - \alpha \frac{p_z}{m_\sigma} \langle A_{1\parallel} \rangle \right) f_\sigma - \epsilon_\delta \frac{1}{2} e \int d\Omega \left( \phi_1 - \alpha \frac{p_z}{m_e} A_{1\parallel} \right) f_e. \quad (2.74)$$

Note that Eq. (2.74) does not depend on the particular choice for the nonlinear Hamiltonian  $H_2$ . Indeed, Eq. (2.74) is also valid for the all order FLR polarization density, Eq. (2.13). This is a direct consequence of the fact that the equations of motion, which are used for rewriting the expression of the energy are obtained from the same field-particle Lagrangian.

Similarly, a second expression for the field energy written in terms of the polarizations and magnetizations and depending on the expression of the nonlinear Hamiltonian  $H_2$  can be obtained. For the full FLR polarization density given by Eq. (2.13), the alternative field energy is given by

$$\begin{aligned} \mathcal{E}_F = & \frac{1}{2} \sum_{\sigma \neq e} \epsilon_\delta \int d\Omega f_{\text{eq},\sigma} \frac{q_\sigma^2}{B} \frac{\partial}{\partial \mu} \langle \tilde{\phi}_1 (\mathbf{X} + \boldsymbol{\rho}_0)^2 \rangle \\ & + \frac{1}{2} \alpha \sum_{\sigma \neq e} \epsilon_\delta^2 \int d\Omega f_{\text{eq},\sigma} \left( \frac{q_\sigma^2}{m_\sigma} A_{1\parallel}^2 + \frac{\mu}{B} A_{1\parallel} \nabla_\perp^2 A_{1\parallel} \right) \\ & + \frac{1}{2} \alpha \epsilon_\delta^2 \int d\Omega f_{\text{eq},e} \frac{e^2}{m_e} A_{1\parallel}^2 + \alpha \epsilon_\delta^2 \int \frac{dV}{8\pi} |\nabla_\perp A_{1\parallel}|^2. \end{aligned} \quad (2.75)$$

For the polarization density in the LWA, Eq. (2.31), the field energy becomes

$$\begin{aligned} \mathcal{E}_F^{\text{LWA}} = & \frac{1}{2} \sum_{\sigma \neq e} \epsilon_\delta \int d\Omega \frac{m_\sigma c^2}{B^2} f_{\text{eq},\sigma} |\nabla_\perp \phi_1|^2 \\ & + \frac{1}{2} \alpha \sum_{\sigma \neq e} \epsilon_\delta^2 \int d\Omega f_{\text{eq},\sigma} \left( \frac{q_\sigma^2}{m_\sigma c^2} A_{1\parallel}^2 + \frac{\mu}{B} A_{1\parallel} \nabla_\perp^2 A_{1\parallel} \right) \\ & + \frac{1}{2} \alpha \epsilon_\delta^2 \int d\Omega f_{\text{eq},e} \frac{e^2}{m_e c^2} A_{1\parallel}^2 + \alpha \epsilon_\delta^2 \int \frac{dV}{8\pi} |\nabla_\perp A_{1\parallel}|^2. \end{aligned} \quad (2.76)$$



For the Padé approximated model, the expression for the field energy is

$$\begin{aligned} \mathcal{E}_F^{\text{Padé}} = & \epsilon_\delta \frac{1}{2} q_i \int d\Omega (1 - \nabla_\perp \cdot \rho_i^2 \nabla_\perp) \langle \phi_1 \rangle f_i - \epsilon_\delta \frac{1}{2} e \int d\Omega (1 - \nabla_\perp \cdot \rho_i^2 \nabla_\perp) \phi_1 f_e \\ & + \epsilon_\delta \sum_\sigma \int d\Omega f_\sigma \left( \frac{p_z^2}{2m_\sigma} + \mu B \right). \end{aligned} \quad (2.77)$$

In the case of the model with adiabatic electrons, the expressions for the conserved energy have to be discussed separately since they are derived from a slightly different variational formulation, which combines a fluid and kinetic formalism. With adiabatic electrons, the corresponding contribution to the energy should be considered as a field term:

$$\begin{aligned} \mathcal{E} = & \sum_{\sigma \neq e} \int d\Omega (H_0 + \epsilon_\delta H_1) f_\sigma + \epsilon_\delta^2 \sum_{\sigma \neq e} \int d\Omega H_2 f_{\text{eq},\sigma} \\ & + \epsilon_\delta \int dV \left[ e n_{e0} \phi_1 + \epsilon_\delta \frac{e}{2T_e} n_{e0} (\phi_1 - \bar{\phi}_1)^2 \right], \end{aligned} \quad (2.78)$$

where the last term is considered as a field term that includes the energy of the adiabatic electrons in the system. Following the general procedure, we substitute the test function  $\hat{\phi}_1 = \phi_1$  into the quasineutrality equation. The field energy is then given by

$$\mathcal{E}_F = \epsilon_\delta \frac{1}{2} \sum_{\sigma \neq e} \int d\Omega q_\sigma f_\sigma \langle \phi_1 \rangle + \epsilon_\delta \frac{1}{2} \int dV e n_{e0} \phi_1 \quad (2.79)$$

and the kinetic part of energy consists of the ion contribution only:

$$\mathcal{E}_{\text{kin}} = \sum_{\sigma \neq e} \int d\Omega f_\sigma H_0 = \sum_{\sigma \neq e} \int d\Omega f_\sigma \left( \frac{m_\sigma v_\parallel^2}{2} + \mu B \right). \quad (2.80)$$

## 2.8 Summary of the chapter

In this chapter, the gyrokinetic model implemented in ORB5 with all the different variants and approximations is presented. To ensure that conserved quantities, such as the energy, are consistently derived, all the approximations to the model are included in the action and the gyrokinetic Vlasov-Maxwell system is derived using a variational approach. In ORB5, the derivation is made up to first order in geometric corrections, i.e.  $\mathcal{O}(\epsilon_B)$ , and up to second order in electromagnetic fluctuations, i.e.  $\mathcal{O}(\epsilon_\delta^2)$ .

All the variants of the physical model, which affect principally the quasineutrality equation, are presented. The polarization density of the gyrokinetic species can be either in integral form, i.e. containing all orders in  $k_\perp \rho_i$ , or approximated using a Padé approximation or the long wavelength approximation. Three electron models are presented: adiabatic, kinetic, and hybrid electrons. The latter, which has been implemented in this work, consists in considering trapped electrons and the flux-surface average of passing electrons as kinetic and the rest of

passing electrons as adiabatic. This correction to the former ORB5 hybrid model has allowed consistent nonlinear simulations including trapped electron modes and at a lower numerical cost than with fully kinetic electrons. An approximation of this model where the flux-surface average is replaced by a  $n = m = 0$  filter is also implemented. The strong flow and collision capabilities of the code are also presented. Finally, the energy conservation associated to the variational formulation is derived for all the variants of the physical model.

## 3 ORB5 numerical implementation, refactoring, and optimization

In this chapter, the numerical aspects of ORB5 are presented. First, the current numerical implementation of the code is shown in Section 3.1. This includes the discretization of the gyrokinetic model presented in Chapter 2 as well as noise control schemes and heat sources. The refactoring and optimization of the ORB5 code are presented in Section 3.2, beginning with a short motivation for the refactoring and a brief description of the optimization techniques that have been used. In Section 3.3, the `PIC_ENGINE` is presented, a test bed that is used to easily implement and test the various algorithms and parallelization schemes before including them into ORB5. Finally, Section 3.4 presents, as an example, the actual optimization of the ORB5 charge deposition step.

### 3.1 Numerical implementation

ORB5 uses a low-noise  $\delta f$  PIC method [69, 70] consisting of separating the full distribution function  $f$  into a prescribed, time-independent background distribution  $f_0$  and a perturbed, time-dependent distribution  $\delta f$  such that only the latter is discretized using markers, or numerical particles, that are used to sample the phase space. Furthermore, the code uses an operator splitting approach which consists of solving first for the collisionless dynamics and then considering the collisions and various sources. The time integration of the collisionless dynamics is made using a 4th-order Runge-Kutta method (RK4). The collisions are treated with a Langevin approach.

This section describes the numerical implementation of the gyrokinetic equations presented in the previous chapter. First, the low-noise  $\delta f$  PIC method as well as the field discretization and solving are presented. Then, the noise reduction techniques, essential to control the noise inherent to the finite sampling of phase space with particles, are described. Finally, the different heat sources, relevant diagnostics, and the parallelization of the code are discussed. In this section, we omit the subscripts  $\sigma$  specifying the species for the sake of simplifying the notation.

### 3.1.1 Discretization of $\delta f$ and equations of motion

In ORB5, phase space is sampled using a set of  $N$  markers that are distributed according to a function  $g(z, t)$  which is discretized as

$$g(z, t) \simeq \sum_{i=1}^N \frac{\delta[z - z_i(t)]}{J(z)}, \quad (3.1)$$

where  $\delta[x]$  is the Dirac distribution,  $z$  is a set of generalized phase-space coordinates,  $z_i(t)$  is the orbit of the  $i$ -th marker in phase space, and  $J(z)$  is the Jacobian associated with the coordinates  $z$ . Even though the choice of the distribution function  $g(z, t)$  is not constrained, we make the convenient choice of using a distribution satisfying

$$\frac{dg}{dt}(z, t) = 0, \quad (3.2)$$

where the  $d/dt$  operator is the collisionless total time derivative defined by the general Vlasov equation, Eq. (2.15). In other words, Eq. (3.2) is ensured by evolving the trajectories  $z_i(t)$  of the markers along the same characteristics as the physical particles, as defined by the collisionless Vlasov equation, Eq. (2.15).

Both background and perturbed distribution functions can be linked to the marker distribution by the weight fields  $W(z, t)$  and  $P(z, t)$ :

$$f_0 = P(z, t)g(z, t) \simeq P(z, t) \sum_{i=1}^N \frac{\delta[z - z_i(t)]}{J(z)} = \sum_{i=1}^N P(z_i(t), t) \frac{\delta[z - z_i(t)]}{J(z)} \quad (3.3)$$

$$= \sum_{i=1}^N p_i(t) \frac{\delta[z - z_i(t)]}{J(z)}, \quad (3.4)$$

$$\delta f = W(z, t)g(z, t) \simeq W(z, t) \sum_{i=1}^N \frac{\delta[z - z_i(t)]}{J(z)} = \sum_{i=1}^N W(z_i(t), t) \frac{\delta[z - z_i(t)]}{J(z)} \quad (3.5)$$

$$= \sum_{i=1}^N w_i(t) \frac{\delta[z - z_i(t)]}{J(z)}, \quad (3.6)$$

where  $p_i(t) = P(z_i(t), t)$  and  $w_i(t) = W(z_i(t), t)$  are the marker weights representing respectively the amplitude of  $f_0$  and  $\delta f$  carried by each marker. The total distribution functions are normalized such that

$$\int f(z, t) J(z) dz = N_{\text{ph}}, \quad (3.7)$$

where  $N_{\text{ph}}$  is the physical number of particles in the system. Note that the coefficient  $N_{\text{ph}}/N$  is hereafter included in the weights such that  $p_i(t) \equiv (N_{\text{ph}}/N) p_i(t)$  and  $w_i(t) \equiv (N_{\text{ph}}/N) w_i(t)$ .

#### 3.1.1.1 Solving for the collisionless dynamics

According to the time splitting approach, the collisionless dynamics is solved first using the standard  $\delta f$  or the direct  $\delta f$  [71] methods. For the standard  $\delta f$  the time evolution of a marker weight  $w_i$  is given by

$$\frac{d}{dt} w_i(t) = \frac{d}{dt} W(z_i(t), t) = \frac{d}{dt} \left[ \frac{\delta f(z, t)}{g(z, t)} \right] = \frac{1}{g(z, t)} \frac{d}{dt} \delta f(z, t) - \frac{\delta f(z, t)}{g(z, t)^2} \frac{d}{dt} g(z, t). \quad (3.8)$$

The last term cancels due to the choice for the evolution of the distribution function  $g(z, t)$ , Eq. (3.2). The total distribution function  $f$  being constant along collisionless trajectories in phase space, the evolution equation of  $w_i$ , Eq. (3.8), can be written as

$$\frac{d}{dt} w_i(t) = -\frac{1}{g(z, t)} \frac{d}{dt} f_0(z_i(t)) = -p_i(t) \frac{1}{f_0(z_i(t))} \frac{d}{dt} f_0(z_i(t)). \quad (3.9)$$

Similarly, an equation for the  $p_i$  weight can also be derived:

$$\frac{d}{dt} p_i(t) = \frac{1}{g(z, t)} \frac{d}{dt} f_0(z_i(t)) = p_i(t) \frac{1}{f_0(z_i(t))} \frac{d}{dt} f_0(z_i(t)). \quad (3.10)$$

In ORB5, both weight equations as well as the numerical particle trajectories are integrated in time using a RK4 method.

On the other hand, the direct  $\delta f$  method exploits the invariance of the total distribution function  $f$  along the nonlinear collisionless trajectories; this property is not ensured in the linear and/or neoclassical limits. It allows one to directly evaluate the weights without numerically solving a differential equation. Adding Eqs. (3.9) and (3.10) leads to

$$\frac{d}{dt} (w_i(t) + p_i(t)) = 0 \implies w_i(t) + p_i(t) = w_i(t_0) + p_i(t_0), \forall t, \quad (3.11)$$

which comes from the invariance of both the  $f$  and  $g$  distribution functions. Furthermore, rewriting Eq. (3.10), we find

$$\frac{d}{dt} \left[ \ln \left( \frac{p_i(t)}{f_0(z_i(t))} \right) \right] = 0 \implies \frac{p_i(t)}{f_0(z_i(t))} = \frac{p_i(t_0)}{f_0(z_i(t_0))}. \quad (3.12)$$

The direct  $\delta f$  algorithm thus consists of first evaluating the  $p_i(t)$  weight according to Eq. (3.12) and then computing the  $w_i(t)$  weight using Eq. (3.11). Note that whatever the  $\delta f$  method used, if the collisionless limit is considered only the  $w_i$  weights are required since the distribution  $g(z, t)$  is invariant along the marker trajectories. Indeed, inserting Eq. (3.12) into Eq. (3.9) leads to

$$\frac{d}{dt} w_i(t) = -\frac{p_i(t_0)}{f_0(z_i(t_0))} \frac{d}{dt} f_0(z_i(t)), \quad (3.13)$$

which, in the limit  $\Delta t \rightarrow 0$ , gives

$$w_i(t + \Delta t) \simeq w_i(t) - \frac{1}{g(z_i(t_0))} [f_0(z_i(t + \Delta t)) - f_0(z_i(t))]. \quad (3.14)$$

Therefore, in the collisionless limit, we do not need to explicitly evolve  $p_i(t)$ .

### 3.1.1.2 Solving for the collisional dynamics

The collision operators are derived assuming linearization with respect to a local Maxwellian distribution. However, ORB5 is typically operated using the canonical background Maxwellian distribution as control variates in order to keep the background distribution in equilibrium in the collisionless gyrokinetic equation. Upon entering the collisions module, the weights  $w_i$  and  $p_i$  are therefore converted to represent the perturbation with respect to a local Maxwellian background distribution, and are reverted when leaving it [66]. In this section,  $f_0$  and  $\delta f$  always refer to these converted distributions, i.e.  $f_0 = f_L$  and  $\delta f = f - f_L$ . At each time step, the collision operators are applied sequentially after the collisionless dynamics.

The electron-ion collision operator and the test-particle component of the intra-species collision operators are applied using a Langevin approach. In the gyrokinetic framework, this corresponds to applying random “kicks” in velocity space.

For electrons colliding on ions, Eq. (2.69) is reformulated in a spherical coordinate system in velocity space with radius  $v$ , polar angle  $\theta$ , and azimuthal angle  $\alpha$  in which the incoming electron’s velocity corresponds to  $\theta = 0$ . Then, coming back to the ORB5 set of coordinates, the outgoing velocity of the electron is given by

$$v_{||,out} = v_{in} \left[ -\sin(\Delta\theta) \sin(\alpha_{out}) \sqrt{1 - \xi_{in}^2} + \xi_{in} \cos(\Delta\theta) \right], \quad (3.15)$$

$$v_{\perp,out}^2 = v_{in}^2 - v_{||,out}^2, \quad (3.16)$$

where  $\Delta\theta = 2R\sqrt{v_{ei}(v)\Delta t}$ ,  $R$  being a random sample of a PDF with mean 0 and variance 1, and  $\alpha_{out}$  is a random sample of a uniform distribution between 0 and  $2\pi$ . Note that the kinetic energy is exactly conserved by this procedure as in the original model.

Applying a similar approach for the “test-particle” self-collisions, Eqs. (2.64)–(2.67) yield the following outgoing particle trajectory:

$$v_{||,out} = \frac{1}{v_{in}} \left[ -\Delta v_y v_{\perp,in} + (v_{in} + \Delta v_z) v_{||,in} \right], \quad (3.17)$$

$$v_{\perp,out}^2 = \Delta v_x^2 + \frac{1}{v_{in}^2} \left[ \Delta v_y v_{||,in} + (v_{in} + \Delta v_z) v_{\perp,in} \right]^2, \quad (3.18)$$

where  $\Delta v_x$ ,  $\Delta v_y$ , and  $\Delta v_z$  are the particle’s change in velocity, with the unit vector  $\hat{z}$  in the

direction of the incoming particle's velocity. These kicks are described by

$$\Delta v_x = v_{\text{th}} \sqrt{\frac{K(v_{\text{in}}) \bar{v} \Delta t}{2}} R_1, \quad (3.19)$$

$$\Delta v_y = v_{\text{th}} \sqrt{\frac{K(v_{\text{in}}) \bar{v} \Delta t}{2}} R_2, \quad (3.20)$$

$$\Delta v_z = -H(v_{\text{in}}) v_{\text{in}} \bar{v} \Delta t + v_{\text{th}} \sqrt{H(v_{\text{in}}) \bar{v} \Delta t} R_3, \quad (3.21)$$

where  $R_1$ ,  $R_2$ , and  $R_3$  are again independent random numbers sampled from a PDF with mean 0 and variance 1. The marker's magnetic moment  $\mu$  is then updated accordingly.

It can be shown [66] that the evolution of the marker weight  $w_i$  due to collisions can be expressed as

$$\frac{d}{dt} w_i(t) = -p_i(t) \frac{C[\delta f, f_L]}{f_L} \Big|_{[z_i(t), t]}, \quad (3.22)$$

where  $z_i(t)$  is the marker position after the “test-particle” kicks. At this point, the “background-reaction” operator, Eq. (2.68), is slightly corrected as follows:

$$\Delta w_i(t) = -\frac{p_i}{n_\alpha} \left[ \left(1 - 3\sqrt{\pi} G(x)\right) \Delta N_\alpha + 6\sqrt{\pi} H(x) \frac{\Delta P_{||, \alpha} v_{||\text{out}, r}}{v_{\text{th}, \alpha}^2} + \sqrt{\pi} G(x) \frac{\Delta E_\alpha}{v_{\text{th}, \alpha}^2} \right], \quad (3.23)$$

where  $\Delta N_\alpha$ ,  $\Delta P_{||, \alpha}$ , and  $\Delta E_\alpha$  are determined by imposing that the combined effect of applying the test particle collision operator and the background reaction term conserves mass, parallel momentum, and kinetic energy within each configuration space bin  $\alpha$  to machine precision.

#### 3.1.1.3 Adding the dynamics related to the source terms

In the presence of sources, Section 3.1.4, or noise reduction schemes such as the Krook operator or the coarse graining, Section 3.1.3, the marker dynamics represented by

$$\frac{d}{dt} w_i(t) = S(t), \quad (3.24)$$

where  $S$  is a source term, is evolved using a first order finite difference approximation:

$$w_i(t^{n+1}) \approx w_i(t^n) + \Delta t S(t^n), \quad (3.25)$$

where  $n$  is the current timestep.

#### 3.1.1.4 Particle loading

At the beginning of a simulation, the markers are loaded in phase space using a Halton-Hammersley sequence [72, 73] and according to the distribution function

$g(z, t = 0) = f_{\text{rad}}(s)f_v(v_{\parallel}, v_{\perp})$ , where  $f_{\text{rad}}(s)$  and  $f_v(v_{\parallel}, v_{\perp})$  define respectively the radial and velocity sampling distributions, and  $s = \sqrt{\psi/\psi_{\text{edge}}}$  is a normalized radial coordinate. In ORB5, the spatial sampling is defined by the *specified loading* distribution function  $f_{\text{rad}}(s) = 1 - f_g + f_g \exp[(s - s_0)^2 / \Delta s^2]$ , where  $f_g \in [0, 1]$ ,  $s_0$ , and  $\Delta s$  are input parameters. In velocity space  $(v_{\parallel}, v_{\perp})$ , the distribution is either uniform in the half-plane  $(v_{\parallel}, v_{\perp} > 0)$  within  $v = \sqrt{v_{\parallel}^2 + v_{\perp}^2} \leq \kappa_v v_{\text{th},\sigma}$  with

$$f_v(v_{\parallel}, v_{\perp}) = \frac{1}{v_{\perp} (\pi \kappa_v v_{\text{th}}(s))^2}, \quad (3.26)$$

or uniform in 3D velocity space within a velocity sphere of radius  $\kappa_v v_{\text{th},\sigma}$  with

$$f_v(v_{\parallel}, v_{\perp}) = \frac{1}{\frac{4}{3}\pi \kappa_v^3 v_{\text{th}}^3(s)}, \quad (3.27)$$

where  $\kappa_v$  is an input parameter usually set at  $\kappa_v = 5$  and which allows one to define a cut-off in velocity sampling.

For the marker weight initialization, two main schemes are implemented. The first option is a *white noise initialization* defined by

$$w_i(t_0) = A(2Q_i - 1)p_i(t_0), \quad (3.28)$$

where  $Q_i$  is a quasi-random number in  $[0, 1]$  given by the  $i$ -th term of a van der Corput sequence [74] and  $A$  the maximum amplitude given as an input parameter, typically of the order of  $A \sim 10^{-3} - 10^{-5}$ . The disadvantage of this scheme is that the initial density perturbation is inversely proportional to the number of particles and the time until physical modes emerge from the initial state is roughly proportional to the number of particles. To accelerate the mode development, the *mode initialization* can be used. It consists of initializing a number of Fourier modes:

$$w_i(t_0) = \frac{A_0 p_i(t_0)}{(m_2 - m_1 + 1)(n_2 - n_1 + 1)} \left| \frac{T(s_0)}{\nabla T(s_0)} \right| \left| \frac{T(s_i(t_0))}{\nabla T(s_i(t_0))} \right| \sum_{m=m_1}^{m_2} \sum_{n=n_1}^{n_2} \cos(m\theta_i^*(t_0) - n\varphi_i(t_0)), \quad (3.29)$$

where  $A_0$ ,  $n_1$ ,  $n_2$ ,  $m_1$ ,  $m_2$  are input parameters. Typically, for linear simulations of microinstabilities with a toroidal mode number  $n_0$ , it is convenient to use  $n_1 = n_2 = n_0$  and  $m_1 = m_2 = -n_0 q(s_0)$  as modes are almost aligned with the magnetic field lines. Finally, whatever initialization is used, the initial average value of the weights is set to zero to avoid breaking quasi-neutrality ensured by the background distributions:

$$\frac{1}{N} \sum_i^N w_i(t_0) = 0. \quad (3.30)$$



As mentioned in section 2.1 the markers are pushed in toroidal magnetic coordinates  $(s, \theta^*, \varphi)$ . To avoid the singularity that would appear in the equations of motion at the magnetic axis, the coordinate system is changed to  $(\xi, \eta, \varphi) = (s \cos \theta^*, s \sin \theta^*, \varphi)$  near the axis. All equilibrium quantities for both *ad-hoc* and MHD equilibria are loaded on an  $(R, Z)$  grid, where  $R$  and  $Z$  are cylindrical coordinates, and are linearly interpolated to an  $(s, \theta^*)$  grid. Markers that exit the radial domain at  $s > 1$  are reflected back into the plasma at a position which conserves toroidal momentum, the particle energy, and the magnetic moment but with a null weight to avoid unphysical accumulation of perturbed density at the radial edge.

#### 3.1.2 Quasineutrality and Ampère equations

In ORB5 the quasineutrality and Ampère equations are solved using the Galerkin method considering linear, quadratic, or cubic B-splines finite elements defined on a  $(N_s, N_{\theta^*}, N_\varphi)$  grid. The perturbed fields  $\phi$  and  $A_\parallel$  hereafter noted  $\Psi = \{\phi, A_\parallel\}$  are thus discretized as follows:

$$\Psi(\mathbf{X}, t) = \sum_{\mu} \Psi_{\mu}(t) \Lambda_{\mu}(\mathbf{X}), \quad (3.31)$$

where  $\{\Psi_{\mu}(t)\}$  are the field coefficients and  $\{\Lambda_{\mu}(\mathbf{X})\}$  are a tensor product of 1D B-splines of degree  $p = \{1, 2, 3\}$ ,  $\Lambda_{\mu}(\mathbf{X}) = \Lambda_j^p(s) \Lambda_k^p(\theta^*) \Lambda_l^p(\varphi)$ , with  $\mu = (j, k, l)$ .

Using the decomposition defined in Eqs. (3.5) and (3.31), and setting the test functions  $\hat{\phi}_1 = \Lambda_v(\mathbf{X})$ ,  $v = (j', k', l')$  of the variational forms of the quasineutrality and Ampère equations, Eqs. (2.10) and (2.14), leads to a linear system of the form

$$\sum_{\mu} A_{\mu v} \Psi_{\mu}(t) = b_v(t), \quad (3.32)$$

where  $A_{\mu v}$  and  $b_v$  are respectively a real symmetric positive-definite square matrix and a column vector that are defined by the physical models used in the quasineutrality and Ampère equations. Due to the finite support of the B-splines, the matrix  $A_{\mu v}$  is usually a block diagonal matrix composed of banded submatrices. Note that in the case of the quasi-neutrality valid to all orders, the matrix  $A$  in general becomes a full matrix. As illustration, we show here the linear system for the case of a single ion species plasma in the limit of adiabatic electrons with the long wavelength approximation for the ion polarization density:

$$A_{\mu v}^{\text{LWA, adiab}} = \int \frac{en_0(\psi)}{T_e(\psi)} (\Lambda_{\mu}(\mathbf{X}) \Lambda_v(\mathbf{X}) - \bar{\Lambda}_{\mu}(s) \bar{\Lambda}_v(s)) dV \quad (3.33)$$

$$+ \int \frac{n_0(\psi) m_i}{B^2} \nabla_{\perp} \Lambda_{\mu}(\mathbf{X}) \cdot \nabla_{\perp} \Lambda_v(\mathbf{X}) dV, \quad (3.34)$$

$$b_v(t) = \sum_{p=1}^N \frac{w_p(t)}{2\pi} \int_0^{2\pi} d\alpha \Lambda_v(\mathbf{X}_p + \boldsymbol{\rho}_{L,p}(\alpha)), \quad (3.35)$$

where  $\boldsymbol{\rho}_{L,p}$  is the Larmor radius of a particle  $p$ . Here, the gradient perpendicular to the mag-

netic field is approximated by the gradient in the poloidal plane, i.e.  $\nabla_{\perp} \simeq \nabla_{\text{pol}} = \nabla s \frac{\partial}{\partial s} + \nabla \theta^* \frac{\partial}{\partial \theta^*}$ . Note that the expression for  $b_v(t)$ , Eq. (3.35), is independent of the choice of coordinates. This is due to the particle representation of  $\delta f$ , Eq. (3.5), and the Galerkin finite element method based on the variational form of the field equations, Eqs (2.10) and (2.14). This is very convenient practically as the charge deposition is totally transparent to the choice of the coordinates system, which greatly simplifies the numerical implementation. A more complete description of the discretized Poisson equation for arbitrary wavelengths can be found in [54].

In ORB5, the linear system of equations, Eq. (3.32), is represented in Fourier space to obtain an equivalent system of equations [8] that is solved using the FFTW library [75] and a direct solver from the LAPACK library [76]. This Fourier representation of the fields is convenient because it allows one to decouple the toroidal mode numbers  $n$ , provided that the field equations are linear and that the unperturbed system is axisymmetric. Furthermore, the modes of interest, e.g. drift-wave type and Alfvén waves, are typically almost aligned with the magnetic field lines and can thus be described with just a small set of Fourier coefficients, which greatly decreases the numerical cost as compared to solving the system in direct space. Finally, in the case of moderate shaping, there is a limited number of couplings of poloidal modes  $m$ . Denoting with  $\mathcal{F}$  the double discrete Fourier transform on both poloidal and toroidal directions, the linear system of equations (3.32) becomes

$$\sum_{\mu} \mathcal{F} A_{\mu\nu} \mathcal{F}^{-1} \mathcal{F} \Psi_{\mu} = \mathcal{F} b_{\nu}, \quad (3.36)$$

$$\mathcal{F} A_{\mu\nu} \mathcal{F}^{-1} = \hat{A}_{(j,j')}^{(n,m),(n',m')}, \quad (3.37)$$

$$\mathcal{F} \Psi_{\mu} = \hat{\Psi}_j^{n,m}, \quad (3.38)$$

$$\mathcal{F} b_{\nu} = \hat{b}_{j'}^{n,m'}, \quad (3.39)$$

where  $n$  and  $m$  are respectively the toroidal and poloidal Fourier mode numbers.

Due to the axisymmetry of the system and because the Poisson-Ampère system of equation is linear, the toroidal direction can be decoupled from the others with  $n = n'$  [77]:

$$\sum_j \sum_m \hat{A}_{(j,j')}^{(n,m),(n,m')} \hat{\Psi}_j^{n,m} = \frac{\hat{b}_{j'}^{n,m'}}{M^{n,p}} \quad \forall n, \quad (3.40)$$

where the matrix  $M^{n,p}$  is defined by

$$M^{n,p} = \sum_{l'=1}^{N_{\varphi}} \int d\varphi \Lambda_{l'}^p(\varphi) \Lambda_l^p(\varphi) \exp \left[ \frac{2\pi i}{N_{\varphi}} n(l' - l) \right], \quad (3.41)$$

and can be computed analytically for any B-spline of order  $p$ :

$$M^{n,1} = \frac{2\pi}{N_\varphi} \left[ \frac{2}{3} + \frac{1}{3} \cos\left(\frac{2\pi}{N_\varphi} n\right) \right], \quad (3.42)$$

$$M^{n,2} = \frac{2\pi}{N_\varphi} \left[ \frac{8}{15} + \frac{13}{30} \cos\left(\frac{2\pi}{N_\varphi} n\right) + \frac{1}{30} \cos^2\left(\frac{2\pi}{N_\varphi} n\right) \right], \quad (3.43)$$

$$M^{n,3} = \frac{2\pi}{N_\varphi} \left[ \frac{136}{315} + \frac{33}{70} \cos\left(\frac{2\pi}{N_\varphi} n\right) + \frac{2}{21} \cos^2\left(\frac{2\pi}{N_\varphi} n\right) + \frac{1}{630} \cos^3\left(\frac{2\pi}{N_\varphi} n\right) \right]. \quad (3.44)$$

This toroidal decoupling is very interesting numerically because the different toroidal mode numbers  $n$  can be solved for separately.

The matrix  $A_{\mu\nu}$  and the right-hand side  $b_\nu$  are modified such that the following boundary conditions are used. At the magnetic axis the *unicity condition* is applied,  $\Psi(s=0, \theta^*, \varphi, t) = \Psi(s=0, \theta^*=0, \varphi, t), \forall \theta^*$ . At the outer radial edge, Dirichlet boundary conditions are applied,  $\Psi(s=1, \theta^*, \varphi, t) = 0$ . Note that ORB5 can also be run in an annulus, i.e.  $s \in [s_{\min}, s_{\max}]$ , with  $s_{\min} > 0$  and  $s_{\max} < 1$ , in which case Dirichlet boundary conditions are applied on both edges. For the quasineutrality equation with polarization density valid to all orders in  $k_\perp \rho$ , the equation is integral and no Dirichlet boundary conditions need to be applied [54].

### 3.1.2.1 Gyroaveraging

For all gyroaveraging operations, the plane of the Larmor ring is approximated to lie in the poloidal plane. The number of gyropoints can be either fixed or determined by an adaptive scheme, in which a fixed number of gyropoints is used for all the particles having a Larmor radius smaller than or equal to the thermal Larmor radius and the number of points increases linearly for larger Larmor radii. Usually, a fixed number of 4 gyropoints is sufficient for perturbations up to  $k_\perp \rho_L \sim 1$ . However, using the adaptive scheme reduces the noise as it acts as a Bessel filter averaging out shorter wavelength fluctuations [78].

In magnetic coordinates, the positions of the gyropoints are parametrized using the gyroangle  $\alpha$ :

$$\mathbf{x}(\alpha) = \mathbf{X} + \boldsymbol{\rho}(\alpha) = \mathbf{X} + \rho \frac{\nabla s}{|\nabla s|} \cos \alpha + \rho \frac{\mathbf{b} \times \nabla s}{|\mathbf{b} \times \nabla s|} \sin \alpha, \quad (3.45)$$

where  $\mathbf{X}$  is the position of the guiding center.

The gradient of gyroaveraged electric potential,  $\nabla \langle \phi_1 \rangle$ , is approximated as

$$\nabla_{\mathbf{X}} \langle \phi_1 \rangle \simeq \langle \nabla \phi_1 \rangle = \frac{1}{2\pi} \oint_0^{2\pi} \nabla_{\mathbf{X}} \phi_1(\mathbf{X} + \boldsymbol{\rho}) d\alpha, \quad (3.46)$$

where the subscript  $\mathbf{X}$  stands for the gradient with respect to the gyrocenter coordinates and  $\alpha$  is the gyroangle. We define a new set of coordinates  $\bar{\mathbf{X}} = (\bar{R}, \bar{Z}) = (R + \rho \cos \alpha, Z + \rho \sin \alpha) = \mathbf{X} + \boldsymbol{\rho}$  representing the position of the particle on the gyro-ring in the poloidal plane where  $\bar{R}$  is

in the direction of the major axis and  $\bar{Z}$  is in the direction of the vertical axis. This coordinate system is the projection of Eq. (3.45) on the  $(R, Z)$  plane of a cylindrical coordinate system. Using the chain rule, the term  $\nabla_X \phi_1$  from Eq. (3.45) can be written as

$$\nabla_X \phi_1(\mathbf{X} + \boldsymbol{\rho}) = \nabla_{\bar{X}} \phi_1 - \frac{\rho}{2} \left( \frac{\partial \phi_1}{\partial \bar{R}} \cos \alpha + \frac{\partial \phi_1}{\partial \bar{Z}} \sin \alpha \right) \frac{\nabla_X B}{B}. \quad (3.47)$$

A similar procedure is done for  $\nabla \langle A_{1\parallel} \rangle$ . In ORB5, Eq. (3.47) can be either directly evaluated as in [58] or approximated by neglecting the second term, leading to  $\nabla_X \phi_1 \approx \nabla_{\bar{R}} \phi_1$ .

### 3.1.2.2 Fourier filter

Typical physical modes of interest, e.g. drift waves and low-frequency Alfvén waves, are mainly aligned with the magnetic field lines, i.e. they have  $m \approx nq(s)$ . Due to this strong anisotropy, only a limited set of  $(n, m)$  Fourier coefficients is required to describe the physical modes as the amplitude of the  $(n, m)$  coefficients rapidly decreases away from  $m = nq(s)$  [79]. It is then beneficial to filter out all the non physically relevant Fourier modes in order to reduce the sampling noise and maximize the timestep size. The filter is applied on the Fourier coefficients of the perturbed density and current to remove all the non physical Fourier modes introduced by the charge and current depositions:

$$\tilde{b}_{(j,k,l)} = \sum_{n,m} f_{j,n,m} \hat{b}_j^{n,m} e^{im\theta_k^*} e^{in\varphi_l}, \quad (3.48)$$

where  $f_{j,n,m}$  is the Fourier filter that in general depends on the radius (radial index  $j$ ), and the poloidal and toroidal mode numbers  $m$  and  $n$ , respectively.

Two different filters are used successively. First, a *rectangular filter*, which is the simplest one, is applied such that all the modes outside of the window  $[n_{\min}, n_{\max}] \times [m_{\min}, m_{\max}]$  specified in input are filtered out. This filter is not sufficient as it keeps modes with  $k_{\parallel}/k_{\perp}$  much bigger than  $\rho^*$ , which is inconsistent with the gyrokinetic ordering [79]. Since the modes of interest are mainly aligned with the magnetic field, i.e. they satisfy  $k_{\parallel} \rho_i \simeq [m - nq(s)] [q(s)R]^{-1} \rho_i = \mathcal{O}(\rho^*)$ , a second surface-dependent *field-aligned filter* is applied. It consists of retaining only  $m$  modes close to  $nq(s)$ , i.e.  $m \in [nq(s) - \Delta m, nq(s) + \Delta m]$ , where  $\Delta m$  is an input parameter specifying the width of the filter. With this field-aligned filter, the maximum value of  $|k_{\parallel}|$  represented is  $|k_{\parallel}|_{\max} = |\Delta m|/qR$ . Since  $|k_{\parallel}|_{\max} \rho_i$  scales with  $\rho^*$ , the value of  $\Delta m$  required to describe all physically relevant modes is invariant with the system size. Typically, a value of  $\Delta m = 5$  is sufficient [8]. In summary, for each mode  $n \in [n_{\min}, n_{\max}]$  only the modes  $m \in [m_{\min}, m_{\max}] \cap [-nq(s) \pm \Delta m]$  are retained.

### 3.1.3 Noise control techniques

Due to the finite number of markers used to sample phase space, PIC simulations are subject to sampling errors, referred to as numerical noise, which in fact increases in a nonlinear

simulation, thus deteriorating the signal quality and forbidding long simulations without noise control techniques. All the difficulty of such noise-reducing schemes is to actually control the weight growth without creating severe non-physical artifacts. In this section we present the different noise control schemes implemented in ORB5.

#### 3.1.3.1 Krook operator

The Krook operator implemented in ORB5 [80] is a source term which weakly damps the non axisymmetric fluctuations without significantly affecting the zonal flows. This is done via a correction term that also allows one to conserve various moments by projecting out some components of the source. The Krook noise-control term,  $S_K^{\text{NC}}$ , is composed of a relaxation term and its correction  $S_K^{\text{corr}}$ :

$$S_K^{\text{NC}} = -\gamma_K \delta f + S_K^{\text{corr}}, \quad (3.49)$$

$$S_K^{\text{corr}} = \sum_{i=1}^{N_{\text{mom}}} g_i(s) M_i f_0, \quad (3.50)$$

where  $\gamma_K$  is the Krook damping rate. The correction term is a sum over the  $N_{\text{mom}}$  moments  $M_i$  one wishes to conserve on a flux-surface average. Typically, in ORB5, the moments that can be conserved are the density, parallel velocity, zonal flows, and kinetic energy. They are respectively defined by  $M_i = \{1, v_{\parallel}, v_{\parallel}/B - \overline{(v_{\parallel}/B)}, \mathcal{E}_K\}$ , where the tilde represents the bounce average and  $\mathcal{E}_K$  is the kinetic energy of a particle. The coefficients  $g_i(s)$  are defined such that there is no contribution of the source to a given moment  $M_j$ , i.e.

$$\overline{\int dW M_j S_K^{\text{NC}}} = 0, \quad \forall j, \quad (3.51)$$

where the bar notation stands for the flux-surface average. Injecting the definition of the Krook source term, Eq. (3.49), in Eq. (3.51) leads to a linear system of equations that is solved at each time step and radial position to find the coefficients  $g_i(s)$ :

$$\sum_i^{N_{\text{mom}}} S_{ij}(s) g_i(s) = \delta S_j(s), \quad (3.52)$$

with

$$S_{ij}(s) = \overline{\int dW M_j(\mathbf{X}, v_{\parallel}, \mu) M_i(\mathbf{X}, v_{\parallel}, \mu) f_0(\mathbf{X}, v_{\parallel}, \mu)}, \quad (3.53)$$

$$\delta S_j(s) = \gamma_K \overline{\int dW \delta f(\mathbf{X}, v_{\parallel}, \mu) M_j(\mathbf{X}, v_{\parallel}, \mu)}. \quad (3.54)$$

Note that the flux-surface average is numerically represented by a binning of the markers in the radial direction. This implies that the conservation is ensured only on average across each radial bin.

As already mentioned, the noise control should not affect significantly the turbulence. To this end, values of the order of one tenth of the maximum linear growth rate are usually used for the Krook damping rate  $\gamma_K$ . In this way, the linear phase is not substantially modified and a high signal-to-noise ratio can be obtained. On the other hand, applying this noise control technique is not appropriate when considering collisions as the damping rate  $\gamma_K$  is comparable to typical values of the collision frequency thus interfering with the effect of collisions.

By construction, the Krook operator damps the fluctuations to restore the full distribution function to its initial state. If the kinetic energy is not conserved while conserving the other moments, it allows one to run temperature gradient-driven simulations by acting as an auto-regulated heat source while allowing for the unconstrained evolution of the density and flow profiles.

### 3.1.3.2 Coarse-graining

Coarse-graining [68, 81] is an additional noise-control method implemented in ORB5 [65] to reduce the problems of weight-spreading and filamentation of the distribution function, that lead to large mean squared particle weights. The idea is essentially to dissipate fine-scale structures of the distribution function in phase space, as represented by the marker weights. This is an improvement compared to the Krook operator, which only preserves certain moments of the zonal distribution function but otherwise somewhat indiscriminately damps the whole distribution function; the Krook operator, for example, was found to be unsuitable for neoclassical studies for which collisional effects are clearly essential.

In a Eulerian code, phase space dissipation is often implemented as a hyper-viscosity operator on the grid in the five spatial and velocity directions. As the PIC approach does not involve a phase-space grid, we need an alternative method to smooth the weights of nearby markers.

Computationally, the method consists of binning the particles in phase space cells which are field-aligned in configuration space, and then reducing the deviation of particle weights in the grid cell from their average value. To avoid smoothing structures at the turbulence scale too strongly, the bins must be small compared to typical length and velocity scales; on the other hand the bins need to frequently contain more than one marker for this procedure to be effective. Elongated field-aligned bins are used because the distribution function varies much

more rapidly perpendicular to the field line than along it.

The bins are volumes in a block-structured Cartesian mesh in coordinates  $(s, z, \theta^*, \lambda, \epsilon)$ , with the number of bins uniform in each direction, except that the number of bins in the  $\theta^*$  direction is proportional to  $s$ , so that the spatial volume of bins is roughly constant. The coordinate  $\epsilon$  is the particle kinetic energy,  $\lambda$  is the pitch angle, and  $z$  is a field-line label that is computed as

$$z = \varphi - q(s)[\theta^* - \theta_0^*(\theta^*)], \quad (3.55)$$

with  $\theta_0^*$  the center of the bin in the  $\theta^*$  direction:

$$\theta_0^*(\theta^*) = \left( \left\lfloor \frac{\theta^* + \pi}{\Delta\theta^*} \right\rfloor + \frac{1}{2} \right) \Delta\theta^* - \pi, \quad (3.56)$$

where  $\lfloor \bullet \rfloor$  is the floor function and  $\Delta\theta^*$  is the width of a bin in the  $\theta^*$  direction. With this choice of  $z$  we have a field-aligned bin, but we also have  $z \sim \varphi$  if there are many bins in the  $\theta^*$  direction because  $\theta_0^*(\theta^*)$  tends to  $\theta^*$  for an infinite number of bins. This is useful because the domain decomposition—discussed in detail in Section 3.1.5—means that markers on a single processor have a small range of values of  $\varphi$ . Thus, the first step in the binning process is to distribute the markers according to  $z$  and move them to this alternative domain decomposition. In the  $z$  decomposition, coarse-graining is local to each domain, so we do not need to communicate quantities on the 5D coarse-graining mesh.

The number of bins in the  $s$  and  $z$  directions are the field mesh quantities  $N_s$  and  $N_\varphi$  respectively and the number of  $\theta^*$ , energy and pitch-angle bins are specified as input parameters. To avoid excessive damping of zonal flows, around 32 bins are needed in each of the energy and pitch-angle directions. Often 16 bins in the  $\theta^*$  direction are sufficient to avoid excessive damping of parallel structures.

The smoothing operation changes the particle weight  $w$  by an amount  $\mathcal{N} \Delta t \gamma_{\text{cg}} (\bar{w} - w)$ , where  $\bar{w}$  is the average particle weight in the bin,  $\mathcal{N}$  is the number of timesteps (of length  $\Delta t$ ) between coarse-graining operations, and  $\gamma_{\text{cg}}$  is a parameter controlling the coarse-graining rate. In the limit of large number of markers, this leads to a damping of fine-scale structures in the distribution function with a rate  $\gamma_{\text{cg}}$ . Note, however, that in practice typical runs have 0.1 markers per bin, so that the effective coarse-graining rate is lower than  $\gamma_{\text{cg}}$  by a factor of 10.

#### 3.1.3.3 Quad-tree particle-weight smoothing

The grid-based coarse-graining procedure has the possible drawback of being inaccurate if the grid of phase-space bins is too fine so that the local statistics is not good enough, or being very diffusive if the grid is too coarse. An alternative procedure, gridless in velocity space and more probabilistic in nature, has been proposed in [82] and implemented in ORB5. It consists of pairing neighbouring markers and replacing their weights by an average, weighted by a function of their distance in velocity space. The way of computing the distance and the

weight has an influence on the diffusivity of the method, for this reason we use a procedure for pairing only particles which are close enough. Since the gyrokinetic velocity space is 2D, the pairing procedure is done using a quad tree algorithm: first, the particles are binned in the configuration space and then, a quad tree procedure is applied to define regions in velocity space within which particles will be paired. This works by subdividing recursively the 2D velocity space in four sub-boxes until the number of particles in a sub-box is smaller than a given value set as an input parameter. At this point, the particles within a sub-box are randomly paired and their weight is changed according to the following procedure: for a pair of two markers with weights  $w_1$  and  $w_2$  and velocities  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , the new weights are given by

$$w_1^{\text{new}} = (1 - \Gamma) w_1^{\text{old}} + \Gamma \bar{w}, \quad (3.57)$$

$$w_2^{\text{new}} = (1 - \Gamma) w_2^{\text{old}} + \Gamma \bar{w}, \quad (3.58)$$

with

$$\Gamma = e^{-\frac{(\mathbf{v}_1^x - \mathbf{v}_2^x)^2 + (\mathbf{v}_1^y - \mathbf{v}_2^y)^2}{h_v^2}}, \quad (3.59)$$

$$\bar{w} = \frac{w_1^{\text{old}} + w_2^{\text{old}}}{2}, \quad (3.60)$$

where the  $x$  and  $y$  superscripts are used to identify the two dimensions of the velocity and the  $h_v$  parameter defines how strong is the smoothing procedure with respect to the distance separating the pair of markers in velocity space. Note that, by construction, the smoothing operation conserves the total weight, ensuring density conservation. By picking different pairs of particles within the same quad tree sub-box, the smoothing operation can be applied several times per timestep. Typically, one smoothing step is done at every timestep.

#### 3.1.3.4 Enhanced control variates

The ORB5 code solves the uncoupled electromagnetic gyrokinetic equations in the  $p_z$ -formulation, Eqs. (2.10) and (2.14) and therefore includes the cancellation problem [83] which, if untreated, in practice limits the electromagnetic simulations to very-low-beta cases,  $\beta < \sqrt{m_e/m_i}$ , where  $\beta$  is the stored kinetic energy divided by the magnetic field energy. Different methods mitigating this problem have been developed for the particle-in-cell framework in Refs. [84, 85, 86, 87, 58] and for the Eulerian approach in Ref. [20]. In ORB5 the cancellation problem is treated [88] using the enhanced control variates scheme presented in [87, 58]. A further development of the mitigation schemes is given in Refs. [89, 90], the so-called pullback mitigation based on the mixed-variable formulation [91] of the gyrokinetic theory, has also been implemented in ORB5 [92]. Mitigation of the cancellation problem has made possible the ORB5 electromagnetic simulations described in Refs. [93, 94].

The enhanced control variates approach is based on the decomposition of the distribution function into the so-called adiabatic and nonadiabatic parts introduced in [95] while



constructing a perturbative procedure for the solution to the gyrokinetic Vlasov equation. The same decomposition can be extracted via the pull-back transformation between the particle distribution function and the reduced gyrokinetic distribution [96]. This transformation requires that the equilibrium distribution commutes with the background dynamics, i.e.  $\{f_{\text{eq}}, H_0\} = 0$ . Furthermore, in ORB5 the distribution function is assumed to be a canonical Maxwellian, i.e. satisfying

$$\frac{\partial f_{\text{eq},\sigma}}{\partial H_0} = \frac{f_{\text{eq},\sigma}}{T_\sigma}, \quad (3.61)$$

where the temperature is defined as

$$T_\sigma = \frac{1}{n_{0\sigma}} \int dW \frac{p_z^2}{2m_\sigma} f_{\text{eq},\sigma}. \quad (3.62)$$

In the enhanced-control-variate scheme, the perturbed distribution function is split according to

$$f_{\sigma,1} = G_\sigma - \frac{f_{\text{eq},\sigma}}{T_\sigma} H_1, \quad (3.63)$$

where the first and second terms are respectively the nonadiabatic and adiabatic parts.

The cancellation problem is related to the coexistence of very large and very small quantities in the variational form of the Ampère equation (2.14). To illustrate the problem, let us consider a case with only one ion species and rewrite Eq. (2.14). Note that the second and third integrals of Eq. (2.14) are the projections  $\mathcal{J}_{\sigma,\parallel}$  of the ion and electron currents onto the basis function  $\hat{A}_{1\parallel}$ :

$$\langle \mathcal{J}_{i,\parallel} \rangle \equiv \int dV \langle j_{i,\parallel} \rangle \hat{A}_{1\parallel} = \int d\Omega f_i \frac{q_i p_z}{m_i c} \langle \hat{A}_{1\parallel} \rangle, \quad (3.64)$$

$$\mathcal{J}_{e,\parallel} \equiv \int dV j_{e,\parallel} \hat{A}_{1\parallel} = \int d\Omega f_e \frac{e p_z}{m_e c} \hat{A}_{1\parallel}. \quad (3.65)$$

Then, the fourth integral and the first term of the fifth integral of Eq. (2.14) are the so-called skin terms and can be written as

$$\int d\Omega f_{\text{eq},\sigma} \left( \frac{4\pi q_\sigma^2}{m_\sigma c^2} A_{1\parallel} \hat{A}_{1\parallel} \right) = \frac{\beta_\sigma}{\rho_{\text{th},\sigma}^2} \int dV A_{1\parallel} \hat{A}_{1\parallel}, \quad (3.66)$$

where one defines  $\beta_\sigma = 4\pi n_\sigma T_\sigma / B^2$  and  $\rho_{\text{th},\sigma}$  is the thermal Larmor radius. Finally, the

remaining terms of Eq. (2.14) are combined to form

$$\begin{aligned} & \int dV \nabla_{\perp} A_{1\parallel} \cdot \nabla_{\perp} \hat{A}_{1\parallel} + \int d\Omega f_{\text{eq},i} \frac{2\pi\mu}{B} [A_{1\parallel} \nabla_{\perp}^2 \hat{A}_{1\parallel} + \hat{A}_{1\parallel} \nabla_{\perp}^2 A_{1\parallel}] \\ &= \int dV \nabla_{\perp} \cdot [(1 - \beta_i) \nabla_{\perp} A_{1\parallel} \hat{A}_{1\parallel}], \end{aligned} \quad (3.67)$$

where the integration by parts has been used and the terms containing second order gradients of the background quantities neglected. Putting Eqs. (3.64)–(3.67) back into Eq. (2.14) leads to

$$\begin{aligned} \frac{4\pi}{c} (\langle \mathcal{J}_{i,\parallel} \rangle - \mathcal{J}_{e,\parallel}) &= \frac{\beta_i}{\rho_{\text{th},i}^2} \int d\Omega f_{\text{eq},i} A_{1\parallel} \hat{A}_{1\parallel} + \frac{\beta_e}{\rho_{\text{th},e}^2} \int d\Omega f_{\text{eq},e} A_{1\parallel} \hat{A}_{1\parallel} \\ &\quad - \int d\Omega \nabla_{\perp} \cdot [(1 - \beta_i) \nabla_{\perp} A_{1\parallel} \hat{A}_{1\parallel}] \end{aligned} \quad (3.68)$$

The two skin terms can become very large, especially for electrons, due to their small mass. They cancel up to the second order FLR corrections with the adiabatic part of the currents  $\langle j_{i,\parallel} \rangle$  and  $j_{e,\parallel}$ . This can be seen by splitting the currents into an adiabatic and nonadiabatic part using the splitting defined in Eq. (3.63) and injecting them back into Eq. (3.68).

The cancellation problem occurs in PIC simulations due to the different discretizations of particles and fields: the currents are typically computed using the particles while the skin terms are computed using the finite element grid. The terms to be cancelled are much larger in magnitude than the remaining terms which are supposed to represent the physics. Therefore, the cancellation must be numerically extremely accurate, otherwise the relevant signal is dominated by numerical noise.

In ORB5, the cancellation problem is mitigated by discretizing both the skin terms and the adiabatic part of the currents in Eq. (3.68) with the same markers. The polarisation-current term,  $-\int d\Omega \nabla_{\perp} \cdot [(1 - \beta_i) \nabla_{\perp} A_{1\parallel} \hat{A}_{1\parallel}]$ , is discretized on the grid since it does not contribute to the cancellation. This approach to the discretization is used in ORB5 in Ampère's law.

Ampère's law, Eq. (3.68), is used to compute the parallel magnetic potential  $A_{1\parallel}$ . Note that the non-adiabatic perturbed distribution function  $G_{\sigma}$  depends on  $A_{1\parallel}$  which is unknown at this point of the computation. The solution is to use an easy-to-compute estimator  $\hat{S}$  and solve iteratively for  $A_{1\parallel}$ . In ORB5, the skin term  $(\beta_{\sigma}/\rho_{\sigma}^2)A_{1\parallel}$  is used as a simple estimator for the  $A_{1\parallel}$ -dependent part of the distribution function. One reformulates Ampère's law using the estimator  $\hat{S}$ :

$$(\hat{S} + L) a = (j - S a) + \hat{S} a, \quad (3.69)$$

where  $a$  is the discretized magnetic vector potential component,  $S$  and  $L$  are respectively the discretized skin terms and polarization current term, and  $j$  represents the sum over the species of the discretized currents. For a good estimator, a small parameter  $\|\hat{S} - S\| = \mathcal{O}(\varepsilon)$  can

be introduced to expand the vector potential,  $a = a_0 + \varepsilon a_1 + \varepsilon^2 a_2 + \mathcal{O}(\varepsilon^3)$ . Ampère's law is then solved iteratively order by order in  $\varepsilon$ :

$$\begin{aligned} (\hat{S} + L) a_0 &= j, \\ (\hat{S} + L) a_1 &= (\hat{S} - S) a_0, \\ &\dots \end{aligned}$$

In practice, for typical production runs, less than 10 iterations are necessary. In ORB5, the estimator is expressed using the finite elements  $\hat{S}_{kl} = \int \beta_\sigma / \rho_\sigma^2 \Lambda_k(\mathbf{x}) \Lambda_l(\mathbf{x}) d^3x$ . The marker-dependent part of the right-hand side of the iterative scheme is written as the enhanced control variates:

$$j_k - S_{kl} a_l^{n-1} = \sum_{v=1}^{N_p} p_{zv} \left( w_v + \frac{q_\sigma p_z \langle A_{1\parallel}^{(n-1)} \rangle}{m_\sigma} \frac{f_{\text{eq},\sigma}(Z_v) \zeta_v}{T_\sigma} \right) \langle \Lambda_k \rangle_v. \quad (3.70)$$

The same enhanced control variates is used also for the perturbed particle density. In practice, it results in a straightforward and computationally cheap modification of the charge and current assignment routines in ORB5.

#### 3.1.4 Heating operators

A primary goal of simulating the full plasma core (by contrast to a local approach) is to examine the self-consistent evolution of plasma profiles in the presence of both turbulence-driven transport and external sources. In practice, even for running global simulations where realistic global profile evolution is not of interest, it is generally inconvenient to run simulations without a heat source: if the goal is to look at transport properties at a specific temperature gradient, simulations where the temperature gradient relaxes rapidly evolve away from the desired parameters. In ORB5 temperature gradient control and injection of energy flux are imposed through sources added to the r.h.s. of the Vlasov equation. These do not model the detailed physics of a realistic heat source (for example, the radially localized temperature anisotropy generated by resonant heating schemes) but simply control moments of the distribution function.

For the control of the temperature gradient, so that it stays close to an initial gradient, a *thermal relaxation operator* is used (this can be seen as an effective interaction with a heat bath) of the form

$$S_{\text{HI}} = -\gamma_{\text{H}} \left[ \delta f(\epsilon, s) - f_0(\epsilon, s) \frac{\int d^3v \delta f(\epsilon, s)}{\int d^3v f_0(\epsilon, s)} \right]. \quad (3.71)$$

Note that contrarily to Eq. (3.49), here  $\delta f$  depends only on the radial position  $s$  and the kinetic energy  $\epsilon$ . This source term maintains the distribution function  $f(\epsilon, s)$  close to the initial value,

i.e. it relaxes back to  $f_0$  with a rate  $\gamma_H$ . Note that the heating operator, Eq. (3.71), does not act as a noise control, unlike the modified Krook operator defined in Section 3.1.3.1. Indeed, this operator applies the same correction to all the markers within a velocity bin, which does not reduce the noise caused by filamentation in velocity space. The second term in the equation ensures that the gyrocenter density is not modified by the source term, i.e. the heat source does not act as an effective particle source. Due to the symmetry of this operator in  $v_{||}$  it also does not add parallel momentum to the system; testing [80, 97] has shown that long wavelength flows are largely unaffected by this heat source although certain higher order effects could lead to significant flow drive on shorter wavelengths [98].

The choice of  $\gamma_H$  determines how strongly the temperature gradient is clamped to the initial gradient; since the form of the heating is not physical, it is necessary to set  $\gamma_H$  small enough to not excessively damp temperature corrugations; empirical investigations suggest that setting  $\gamma_H$  ten times smaller than typical instability growth rates is appropriate to ensure convergence. It is possible to specify this heat source to be active only in certain regions of the plasma, so that, for example, a “source-free” region in the middle of the simulation domain may be obtained.

Fixed-input power simulations may be obtained by using a *fixed heat source* of the form

$$S_{H2} = \gamma_R(s) \frac{\partial f_0}{\partial T}, \quad (3.72)$$

where  $\gamma_R(s)$  defines the spatial heating profile written in terms of an effective inverse timescale<sup>1</sup> over which the local temperature would vary in the absence of transport. Generally this operator is used to represent a fixed input power source in the core of the tokamak. To model the energy losses near the edge, two options can be chosen: first, to define a fixed heat sink by setting a profile  $\gamma_R(s)$  with negative values in the edge region; second, to define a buffer region near the boundary in which a Krook operator is specified (see Section 3.1.3.1), thus damping the edge profiles close to their initial values.

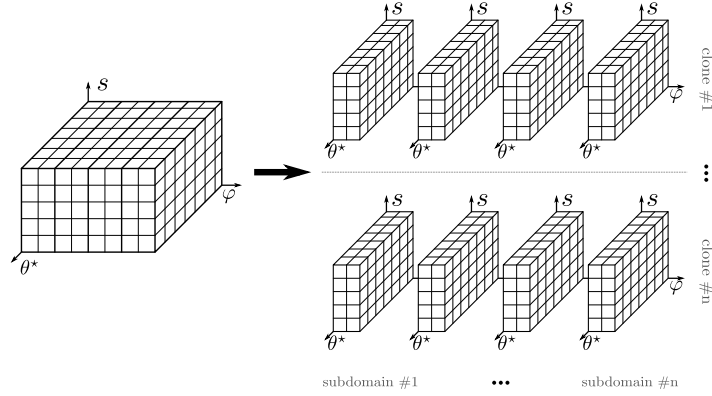
#### 3.1.5 Parallelization

In order to simulate complex physical systems in a reasonable amount of time, the ORB5 code is massively parallelized using a hybrid MPI/OpenMP and MPI/OpenACC implementation. The MPI parallelization is done using both domain cloning and domain decomposition [99, 100] techniques, Figure 3.1.

The physical domain is first replicated into disjointed clones and the markers are evenly distributed among them. Each clone can be further decomposed by splitting the physical domain in the toroidal direction into subdomains. Each subdomain of each clone is attributed to an MPI task such that the total number of processes is given by  $P_{\text{MPI}} = P_{\text{sub}} \times P_{\text{clones}}$ , where  $P_{\text{sub}}$  and  $P_{\text{clones}}$  are respectively the number of tasks attributed to the subdomains and clones.

---

<sup>1</sup>To be accurate,  $\gamma_R$  has units of a temperature over time.



**Figure 3.1 :** MPI parallelization using domain decomposition in the toroidal direction  $\varphi$  and domain cloning.

After each time step, data must be transferred between the clones and subdomains. For the clones, mainly global reductions of grid quantities are required, e.g. after each charge deposition step all the contributions from the clones must be gathered to compute the self-consistent electromagnetic fields. For the subdomains, it consists of nearest neighbour communications for the guard cells, global communications of grid data (parallel data transpose) for Fourier transforms and point to point communications of particle data where we exchange the particles that have moved from a subdomain to another. Note that in ORB5, the particle exchange algorithm is not restricted to the nearest neighbours, all-to-all is supported.

While the domain decomposition scales well with the number of subdomains, a large number of clones is problematic in terms of performance. Indeed, the domain cloning approach is quickly limited by the more demanding communications and the memory congestion due to the field data replication. To overcome this issue each MPI task is multithreaded using OpenMP. This has the main advantage of limiting the number of clones while still increasing the code performance by sharing the workload among threads.

To take advantage of the new HPC platforms equipped with accelerators, the ORB5 code has been recently ported to GPU using OpenACC. These developments will be detailed in a separate paper [101]. The choice of using OpenMP and OpenACC was motivated by the intent to keep all parallelization options in a single source code version.

### 3.2 Refactoring and optimization for multicore platforms

ORB5 is a large Fortran code of around 60'000 lines that has been developed for the last 20 years by many scientists. Over time, the code has grown and as more and more physics and numerical schemes have been added to it, it lost modularity and clarity due to a lack of refactoring<sup>2</sup>. The refactoring is a very important process in the development cycle as it

<sup>2</sup>The refactoring is a process during which the design of the code is improved in such a way to make it more

improves the design of the code which makes it clearer and easier to extend. This is in contrast to what is usually done in a simulation code where a lot of resources are spent adding new features without ensuring an overall coherent programming structure, which ultimately leads to a “spaghetti code”. A few issues made the code particularly difficult to work with. First, multiple versions with different physics modules existed, which forced the user to make tedious manual merges between these versions if needed. Second, there was no test suite to ensure that no bug was introduced during development.

A few years ago, it was decided to make a large refactoring of the ORB5 code which was mainly motivated by the new trend in high performance computing (HPC) platforms to include multi- and many-core processors such as the Intel many integrated cores (MIC) Xeon Phi and GPUs. To make an efficient use of these platforms, a code has to follow a certain number of programming practices and it was decided to refactor the ORB5 code to facilitate future testing and development. The refactoring process essentially consisted of:

- a big merge of all the code versions into a unique source code,
- a thorough cleaning of the code to eliminate all unused or out-of-date lines of code,
- a code documentation to describe the numerous routines and input parameters,
- a new workflow allowing developers to implement new features without impacting the production code,
- an automatic testing setup that allows one to quickly detect new bugs in the code.

In addition to the refactoring, an optimization process took place. It allowed the code to fully make use of the computing power of the modern HPC platforms. During this process, the data structures of the code have been changed and different optimization techniques have been applied to increase the code performance. Furthermore, the parallelism has been increased to make use of the multi- and manycore platforms. This was done through the inclusion of OpenMP and OpenACC directives that allow one to use thread-level parallelism and GPUs.

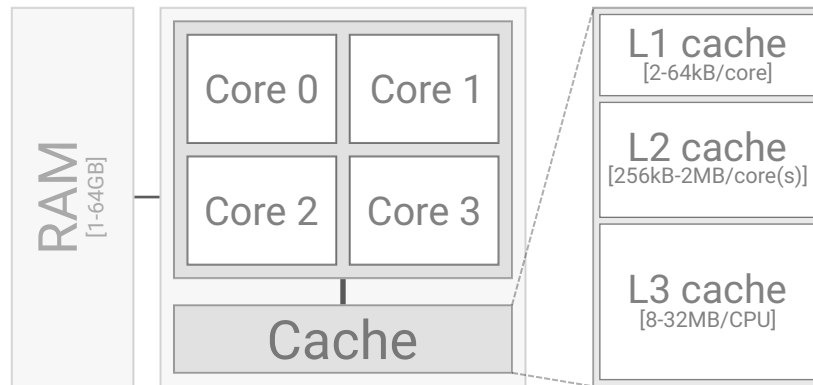
#### 3.2.1 Optimization techniques used to improve the code performance

ORB5 uses a PIC method and, as almost all particle-based codes, it is memory bound, i.e. the performance of the code is mainly limited by the speed of the data transfers from the main memory to the processor and vice-versa. It is thus important to improve data management, either on the hardware level or on the software level, to reduce this performance bottleneck.

Before presenting the optimization techniques used to improve the code performance, it is important to review how computers manipulate and work with the huge amount of data and instructions in order to write efficient code. To this end, let us focus on a conventional computer using a CPU; the case of GPUs will not be treated here.

---

readable, and easier to extend and work with, without changing its external behaviour.



**Figure 3.2** : Illustration of the CPU and its adjacent memory levels. Note that this sketch is only a simplified abstraction of the CPU architecture and does not represent the actual layout, which is more complicated in practice.

An illustrative abstraction of the CPU and its adjacent memories is sketched in Figure 3.2. From left to right, the first block is the random access memory (RAM) which can hold around 1–64 gigabyte (GB) of data. Then, the central block is the CPU chip, which is composed of several cores that perform the actual computations, and a cache memory. The cache memory can be decomposed into three levels, L1, L2, and L3 that can respectively store around 2–64 kilobyte (kB) per core, 256kB–2 megabyte (MB) per up to two cores, and 8–32 MB for the whole CPU. Note that Figure 3.2 is a simplified sketch. In reality, the CPU architecture is very complex and treating it in detail is beyond the scope of this discussion.

The purpose of the cache memory is to increase the CPU performance by providing a fast memory access. Indeed, following the *principle of locality*, a program tends to access the same data multiple times (temporal locality) and tends to access successively data that are stored close to each other in memory (spatial locality). To address this principle, the cache memory is an extremely fast—but also very expensive—memory. To limit the cost of a CPU, cache memories are usually small and most of the temporary storage is provided by the RAM. The main optimization process of memory-bound programs consist in improving the correct usage of cache memories. To illustrate the importance of a correct usage of the data in cache, Table 3.1, lists the estimated memory access times for a hypothetical 3.3GHz CPU with a cycle time of 0.3 ns. To better grasp the different access times, the last column of the table shows scaled values where the CPU cycle is set to 1 s.

The general workflow used by the CPU to fetch data is the following. Whenever the program requires a specific data, the CPU will first query the cache memories. If the data is present in the cache, it is a *cache hit* and the data can directly be loaded in the core registers. If the data is not present in the cache, it is a *cache miss* and the data has to be loaded first from the RAM into the cache, and then from the cache into the core registers. To address the spatial

### Chapter 3. ORB5 numerical implementation, refactoring, and optimization

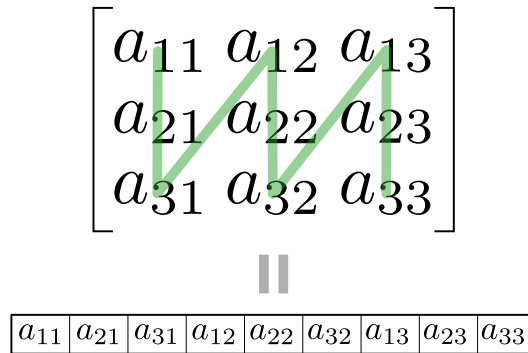
**Table 3.1** : Table illustrating the memory access latency for a hypothetical 3.3GHz CPU. The table shows estimated memory access times as well as scaled timings relative to the CPU cycle. The scaled values are obtained by scaling the reference CPU cycle to one second and normalizing the other timings accordingly. This table is partially reproduced from [102].

Event	Latency	Scaled
1 CPU cycle	0.3 ns	1 s
L1 cache access	0.9 ns	3 s
L2 cache access	2.8 ns	9 s
L3 cache access	12.9 ns	43 s
RAM access	120 ns	6 min
Solid-state disk access	50–150 $\mu$ s	2–6 days
Hard-disk drive access	1–10 ms	1–12 months

locality principle, if the data query results in a cache miss, not only the required data is read from the RAM and stored into the cache but a whole *cache line*, which is typically 64 bytes long, is loaded. This way, data stored in adjacent memory locations, that will most probably be used later, are already present in the cache.

Without presenting the details of implementation, we now discuss the different techniques used during the optimization process of ORB5. All these techniques mainly aimed at increasing data locality so as to better exploit the cached data.

Common memory storage, such as RAM, stores data in a linear way and a convention must be chosen for knowing in which order the dimensions of multidimensional arrays should be stored. The Fortran language uses a *column-major order* method in which the consecutive elements of a column are stored contiguously in memory, Figure 3.3. This means that whenever



**Figure 3.3** : Illustration of how a  $3 \times 3$  matrix is stored in a linear memory device such as RAM for a programming language using column-major order.

multiple loops are used to address all elements of an array, the fastest varying index, i.e. the

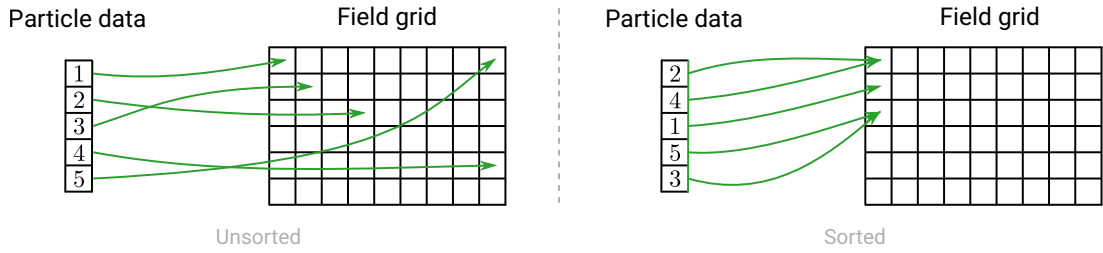


one of the inner-most loop, must represent the row index in order to have maximum performance. The benefits of doing this are twofold. First, since memory locations are accessed in a contiguous way, the ratio cache hit to cache miss is maximized thus improving performance. Second, this allows modern processors to use their *vector registers*. This particular type of register can perform the so-called single instruction on multiple data (SIMD) which consists in executing the same type of operation on multiple data in one clock step. This is the so-called vectorization.

One of the particularity of a PIC code is that it has both particle and field data. Usually, particle data consists of a set of attributes for each marker such as the phase space coordinates and particle weights. Many operations involve loops over particles. The way particle data is stored can again influence the code performance. For a storage using multidimensional arrays, which is the case for ORB5, particle data has been changed from array of structures (AOS) to structure of arrays (SOA). In the former, particle attributes are stored contiguously in memory, i.e. `particles(natts, np)`, where `particles` is a 2D array of size `natts×np` with `natts` and `np` being respectively the number of attributes and particles. In the latter, the array is transposed with respect to AOS and the particles are now contiguously stored in memory, i.e. `particles(np, natts)`. This option offers better performance because several neighbouring particles are loaded in the cache at once, thus increasing data locality.

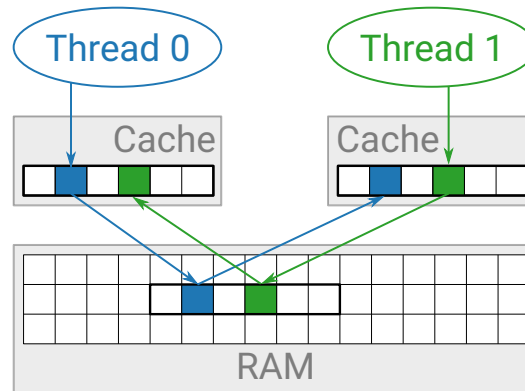
One of the main technical difficulties of PIC codes is the fact that particle data has to be interpolated to field grids and vice versa. To understand why this is problematic for performance, let us look at the charge deposition step. The goal of this operation is to interpolate the charge carried by the markers to the field grid. The deposited charge is then used in the quasi-neutrality equation to solve for the electric potential. This deposition operation is typically done by looping over the particles, and involves computing in which cell they lie in the field grid, and depositing the charge. Since particles and the charge array are stored independently in memory, there is no direct relation between them. This means that particles that are stored contiguously in memory may lie in very different grid cells, Figure 3.4 (left). This indirect addressing gives rise to a lot of random memory accesses that typically result in near to 100% cache misses. To avoid the indirect addressing, markers can be sorted according to their position. Note that the markers are not required to be fully sorted but only within the grid cell level, i.e. particles within the same grid cell do not need to be further sorted. In this way, the particle data structure and field grid are now linked since the markers are sorted so that neighbouring particles in the physical space are also stored contiguously in memory, Figure 3.4 (right). The use of particle sorting greatly increases data locality and removes the need for indirect addressing, also allowing for vectorization, which otherwise would be forbidden by the compiler.

As mentioned in Section 3.1.5, ORB5 makes use of OpenMP to further divide the workload among different OpenMP threads. It is exclusively used in routines that involve looping over particles. OpenMP usually ensures good scalability up to many cores but this performance



**Figure 3.4 :** Illustration of the charge deposition step with unsorted (left) and sorted (right) markers. The unsorted case leads to indirect addressing causing a lot of random memory access. On the other hand, in the sorted case, the indirect addressing is suppressed and all memory accesses are contiguous.

can be compromised by *false sharing*: to ensure consistency between the data stored in the cache and the data in the main memory, the CPU implements a cache coherence protocol that invalidate the data in the cache if the same data is modified in the main memory. This will automatically lead to a cache miss that will force the data in the cache to be reloaded. The false sharing occurs when multiple threads share the same cache line but work in distinct memory locations—this is typically the case of ORB5 1D diagnostics. Once one thread modifies its memory location, all the other threads sharing the same cache line in RAM must reload their data due to the cache coherence protocol, as illustrated in Figure 3.5. This leads to an increasing number of cache misses that slow down the program execution. A way to avoid false sharing is to make sure that the data used by a thread is located in a memory location that is at least a cache line size apart from the data used by another thread.



**Figure 3.5 :** Illustration of the false sharing. When a thread modifies a memory location in the main memory, all the other threads having this memory location in their cache must reload the data to ensure cache coherence.

### 3.3 The Pic\_Engine: a simple test bed for algorithmic testing

To test the different techniques presented in the previous section, as well as various new algorithms to improve the code performance, we have developed a test bed called PIC\_ENGINE. This platform has been developed with the aim of allowing one to easily implement and test algorithms in a framework retaining only the key elements of the PIC algorithm. It is designed to be modular and portable such that it can be used on different multi- and manycore architectures such as GPU-equipped machines [103, 104, 105]. Here, this study focuses on multicore CPUs and the Intel Xeon Phi Many Integrated Core (MIC) [106] processors. The portability is an important aspect of the work since the PIC\_ENGINE is designed to develop generic optimizations for the PIC algorithm that are then used in porting to multi- and manycore platforms full application codes such as the global gyrokinetic code ORB5 [77, 79, 10] and the RAMSES [107] code that models astrophysical systems. To this end, the PIC\_ENGINE is designed to be the simplest abstraction possible of the PIC algorithm but nevertheless retaining its essential elements. In this way, we can study the optimization techniques implemented on the most general parts of the algorithm and treat the fundamental parallelization problems inherent to the PIC method.

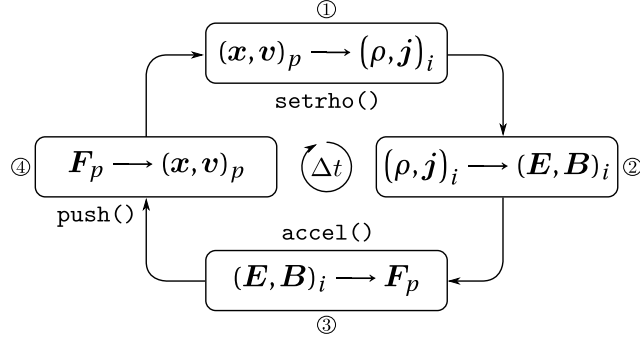
As a base case for this work, the PIC\_ENGINE evolves a simple plasma physics system and the methods will thus be designed with this goal in mind. The portability is ensured by only treating the elements of the PIC algorithm that are common between different application codes.

#### 3.3.1 The Pic\_Engine platform

The standard PIC algorithm adopts a Lagrangian description of the particle distribution which is evolved following four steps, as seen in Figure 3.6. The first one is the charge assignment, called `setrho`, in which the charge and current carried by the particles are deposited from the particle positions ( $p$  index) to the field grid ( $i$  index). In the second step, the self-consistent electromagnetic fields  $\mathbf{E}$  and  $\mathbf{B}$  are computed on the grid according to Maxwell's equations and for the charge and current deposited previously. In the following step, called `accel`, the Lorentz force acting on the particles is evaluated and interpolated back from the field grid to the particle positions. Finally, the particle positions and velocities are updated in the push step. A PIC simulation typically consists of a sequence of time iterations during which each of these four steps is carried out.

##### 3.3.1.1 Structure of the Pic\_Engine

With the aim of developing and testing optimization techniques and parallelization schemes for porting PIC codes to the new multicore and MIC architectures, the PIC\_ENGINE has been implemented using Fortran and parallelized with both OpenMP and MPI. The OpenMP API (Application Programming Interface) has been chosen as it is standardized and portable. Furthermore, it is usually easy to handle and can be implemented incrementally. Note that



**Figure 3.6 :** Time loop of the PIC algorithm. Each iteration consists of four steps. (1): The contribution to the charge and current fields  $(\rho, j)$  are deposited from the particle positions  $(p$  index) to the field grid  $(i$  index). In the `PIC_ENGINE`, this step is called `setrho` and only the charge is deposited. (2): Using the deposited charge and current, the electromagnetic fields  $E$  and  $B$  are computed by the field solver. (3): The Lorentz forces  $F$  acting on the particles are estimated based on the electromagnetic fields which are interpolated from the grid to the particle positions  $x$ . (4): These forces allow updating the particle velocities  $v$ . Finally, the particle positions are updated. In the `PIC_ENGINE`, the last two steps are called `accel` and `push` respectively.

with this programming model, the same source code can be run either on CPUs or MICs in native or symmetric modes, i.e. treating the MICs as standalone processors. The `PIC_ENGINE` corresponds to an abstraction of the PIC algorithm presented in the previous paragraph, retaining only its key elements. For the sake of simplicity, it evolves a single-species plasma in the electrostatic limit. No magnetic field is considered and the electric field  $E(x)$  is imposed, i.e. dropping step (2) in algorithm summarized in Figure 3.6, assuming periodic boundary conditions in all three configuration space dimensions and stationary in time. The electric field has been defined as follows:

$$\phi(x, y, z) = \cos(k_x x) \cos(k_y y) \cos(k_z z), \quad (3.73)$$

$$E(x, y, z) = -\nabla \phi(x, y, z), \quad (3.74)$$

where  $\phi(x, y, z)$  is the electrostatic potential and  $k_i = n_i 2\pi / L_i$ ,  $n_i \in \mathbb{N}$ ,  $i = x, y, z$  are the components of an arbitrary wave vector. Consequently, no field solver is required here. Note that the computation of the fields reduces to a linear problem that can be solved with various types of approaches, e.g. Fourier, direct, iterative and multigrid methods. Since these methods and their parallelization are well known, they are not considered here. Furthermore, in general, the field solve duration only represents a small fraction of the timeloop in ORB5. Due to the electrostatic approximation, only the charge is deposited on the field grid in the charge assignment step `setrho`. In the `accel` step, to evaluate the Lorentz force  $F$ , the electrostatic field is interpolated from the field grid to the particle positions and then used to evolve the

### 3.3. The Pic\_Engine: a simple test bed for algorithmic testing

---

particle velocities in the push step. Similarly, the positions of the particles are also updated in the push method. In these last two steps, the time integration is done using a second order accurate leap-frog scheme that is energy conserving and reversible [108], see Section 3.3.1.1 for more details.

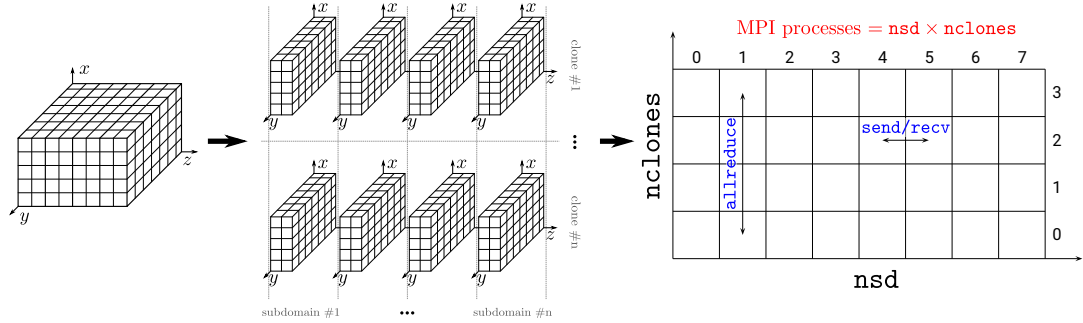
At this point, it is important to note that the `PIC_ENGINE` is only meant to study the algorithmic performance of the different parts of the PIC algorithm and no physical results are sought. Thus, the absence of a field solver and the use of the aforementioned approximations are not critical for our study. Furthermore, with a field solver and some minor changes, the `PIC_ENGINE` can solve different plasma physics problems of interest [109, 110, 111, 112]. Also, we keep in mind that for application codes such as ORB5 using non-standard PIC methods, e.g. involving higher order schemes, or more complex equations of motion such as the gyrokinetic Vlasov-Maxwell system, the numerical cost of `accel`, `push` and `setrho` is greatly enhanced by the complexity of the problem. Despite the very simple abstraction of the PIC algorithm implemented in the `PIC_ENGINE`, all the conclusions drawn from this test bed were found to also hold for ORB5.

**Data structure and particle initialization** In the `PIC_ENGINE`, field and particle data structures have to be distinguished. The field quantities, such as the electrostatic field and the particle charge density, are represented on a 3D periodic domain of length  $l_x$ ,  $l_y$  and  $l_z$  discretized in  $n_x$ ,  $n_y$  and  $n_z$  intervals. The  $x$ ,  $y$ ,  $z$  letters represent along which direction of the Cartesian coordinates  $(x, y, z)$  the quantity applies. On the other hand, particle quantities, i.e. their position and velocity, can take any values in the phase space  $(\mathbf{x}, \mathbf{v})$ , and are stored in the `part_att(natts,np)` array, where `natts` represents the six particle attributes, namely the Cartesian coordinates of the position and the corresponding three coordinates of the velocity, and `np` is the number of particles in the system. The `part_att` array can be stored either in Array Of Structures (AOS), i.e. `part_att(natts,np)` or in Structure Of Arrays (SOA), i.e. `part_att(np,natts)`. As we shall see in the next sections, the way of storing data in memory significantly affects the performance.

At the beginning of the simulation, the particle position and velocity components are initialized using the Fortran `random_number` method. The positions are uniformly distributed in the periodic configuration domain. The velocities  $v_x$  and  $v_y$  are initially distributed uniformly with  $|v_x, v_y| \leq v_{\max}$ , where the parameter `vmax` sets a maximum velocity while  $v_z$  can be either uniformly distributed as  $v_x$  and  $v_y$ , or distributed according to a normalized Boltzmann distribution.

**Parallelization** The `PIC_ENGINE` parallelization is achieved using a hybrid MPI/OpenMP implementation, see Figure 3.7. Note that the MPI parallelization is essentially the same as in ORB5, Section 3.1.5 and Figure 3.1. First, the 3D domain is decomposed into `nsd` subdomains in the  $z$  direction. Each subdomain grid data is then replicated into `nc1ones` clones and this

2D decomposition is mapped onto an MPI grid. Finally, a third level of parallelism is added using OpenMP by assigning `nthread` threads to each MPI task, so that the total number of cores<sup>3</sup> used is equal to the product  $\text{nsd} \times \text{nclones} \times \text{nthread}$ . Usually, the number of cores used by the program is equal to the number of physical cores on the processor. However, using Intel's Hyper-Threading technology, it is possible to initialize more threads than physical cores. Depending on the application, this can be beneficial or, in the worst case, can decrease the performance.



**Figure 3.7 :** Illustration of the PIC\_ENGINE parallelization. The 3D domain is first decomposed into  $\text{nsd}$  subdomains in the  $z$  direction. Each subdomain is then copied  $\text{nclones}$  times and this 2D decomposition is mapped onto an MPI 2D grid. Finally, OpenMP is introduced and  $\text{nthread}$  threads are affected to each clone. With this parallelization scheme, only two types of communications are needed for three operations: a reduction is done across the clones to gather the partial charges with MPI `allreduce` and particles and guard cells are transferred across subdomains with MPI point-to-point `sendrecv`.

It is important to note that communications between MPI tasks are only needed for three operations. First, all the partial charges computed by `setrho` on each clone of one subdomain are added using MPI `allreduce` operations and the guard-cells must be exchanged with neighboring subdomains using MPI point-to-point `sendrecv`. Then, the particles have to be transferred between subdomains; this is done by the `pmovez` routine using MPI point-to-point `sendrecv` directives. The `pmovez` method allows one to transfer the particles not only to the nearest neighboring subdomains but is applicable to an arbitrary source/destination pattern.

**Charge deposition, particle accelerating and pushing** As we have seen at the beginning of this section, the PIC\_ENGINE consists of three main steps: the charge deposition, the particle accelerating, and pushing. In the program, they are implemented respectively in the `setrho`, `accel` and `push` methods. In this section, we discuss their implementation which is mainly inspired by [108].

The first step in the PIC algorithm is the charge deposition. Its goal is to collect the electric

<sup>3</sup>Here, core refers either to MPI task or OpenMP thread.

### 3.3. The Pic\_Engine: a simple test bed for algorithmic testing

charge density  $\rho$  of all particles and to deposit it on the field grid with the aim of computing the electrostatic field. Thus, for each particle, we have to compute

$$\rho_{\text{grid}} \leftarrow \rho_{\text{grid}} + \rho_p^{\text{dep}} \text{ (particle to grid)}, \quad (3.75)$$

where  $\rho_{\text{grid}}$  is the charge density already deposited on the field grid and  $\rho_p^{\text{dep}}$  is the contribution of the current particle deposited on this same grid using a first-order deposition scheme. This approach, also called Cloud-In-Cell (CIC), works as follows. For a particle that lies in the interval  $[x_i, x_{i+1})$ , we first compute its relative position with respect to  $x_i$ :

$$\xi_p = \frac{x_p - x_i}{\Delta x},$$

where  $x_p$  is the particle position and  $\Delta x$  is the length of the interval. Then, we deposit the charge according to

$$\rho_p^{\text{dep}}(x_i) = 1 - \xi_p, \quad \rho_p^{\text{dep}}(x_{i+1}) = \xi_p.$$

Note that in the PIC\_ENGINE, the equations are normalized such that the elementary charge  $e$  and the mass  $m$  of the particles are unity in absolute value. Because particles are distributed randomly in configuration space, the charge deposition clearly involves an indirect addressing to the  $\rho$  array when depositing the contribution from each particle. This indirect writing has to be taken care of carefully when parallelizing the `setrho` method for shared memory systems to avoid race conditions, i.e. when two or more threads try to modify data stored at the same memory location.

The particle equations of motion

$$\frac{d\mathbf{v}}{dt} = \mathbf{F} = -\mathbf{E}, \quad (3.76)$$

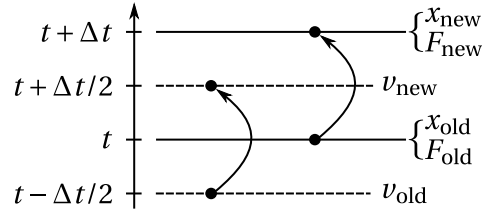
$$\frac{d\mathbf{x}}{dt} = \mathbf{v}, \quad (3.77)$$

are respectively solved by the `accel` and `push` methods. To evolve the particle positions and velocities, we use the leap-frog method [108] that consists in discretizing the equations with a finite-difference scheme:

$$\frac{\mathbf{v}_{\text{new}} - \mathbf{v}_{\text{old}}}{\Delta t} = \mathbf{F}_{\text{old}} \implies \mathbf{v}_{\text{new}} = \mathbf{v}_{\text{old}} + \Delta t \mathbf{F}_{\text{old}}, \quad (3.78)$$

$$\frac{\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}}{\Delta t} = \mathbf{v}_{\text{new}} \implies \mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \Delta t \mathbf{v}_{\text{new}}, \quad (3.79)$$

where  $\mathbf{F}_{\text{old}}$  is the Lorentz force computed at  $\mathbf{x}_{\text{old}}$  and  $\Delta t$  is the numerical time step. They are then evolved on two separate time-grids staggered by  $\Delta t/2$ , see Figure 3.8. By doing so, the leap-frog method is time centered and thus second order accurate with respect to  $\Delta t$ .



**Figure 3.8** : Illustration of the leap-frog method.

When using this method, two issues must be correctly addressed. First, since the velocities and positions are evolved on two different time grids, care must be taken when computing any quantity involving both the particle positions and velocities such as the total energy. Secondly, at the beginning of the simulation, both the velocities and positions have to be correctly initialized on their respective time grid. Even though the `PIC_ENGINE` is not meant to produce physically relevant results, the energy diagnostic is very useful to check the correctness of the implementation of the various algorithms.

In the `accel` method, the electrostatic potential is linearly interpolated from the field grid to the particle positions in the  $x$ ,  $y$  and  $z$  directions. The electrostatic field components  $E_x$ ,  $E_y$  and  $E_z$  that result from the gradient of the potential are, in their turn, piece-wise *constant* in the  $x$ ,  $y$  and  $z$  directions, respectively, and piece-wise *linear* in the other two directions. This method is not subject to race conditions because there are no concurrent writes and can, therefore, be parallelized in a straightforward way. However, due to the indirect read of the field array, vectorized operations are not possible because they are not allowed by the compiler.

In summary, the push method can be trivially parallelized using OpenMP since it consists of vector add and multiply (SAXPY) operations. The most challenging methods to parallelize are in fact `accel` because of the indirect addressing and `setrho`, as it involves indirect writes that are subject to race conditions. For these reasons, we shall particularly focus here on `accel` and `setrho`.

### 3.3.1.2 Particle sorting

In the `PIC_ENGINE`, and more generally in most PIC codes, the particles are randomly distributed in space. As a consequence, data locality is poor and performance drops due to non-contiguous memory writings in `setrho` and accesses in `accel`. Indeed, cache memories are designed to be efficient with consecutive data accesses. If this is not ensured, significant overhead time will be spent by reloading the data from the main memory. With increased data locality, it is more likely that the data already in the cache can be reused for further computations, thus avoiding cache misses and reloading.

To this end, we have implemented a sorting method based on the counting sort algorithm



[113]. As a reminder, the algorithm sorts the particles into buckets that typically represent one or more grid cells. This is done in three main steps, see Algorithm 1:

1. In this first step, we create a histogram of the  $np$  particles by parsing them and computing the indices of their recipient bucket. To this end, the 2D  $(x, y)$  plane is mapped on a 1D vector using a linear mapping, i.e.

$$\text{index} = \left\lfloor \frac{x_p}{\Delta x} \right\rfloor + \left\lfloor \frac{y_p}{\Delta y} \right\rfloor nx, \quad (3.80)$$

where  $\Delta x$ ,  $\Delta y$ ,  $x_p$  and  $y_p$  are the interval lengths of the sorting buckets and the particle positions in the  $x$  and  $y$  directions respectively. Note that we consider only domains with one grid interval in the  $z$  direction but a 3D generalization is straightforward. Other options like Hilbert filling curves [114] are also of interest to preserve data locality across the mapping but they are not considered here.

2. In the second step, an inclusive prefix sum (cumulative sum) is made. It essentially consists of parsing the  $nbtot$  buckets to compute the *ending* position of each bucket in the sorted particle array. In other words, it consists of identifying how many particles each bin contains.
3. Finally, using the results of the prefix sum, the particles are put in their sorted place.
4. The temporary sorted array is copied back to `part_att`.

---

**Algorithm 1** The counting sort algorithm is composed of three main loops that respectively compute the histogram of the particles in the buckets (lines 1 to 4), compute the starting and ending indices of each bucket with a prefix sum (lines 5 to 8) and move the particles in their correct order with the help of a temporary array (lines 9 to 13).

---

```
1: for  $ip = 1, np$  do
2:    $id = \text{index}(ip)$            ▷ The index function returns the particle grid-cell index
3:    $count(id) = count(id) + 1$ 
4: end for
5:  $displs(1) = count(1)$ 
6: for  $ib = 2, nbtot$  do
7:    $displs(ib) = displs(ib - 1) + count(ib - 1)$ 
8: end for
9: for  $ip = np, 1, -1$  do
10:   $id = \text{index}(ip)$ 
11:   $part\_att\_tmp(:, displs(id)) = part\_att(:, ip)$ 
12:   $displs(id) = displs(id) - 1$ 
13: end for
```

---

Note that the fourth step was added because this sorting algorithm is out-of-place and hence requires a temporary array. Memory can be saved with in-place versions but such methods [113] have not been implemented because memory was not an issue for this application.

A straightforward algorithmic analysis shows that the timing of this sorting is linear in the number of particles and buckets, i.e.  $T_{\text{psort}} = \mathcal{O}(np) + \mathcal{O}(nb)$ , where  $T_{\text{psort}}$  is the serial timing of the sorting method.

**Vectorization** Both `setrho` and `accel` methods have to read the particle positions to compute in which grid cell they belong. This requires an indirect addressing in the particle loop which prevents the compiler to use auto-vectorization because no such vectorization operations are in the instruction set of the processor. To solve this issue, we can take advantage of the particle sorting. Indeed, with a full sorting, i.e. if the buckets coincide with the grid cells, all the particle loops with indirect addressing can be replaced by a double loop involving first a loop over the buckets and a second loop over the particles in each of these buckets, see Algorithm 2. Indeed, by knowing which bucket one is treating, one automatically knows the field grid positions of the particles if they have been fully sorted. By doing so, the inner loop is totally independent of the indirect addressing and the compiler can vectorize it. This method can be beneficial if the inner loop is long enough to allow a good vectorization.

---

**Algorithm 2** Illustration of the vectorization procedure. In the first method (lines 1 to 4), the grid cell index of each particle is computed and rest of the routine is executed. In the second method (lines 6 to 12), the loop over the particle is decomposed into a loop over the grid cells and a loop over the particles in the grid cells. With this approach, all the particles within each bucket have the same grid cell index for the inner loop. Hence, the compiler can vectorize it.

---

```
1: for  $ip = 1, np$  do
2:   ! Compute particle grid cell index
3:   ...
4: end for
5:
6: for  $ib = 1, nx * ny * nz$  do
7:
8:   for  $ip = displs(ib) + 1, displs(ib + 1)$  do
9:     ! Then do the calculation for each particle in the current grid cell
10:    ...
11:   end for
12: end for
```

---

### 3.3.2 Performance analysis

In this section, we present the different optimizations made to the `PIC_ENGINE` as well as the timings and performance gains obtained. The study was performed on the Piz Daint and Piz

### 3.3. The Pic\_Engine: a simple test bed for algorithmic testing

---

Dora supercomputers from the Swiss National Supercomputing Centre (CSCS) and the Helios supercomputer at the Computational Simulation Center of the International Fusion Energy Research Centre (IFERC-CSC) in Japan. All the relevant characteristics of these machines are summarized in Table 3.2.

For both CSCS machines, Piz Daint and Piz Dora<sup>4</sup>, the Cray compiler cce 8.3.12 and the Cray MPICH 7.2.2 library were used. On Helios, the Intel compiler 15.0.2 in conjunction with Intel MPI 5.0.3 was used. Except for the MICs, which showed a better performance with four threads per core, the hyper-threading was always disabled to obtain an optimal performance. All the available cores were used by ensuring that the product  $nsd \times nclones \times nthread$  was always equal to the total number of cores. In this work,  $nclones$  equal to 1 corresponds to a pure OpenMP run and  $nclones$  equal to the number of cores per node corresponds to a pure MPI run. Note that when we talk about pure OpenMP runs, there may be several MPI processes for the inter-node communications which is typically the case for multi-node runs. This terminology is only used here to emphasize that within each node, all the parallelization is made using OpenMP.

For the MIC nodes, three different run modes are available on Helios: offload, symmetric and native. In this work, to ensure a homogeneous performance among the processors and avoid complicated interactions between MICs and CPUs, we focus on the latter which consists of using the MICs as standalone processors. This is possible because each of them is equipped with a Linux micro operating system. In this mode, only the MICs are used, as opposed to the symmetric mode where both the CPUs and MICs are used in a similar way. Finally, the offload mode consists of using the MIC as an accelerator, in which case the host, generally the CPU, starts the application and offloads some data and computations to the MIC.

The standard single-node problem tested in this study consists of a  $512(nx) \times 256(ny) \times 1(nz)$  grid with  $10^6$  particles similar to [103] for the sake of comparison. Whenever activated, the sorting was always full, i.e. we considered 512 and 256 buckets in the  $x$  and  $y$  directions respectively; the  $z$  direction is trivially sorted as we consider only one  $(x, y)$  plane per subdomain. All the timings given here are normalized to the standard single-node problem and are reported in nanoseconds per particle and per time step.

In the following, if not mentioned otherwise, only the results on Helios CPU and MIC nodes are presented for the sake of conciseness. For every conclusion, it has been checked that it also applies to Piz Daint and Piz Dora.

#### 3.3.2.1 Data structure

As a first result, the effect of the data structure is studied. To this end, the code is compiled with either SOA or AOS data structures and is run on the Helios machine on both CPU and MIC nodes. The single-node results for the CPU case are presented in Table 3.3 for both pure

---

<sup>4</sup>Piz Daint and Piz Dora were two clusters from CSCS before the upgrade that took place in November of 2016 and during which the CPUs were updated and both machines merged into one.

### Chapter 3. ORB5 numerical implementation, refactoring, and optimization

**Table 3.2 :** Characteristics of the clusters used for this study. Piz Daint and Piz Dora are two computers from the Swiss Supercomputing Centre (CSCS) and Helios is a magnetic fusion research supercomputer from the Computational Simulation Centre (CSC) in Japan. Note that this last computer is equipped with both CPU and MIC nodes.

	<b>Piz Daint</b> (Cray XC30)	<b>Piz Dora</b> (Cray XC40)	<b>Helios</b> (CPU)	<b>Helios</b> (MIC)
<b>Socket(s)</b>	1	2	2	2
<b>CPU</b>	Intel Xeon	Intel Xeon	Intel Xeon	Intel Xeon
	E5-2670	E5-2690 v3	E5-2680	Phi
	Sandy Bridge	Haswell	Sandy Bridge	KNC
<b>Frequency</b>	2.6 GHz	2.6 GHz	2.7 GHz	1.3 GHz
<b>Cores</b>	8/CPU	12/CPU	8/CPU	60/MIC

MPI (nclones = 16) and pure OpenMP (nclones = 1). The SOA data structure is always faster than AOS. Most of the gain comes from the `setrho` routine resulting in an overall time gain of 6.4% for the pure MPI case and 14.4% for pure OpenMP. Similar results are observed on the MIC nodes.

In the remaining of this work, we have always used the SOA data structure as it is the most efficient option.

**Table 3.3 :** Comparison between the AOS and SOA data structures on Helios (CPU) for the pure MPI and OpenMP cases. All the timings are in ns/particle/ $\Delta t$ . In both cases, the standard problem with unsorted particles is considered. Similar results are observed for the MIC equipped nodes.

	<b>Pure MPI</b>			<b>Pure OpenMP</b>		
	AOS	SOA	Gain	AOS	SOA	Gain
<code>push()</code>	0.84	0.78	7.95 %	1.43	1.40	2.00%
<code>accel()</code>	8.58	8.47	1.25 %	4.25	4.15	2.53%
<code>setrho()</code>	9.17	8.21	11.67%	5.78	4.47	29.34%
Total	18.62	17.5	6.40 %	11.47	10.03	14.36%

#### 3.3.2.2 Particle sorting

Different thread-parallel implementations of the sorting algorithm have been designed using OpenMP. Essentially, out of the three loops presented in Algorithm 1, only the prefix scan (second loop) is not parallelized because it is quite complex to do so and it never represents more than 4% of the total timing of the sorting. Here is a brief description of the different versions implemented. More details can be found in A.1.

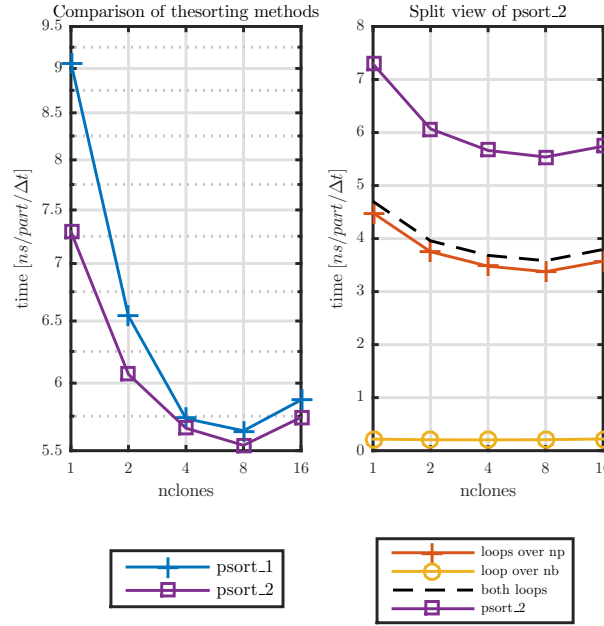
`psort_1`: The first loop is parallelized using OpenMP `reduction` while the third loop uses

### 3.3. The Pic\_Engine: a simple test bed for algorithmic testing

the `parallel do` directive in parallel with `atomic` to avoid race conditions.

`psort_2`: This implementation slightly differs from Algorithm 1. During the first loop, the histogram is computed as well as the index of each particle in the sorted array and OpenMP `atomic` is used to avoid race conditions. By doing so, the third loop only consists in a simple OpenMP `parallel do` loop.

In Figure 3.9 (left), we show the timings of the sorting methods for different hybrid runs, i.e. we vary the number of clones and threads such that their product corresponds to the total number of available cores per node.



**Figure 3.9** : Comparison of the different sorting parallelizations on Helios CPU (left) and split view of `psort_2` showing the timings of the loops composing the sorting algorithm (right). In each case, the timings are shown on a full node such that the product `nclones × nthread` is always equal to the total number of cores.

For the CPU nodes, the timings of the two sorting methods have the same qualitative behavior. Their costs are maximum for pure OpenMP (`nclones` = 1). They then decrease towards their minimum value for `nclones` = 8 and finally, increase for pure MPI (`nclones` = 16). To explain this, first recall that the sorting timing depends linearly on both the number of particles and buckets:

$$T_{\text{sort}} = T_{\text{np}} + T_{\text{nb}}, \quad (3.81)$$

where  $T_{\text{np}}$  and  $T_{\text{nb}}$  are two linear functions of the number of particles and buckets respectively.

Also, with the decomposition in clones, we have:

$$N_{p_{\text{clones}}} \equiv \frac{N_{p_{\text{tot}}}}{N_{\text{clones}}}, \quad (3.82)$$

where  $N_{p_{\text{clones}}}$  and  $N_{p_{\text{tot}}}$  are respectively the number of particles per clone and the total number of particles and  $N_{\text{clones}}$  is the number of clones. Thus, with an increasing number of clones, each of them must perform the sorting on fewer particles.

In Figure 3.9 (right) we show the timings of the loop over the buckets (loop over nb) and the two loops over the particles combined (loops over np) composing the sorting method for `psort_2`. First, note that the number of particles sorted per core is constant for the different hybrid runs since the product `nclones × nthread` is constant. This means that  $T_{\text{np}}$  should be constant. However, for pure OpenMP, we think that the high number of threads tends to increase false sharing. The loop over the number of buckets is negligible, never more than 4% of the sorting total timing. Finally, the difference between the timing of `psort_2` and the sum of the three loops is due to the copy of the temporary particle array to `part_att`.

In conclusion, method `psort_2` is found to be the most efficient on the CPU nodes. For that reason, it is always used in the following results whenever sorting is activated. Equivalent conclusions are found for the MIC nodes.

#### 3.3.2.3 Charge deposition

We have seen that both `accel` and `push` are thread-parallelized in a straightforward way, typically with OpenMP parallel loops, whereas care must be taken with `setrho` because of the race conditions, furthermore complicated by the indirect writing.

In the `PIC_ENGINE`, four different parallelization procedures for the charge deposition are implemented:

**setrho\_1:** *Threads on particles, collision free, with data replication.* In this method, threads are defined on the particles and we try to mimic OpenMP `reduction`. To this end, race conditions are avoided using private copies of the  $\rho$  array for each thread. Although being data safe, this algorithm requires more memory and a scalar reduction must be made at the end of the routine to collect all the data from the threads.

**setrho\_2:** *Threads on particles, collision resolving, no data replication.* The OpenMP `atomic` directive is used to resolve the race conditions.

**setrho\_3:** *Threads on particles, collision free, with data replication.* Similarly to the first method, we use the OpenMP `reduction` directive to avoid race conditions.

**setrho\_4:** *Threads on grid points, collision free, no data replication.* In this method, we use the finite support of the linear CIC method to compute the charge and avoid race conditions. Indeed, with this scheme, a particle will only deposit its charge to the eight

nearest grid points (for a 3D problem) so the algorithm consists basically in parsing first the grid points and then the particles contained in the eight nearest grid cells. Note that this method requires the particles to be sorted according to their positions in bins that coincide with the cells of the field data.

More details on these parallelizations can be found in A.2.

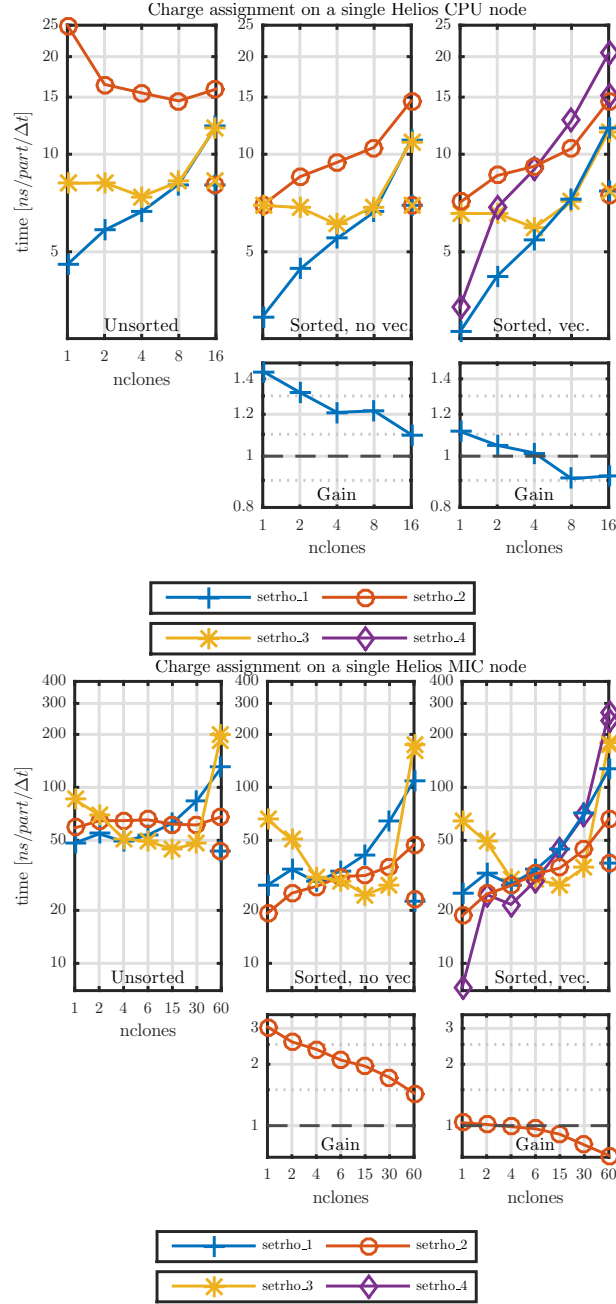
In Figure 3.10, we show the `setrho` timings for the four implementations, for unsorted and sorted particles, without and with vectorization, on a single Helios node. Both CPU (left) and MIC (right) results are shown.

Beginning with the CPU case with unsorted particles, we see that `setrho_1` is the fastest with a best timing of 4.8 ns per particle and per timestep in the pure OpenMP case. This represents a gain of around 2.8 compared to the pure MPI run that is mainly explained by the reduction of memory contention due to the use of a shared memory programming language such as OpenMP. For 8 and 16 clones, `setrho_1` and `setrho_3` have similar timings. This is not surprising as they are both based on a reduction approach. Indeed, using the OpenMP standard, we tried to mimic the OpenMP reduction in `setrho_1`. However, `setrho_3` is slower for `nclones` < 8. We think that this is because, in our case, the problem is too simple and thus dominated by the OpenMP overhead.

For the sorted particles without vectorization, we first observe that `setrho_2` is greatly improved because, as the particles are now sorted, the OpenMP `atomic` occurrences are reduced. Furthermore, due to the enhanced data locality, the timings are improved for all methods. On the gain graph below, we show the time gain of the optimal method, `setrho_1`, as compared to the unsorted case. The maximum gain of 44% is obtained for the pure OpenMP case and decreases to 10% for the pure MPI run.

Similarly, the third column of the figure shows the timings in the sorted and vectorized case. Note that we have now included `setrho_4` as it requires a full sorting and is vectorized by default. This method has the advantage of not being subject to race conditions and not using additional memory. However, it is only beneficial for high number of threads (`nthread` > 8) otherwise, the method is the slowest for `nthread` < 4. For the other methods, the conclusions are the same as for the sorted case. The vectorization allows one to have a small gain for a number of clones between one and four but it becomes negative for higher values. It has to be noted that the vectorization is poor because there are too few particles per grid cell. Indeed, as we will see later, the performance of the vectorization depends on the number of particles per grid cell; it needs a high particle density to be efficient.

In summary, for all CPU cases, `setrho_1` with homemade reduction is the fastest method. The method `setrho_3` using OpenMP reduction has a similar timing as the first method for `nclones` ≥ 8 but becomes surprisingly slower for `nclones` < 8. Although greatly improved by the sorting, `setrho_2` has a poor performance compared to the reduction methods because of the OpenMP `atomic` directives. Finally, `setrho_4` uses no additional memory and is not subject to race conditions. Furthermore, for `nthread` = 16, it is the second fastest candidate



**Figure 3.10 :** Comparison of the different charge assignment methods on a single CPU (top) and MIC (bottom) Helios node. Unsorted and both sorted without and with vectorization cases are considered. Method `setrho_4` is only present in the sorted and vectorized case because it requires a full sorting and it is intrinsically vectorized. For both CPU and MIC, the center gain plots show the performance gain of the sorted version of respectively `setrho_1` and `setrho_2` against the unsorted. Similarly, the right gain plots show the performance gain of the vectorized version of respectively `setrho_1` and `setrho_2` compared to the sorted version without vectorization. The unconnected data points (`nclones` = 16 for CPU and `nclones` = 60 for MIC) show the timing of the pure MPI version of the code compiled without

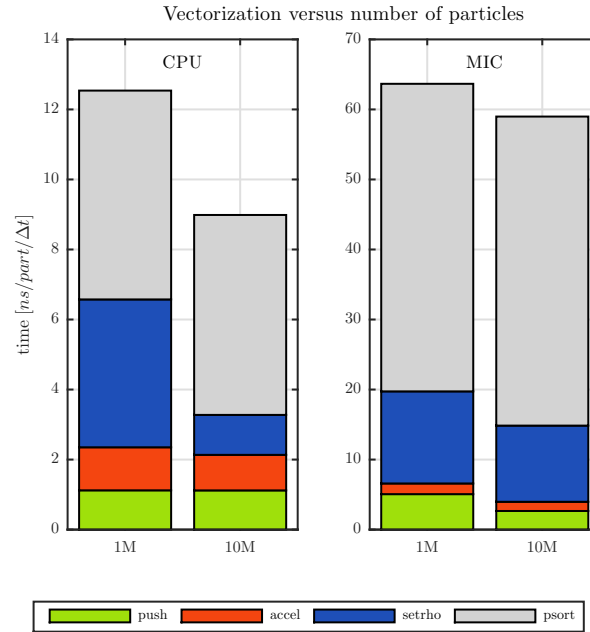


but otherwise, it is among the slowest of our charge deposition methods.

For the MIC case, the best option is not as obvious. Indeed, depending on the number of clones, different `setrho` methods are optimal. Even though `setrho_4` is more than two times faster for pure OpenMP, we have chosen, for the following, `setrho_2` (with `atomic` directives) because it is the second fastest around pure OpenMP, where we expect the program to run fastest, and it is not constrained to sorted particles with vectorization. Furthermore, where it is not optimal, only a small difference is observed compared to the fastest method.

The sorting allows one to reduce the timing of `setrho_2` by factors from 1.5 to 3 for the same reasons as the CPU case. Essentially, with sorted particles, the atomic occurrences are reduced.

Similarly to the CPU case, the performance gain due to the vectorization is very small because too few particles per grid cell are present. It has been checked, see Figure 3.11, that with around 10 times more particles per grid cell, i.e. a total of 10 million particles, the vectorization performance is enhanced. In the CPU case, the overall gain is of 40% with an



**Figure 3.11** : Performance gain with different number of particles for both CPU (left) and MIC (right). In each case, the timings of `push`, `accel`, `setrho` and `psort` are presented for the standard case with 1 million particles and a test case with 10 million particles.

impressive gain of a factor 3.7 for `setrho`. For the MIC, the overall gain with 10 million particles is of only  $\sim 10\%$ . This is less than the CPU case because the MICs have vector registers twice as big as the CPU registers and they need much more particles per grid cell to fully benefit from the vectorization. In our application, we could not test the program with much more than 10

million particles due to the limited memory capacity.

In all cases shown in Figure 3.10, the unconnected data points at `nclones = 16` for the CPU and `nclones = 60` for the MIC show the timings of the `PIC_ENGINE` compiled as pure MPI, i.e. without OpenMP. As expected, the first three methods have the same timings because they rely on the same algorithm, only the OpenMP parallelization is different. Surprisingly, the timings of pure MPI are between 30% and 90% (!) lower than the runs with one OpenMP thread. This is explained by the OpenMP overhead introduced to manage the threads and indicates that it is better to compile the code without OpenMP if we want to use it for a pure MPI run.

The timings of the MICs are on average more than five times higher than CPUs. This is mainly because the MIC heavily relies on vectorization and in our case, there is either no vectorization or too few particles per grid cell to benefit from it.

#### 3.3.2.4 Comparison of the different HPC platforms

In the previous sections, we have identified the optimal methods for the data structure, sorting, and charge deposition. We found that the SOA data structure with `psort_2` and `setrho_1` were the most efficient methods on CPU while `psort_2` and `setrho_2` were the most efficient on MIC. In the following, we study the overall timings of the `PIC_ENGINE` on Helios and then compare the results with the other machines, Piz Daint and Piz Dora.

In Figure 3.12, we show the timings of the different components of the `PIC_ENGINE` for the unsorted, and sorted without and with vectorization cases on a single Helios CPU node.

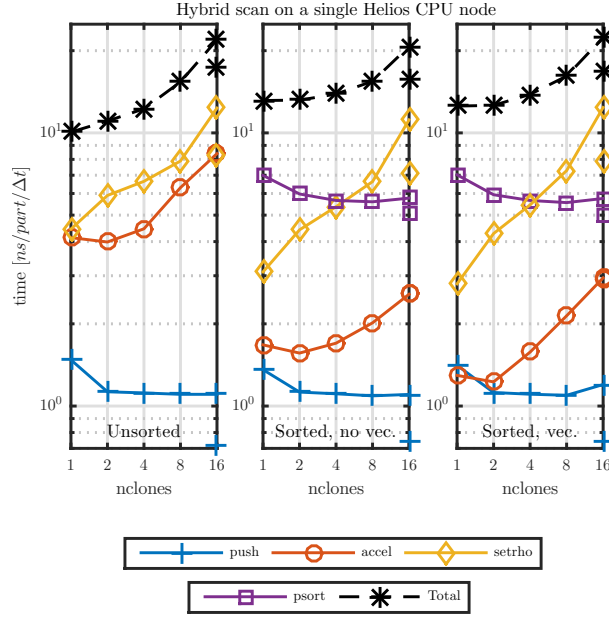
The hybrid implementation using OpenMP allows one to reduce the total timing by a factor of around 2 in the unsorted case. Indeed, with a hybrid approach, fewer data replications are made among the clones which improves the memory usage.

When the sorting is activated, the timings of both `accel` and `setrho` are reduced by a factor of 3.25 and 1.44, respectively, due to the increased data locality in the memory. However, we note that the total timing is effectively slower than the unsorted case as a result of the sorting cost. It has to be noted that different approaches or sorting algorithms may lead to a better performance but they are not tested here. For example, particles that have not moved enough to change grid cell do not need to be sorted. Also, sorting may not be needed at each timestep.

With vectorization, the timings of both `accel` and `setrho` are further decreased by  $\sim 25\%$  and  $\sim 8\%$  respectively for `nclones < 8` and are increased as we approach the pure MPI runs. This loss of performance is explained by the small particle density per grid cell that makes the vectorized loops too short to be efficient.

The unconnected data points located at `nclones = 16` represent the timing of the `PIC_ENGINE` when compiled without OpenMP. As for the charge deposition, it is seen that OpenMP, even with a unique thread introduces a lot of overhead in this simple application as the program runs around 30% faster without OpenMP activated.

### 3.3. The Pic\_Engine: a simple test bed for algorithmic testing



**Figure 3.12 :** Complete hybrid scan on a single Helios CPU node using the most efficient methods, namely `psort_2` and `setrho_1`. The unconnected points at `nclones = 16` show the timing of the pure MPI version of the code compiled without OpenMP.

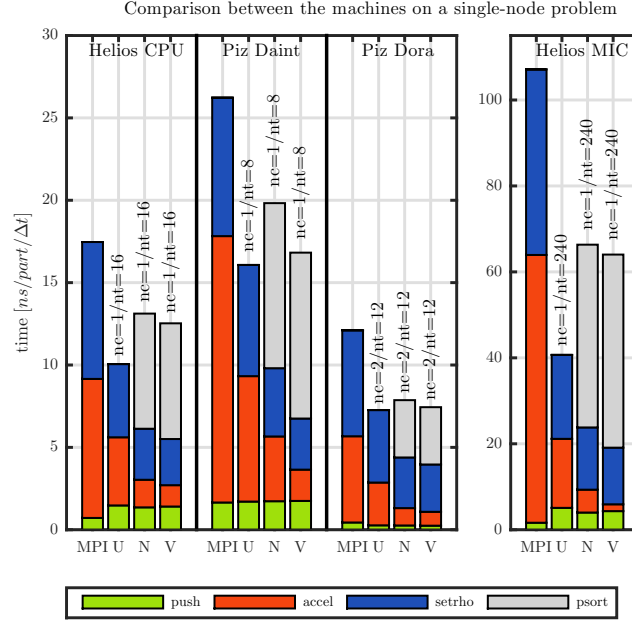
In Figure 3.13, a comparison is shown between Helios CPU and MIC nodes, and the CSCS machines Piz Daint and Piz Dora. The timings are obtained as follows. For each case, a hybrid scan similar to Figure 3.12 is made and the minimum total timing is retained. For each machine, four cases are presented. First, the timing of the `PIC_ENGINE` made in the pure MPI version of the code compiled without OpenMP (MPI) is shown. Then, the usual unsorted (U) and sorted without (N) and with vectorization (V) timings are presented.

As we will see shortly, the conclusions are the same for all the machines but differ quantitatively as the architectures and compilers are different.

The hybrid unsorted version of the code is very beneficial for the performance with a time gain of at least 60% compared to the pure MPI due to the reduced memory contention. Furthermore, the sorting and vectorization add a significant performance gain on both `accel` and `setrho` on a single node.

Note that except for Piz Dora, the optimum timings are obtained for a pure OpenMP run with one clone. For Piz Dora, the optimal configuration is found for two clones and twelve threads which corresponds to one clone per socket.

Finally, the performance of the MIC is surprisingly poor compared to the CPU timings,  $\sim 5$  times slower for our simple application. In fact, we have shown that for an increased particle density per grid cell, the MIC performance is enhanced. Indeed, the main advantage of the MICs is their improved vectorization with 512-bits vector registers but it is bound in



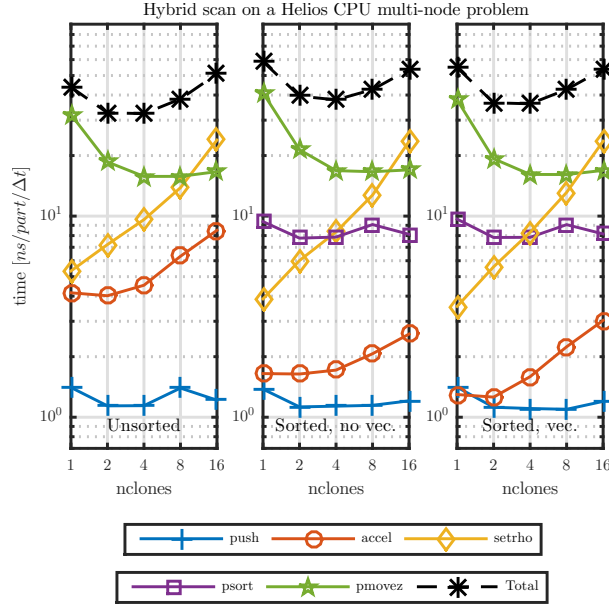
**Figure 3.13 :** Comparison of the performance on the different computers considered in this study. For each machine, the best timing of the MPI version and the unsorted and sorted with and without vectorization cases are shown, denoted respectively by MPI, U, N and V in the graph. Above each hybrid result, the number of clones and threads used to obtain this timing is shown. Note that for the MIC, hyperthreading has been used with four threads per core.

our application by the small number of particles per grid cell. Note that for these results, the optimal performance was found for a pure OpenMP run with 4 (!) hardware threads per physical core.

**Multi-node study** Single-node runs as presented previously are a useful way to provide insight into the program performance. However, for production runs, we generally need to use several nodes to partition the large problem size and number of particles across the cluster nodes. For this reason, in this section, we present multi-node timings of the PIC\_ENGINE. Specifically, we will be able to measure the timings of the inter-node communications that are present in `setrho` to transfer the guard cells between subdomains and in `pmovez` that transfers the particles between the subdomains.

The results presented in this section are made on 16 nodes of the different machines. To be able to compare them with the single-node results, the cases tested here correspond to 16 copies of the single-node problem. In other words, we evolve  $16 \times 10^6$  particles on a  $512 \times 256 \times 16$  grid such that each node has the equivalent workload of a single-node problem. Furthermore, the timings are again given in *ns* per particle and per timestep normalized to one node, i.e. for one million particles.

### 3.3. The Pic\_Engine: a simple test bed for algorithmic testing

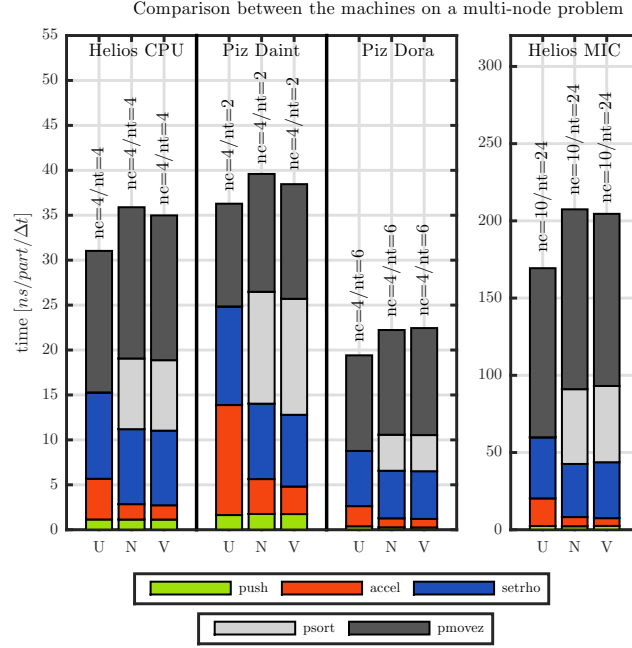


**Figure 3.14 :** Hybrid scan on 16 Helios CPU nodes using the most efficient methods, namely `psort_2` and `setrho_1`. The single points show the timing of the pure MPI version of the code compiled without OpenMP.

In Figure 3.14, we show the multi-node hybrid scan on the Helios computer for the unsorted, and sorted without and with vectorization cases. The timings are qualitatively very similar to the single-node runs. The main difference comes from the `pmovez` routine that is now activated and that represents around half of the total timing. In our case, less than 1% of the particles per subdomain are moved. Thus, the communication time is negligible. However, the `pmovez` routine spends a lot of time preparing the communication by, for example, determining the particles that should be moved, which represents, in this case, most of the `pmovez` timing. Similar to the sorting method, this is in fact not an issue for some application codes with a higher physical complexity, as the relative cost of `pmovez` will be significantly reduced. Indeed, it has been found in ORB5, for example, that it does not represent more than 10% of the total timing for all production runs made so far.

In Figure 3.15 are shown the timings of the multi-node hybrid runs on all the machines. We can see that the `pmovez` method dominates the timings in all cases. This conclusion is even enhanced for the MIC that suffers from a poor communication network as the messages must first pass through the CPU before being sent to the recipient MIC. Again, let us emphasize that only the improvements of `setrho` and `accel` are relevant as they are the most time-consuming methods of PIC codes such as ORB5.

The single and multi-node results are compared in Table 3.4. As expected, the timings of `push` and `accel` are the same for single and multi-node cases.



**Figure 3.15 :** Comparison of the performance of a 16 nodes run on the different computers considered in this study for a multi-node simulation. For each machine, the unsorted (U) and sorted without (N) and with (V) vectorization cases are shown. Above each hybrid result, the number of clones and threads used to obtain this timing is shown. Note that for the MIC, hyperthreading has been used with four threads per core.

The timings of `setrho` are slightly higher in the multi-node case because we need MPI communications between the subdomains to transfer the guard cells. This represents around 20% of the total `setrho` timing.

Surprisingly, the sorting method is slower for the multi-node runs. This is however explained by the fact that in multi-node simulations, the particles travel through the subdomains in the  $z$  direction and their positions are not sorted in the  $(x, y)$  plane in the recipient subdomain.

### 3.4 Application to the ORB5 code: the color scheme

The `PIC_ENGINE` is a very useful tool that has allowed us to develop new algorithms and test different parallelization strategies before implementing them in ORB5. Starting with a simplified PIC code yet implementing similar features of ORB5, e.g. parallelization and data structures, we were able to easily develop various numerical schemes without the tedious task of implementing them in a large production code. Using the knowledge acquired from the `PIC_ENGINE`, we have then implemented the most promising algorithms in ORB5. One of the most critical parts of the PIC scheme is the charge/current deposition routine because

### 3.4. Application to the ORB5 code: the color scheme

**Table 3.4 :** Comparison of the performance for the single- and multi-node simulations. The timings shown here correspond to pure OpenMP runs and are presented in ns/particle/ $\Delta t$  normalized to the single-node problem to compare the results.

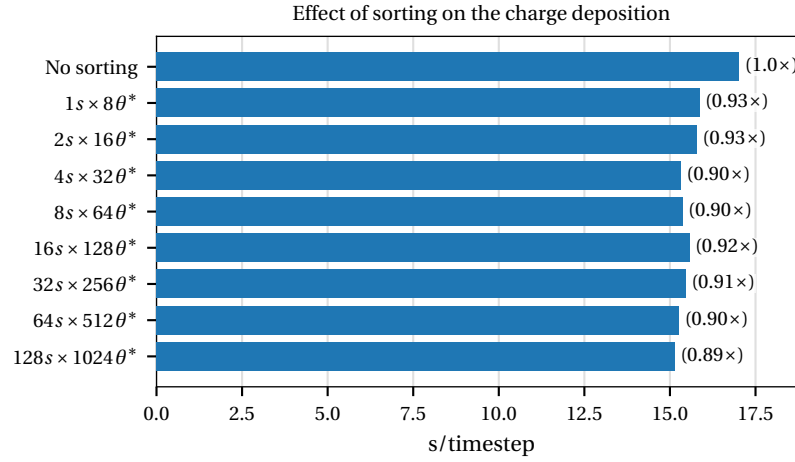
Nodes	Helios CPU		Piz Daint		Piz Dora		Helios MIC	
	Single	Multi	Single	Multi	Single	Multi	Single	Multi
push	1.41	1.37	1.75	1.76	0.73	0.73	4.33	5.01
accel	1.30	1.30	1.90	1.93	0.81	0.81	1.59	1.69
setrho	2.80	3.56	3.10	3.54	2.76	3.28	13.18	15.93
psort	7.02	9.66	10.07	12.12	4.65	5.43	44.93	47.65

it interpolates data from hundreds of millions of particles onto the field grid. The way it was originally implemented in ORB5 essentially consisted of indirect memory addressing, which is known to lead to poor performance. In this section, we study the performance of the charge/current deposition routine of ORB5 after having implemented the sorting and various other schemes developed using the PIC\_ENGINE. The other parts of the code have also been optimized, but they are not presented in this work.

All the results presented in this section are produced using a typical nonlinear TEM simulation which is scaled down to a single-node problem. The main relevant parameters for this study are the following. The timestep is set to  $\Delta t = 2 [\Omega_{ci}^{-1}]$  and the resolution of the field grid is  $N_s \times N_{\theta^*} \times N_\varphi = 128 \times 1024 \times 1$  with second order B-splines. The number of markers is  $4 \cdot 10^6$  for each of the two species (ions and electrons) but it has to be noted that 4 points were used for the gyroaveraging, effectively leading to  $16 \cdot 10^6$  gyropoints to be treated in the deposition step for the ions. The timings are done on the upgraded XC40 compute nodes of the Piz Daint machine at CSCS, which are equipped with two 18-core Intel Xeon E5-2695 v4 with a clock frequency of 2.10GHz.

#### 3.4.1 Effect of sorting on the charge deposition step

As we have already seen with the PIC\_ENGINE in Section 3.3.2.3, sorting the particles with respect to their position allows for a better performance due to an improved data locality. In Figure 3.16, the timings of the former ORB5 charge deposition, i.e. with loop over particles and no OpenMP, are presented for various sorting granularities and with the MPI version of the code. For reference, the case without sorting is also shown. As the granularity of sorting is decreased, the timings of the charge deposition step are reduced with a maximal timing reduction of 11% with a full sorting, i.e. 128 buckets in the  $s$  direction and 1024 buckets in the  $\theta^*$  direction. This timing reduction is due to a better cache usage since the L1 cache misses are halved in this case. Note that, as for the PIC\_ENGINE, the sorting has a non negligible timing which is not compensated by the time gain on the charge deposition step only. Further optimizations on the sorting algorithm are possible but not considered in this work.

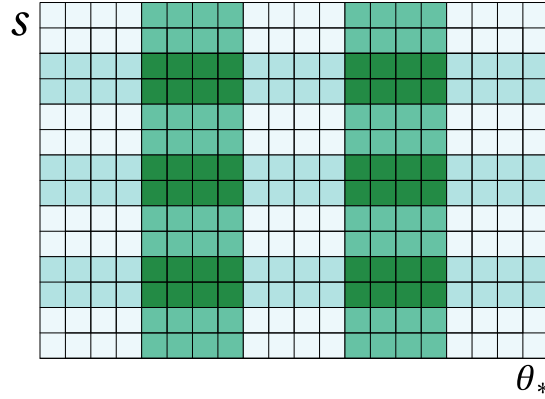


**Figure 3.16 :** Influence of sorting on the charge deposition routine. The scan goes from no sorting to a full sort ( $128s \times 1024\theta^*$ ) with a single MPI task and no OpenMP threads. The labels on the y axis represent how many buckets are used for the sorting in the  $s$  and  $\theta^*$  directions.

#### 3.4.2 OpenMP performance and the color scheme

Let us now consider the inclusion of the third level of parallelism using OpenMP. To this end, three charge deposition algorithms are considered. The first two are the `setrho_1` and `setrho_4` routines developed using the `PIC_ENGINE`, Section 3.3.2, and which consist of looping over the particles with manual data replication for the former and looping over the grid points, thus taking advantage of the full sorting, for the second. Note, however, that in the ORB5 case, the cloud-in-cell deposition is replaced by a second order B-spline deposition. The third charge deposition method, inspired from [105], is called the color scheme. The basic idea of this algorithm is to partition the grid into four sets of sub-grids in such a way that all the sub-grids from each sets are totally disjoint, Figure 3.17. For a better visualization, a different color is attributed to each set of sub-grids. Note that since the deposition is considered to be done within a single poloidal plane per subdomain, which is the usual setup for an ORB5 production simulation, only four colors are needed. This number would be increased to eight for a 3D version of this algorithm. To avoid race conditions, the charge deposition step is then performed as follows. A first loop over the four colors is made. Then, all the sub-grids and corresponding sorted particles of the current color are shared among OpenMP threads for charge deposition. The main advantage of the color scheme is that it is a collision free algorithm by construction. It does not need to use OpenMP `atomic` directives which were leading to a poor performance of the `PIC_ENGINE`. Furthermore, a small or no data replication nor data reduction are needed. The main disadvantage is that it requires a sorting of the particles, which has a certain cost.



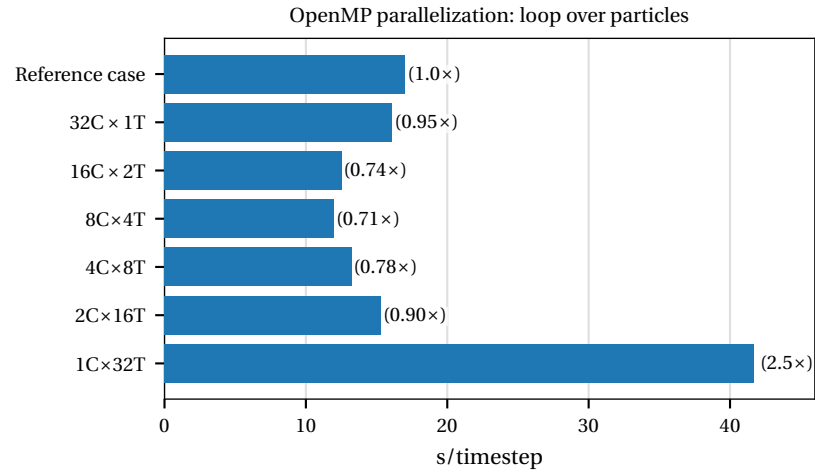


**Figure 3.17 :** Illustration of the color scheme. The 2D poloidal grid is divided into four sets of sub-grids and each set is attributed a color in such a way that all the sub-grids of each set are totally distinct.

In all the following tests, full sorting is always activated, i.e. we use 128 buckets in the  $s$  direction and 1024 buckets in the  $\theta^*$  direction. Furthermore, all the timings are compared with a reference case which is the former ORB5 charge deposition routine without sorting and with no OpenMP threads. Then, similarly to the PIC\_ENGINE tests, a scan is made in the number of MPI clones and OpenMP threads in such a way that  $n_{\text{clones}} \times n_{\text{threads}} = N_{\text{cores}} = 32$ , which ensures that all the resources are used. All the timings are done on the XC40 partition of the Piz Daint machine at CSCS.

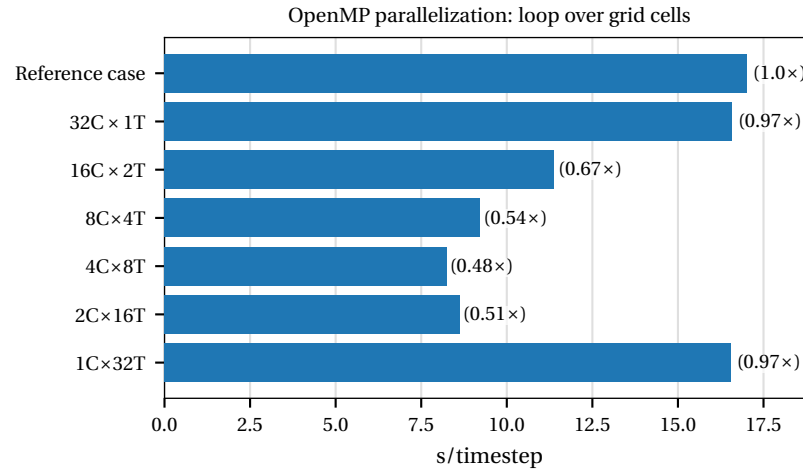
First, Figure 3.18 shows the results for the `setrho_1` routine, which loops over the particles and uses a manual data reduction. The timings decrease as the number of threads is increased with an optimal configuration using 8 clones and 4 OpenMP threads, which leads to a performance improvement of  $\sim 25\%$  as compared to the reference case. As the number of OpenMP threads is further increased, a tendency reversal is observed and the performance worsens. The pure OpenMP case is surprisingly 2.5 times slower than the reference case. This is mainly due to the allocatable arrays used for the data replication that are allocated/de-allocated at every time step. Another important factor which alters the performance of the pure OpenMP case is that the compute nodes used for this study have two CPU sockets. With only one clone, threads on one CPU may have access to the memory of the other CPU. Note that the case with 32 clones and 1 OpenMP thread is actually similar to the case with full sorting presented in Figure 3.16, which showed a performance improvement of  $\sim 10\%$  compared to the reference case. The only difference is the unique OpenMP thread. As already observed using the PIC\_ENGINE, the difference in timing is essentially due to the overhead brought by OpenMP. It is however less important in this case since it only represents a slowdown of  $\sim 5\%$  while it was  $\sim 30\%$  in the PIC\_ENGINE case.

In Figure 3.19, the timings for the `setrho_4` method, which loops over the grid cells are presented. As for the first case, the performance improves as the number of OpenMP threads



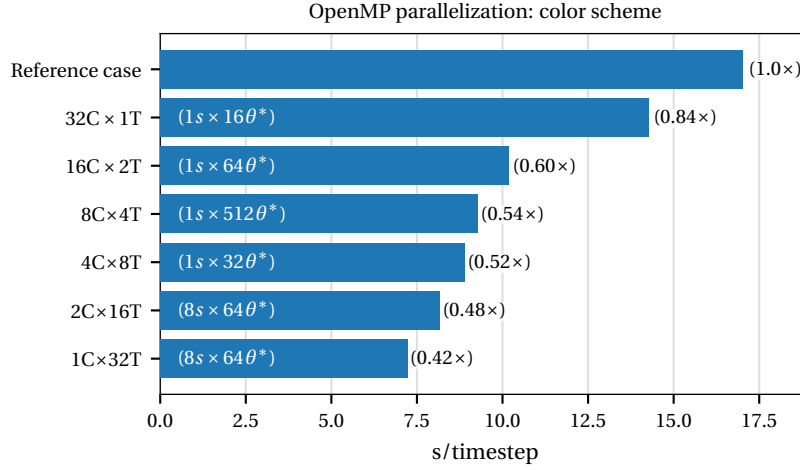
**Figure 3.18 :** Timings for charge deposition with loop over particles. Hybrid scan clones versus threads for the charge deposition with loop over the particles. The reference case corresponds to a pure MPI run without sorting. The labels of the y axis represent how many clones (C) and threads (T) are used.

is increased up to an optimal configuration with 4 clones and 8 threads which gives a ~50% of improvement with respect to the reference case. After this optimal setup, the timings increase. Once again, the poor performance of the pure OpenMP case is attributed to the two-socket compute node used.



**Figure 3.19 :** Same as Figure 3.18 but with loop over the grid cells.

Finally, the results for the color scheme are presented in Figure 3.20. For each of the  $N_{\text{clones}} \times N_{\text{threads}}$  configuration, a scan is made in the sub-grid size for the color scheme decomposition and only the optimal value is retained. Contrarily to the previous cases, the



**Figure 3.20** : Same as Figure 3.18 but using the color scheme.

performance of the color scheme increases as the number of OpenMP threads is increased up to its maximum leading to a ~60% timing reduction compared to the reference case.

### 3.5 Summary of the chapter

In this chapter, all the numerical aspects of ORB5 have been presented. We first discussed the current numerical implementation of the ORB5 code followed by a short summary of the refactoring process as well as the optimization techniques used to optimize the code.

With the goal of porting the ORB5 code to the new multi- and manycore HPC platforms, we have developed and tested different optimization techniques and parallelization schemes. For this purpose, we have developed a test bed called the `PIC_ENGINE` serving as a simple and portable abstraction of the PIC algorithm and allowing us to easily test these techniques on a simplified yet realistic code. To that end, only the key elements common to all PIC applications are retained.

In this work, we have considered a 6D Vlasov equation describing the evolution of a single-species plasma in the electrostatic limit in Cartesian coordinates. For the sake of simplicity, no magnetic field is considered and no field solver is used. The charge depositions and grid-to-particle interpolations are done using, respectively, a cloud-in-cell method and a linear interpolation on the electrostatic potential. The time integration scheme evolving the particle equations of motion is based on the second order leap-frog method.

In order to preserve the portability of the program, it has been coded using Fortran and a hybrid implementation of MPI and OpenMP for the parallelization. By doing so, the code can be run on most modern HPC clusters including MIC-equipped computers.

We have presented different possible optimizations, which have been tested on a single-node problem in order to apprehend the shared memory programming model and the inherent difficulties.

We have first discussed the data structure used in the `PIC_ENGINE` to store particle quantities. Both structure of arrays and array of structures have been considered. Since most of the memory access is made following the particle order, it is found that the structure of arrays is more efficient because data is accessed contiguously—recall that Fortran uses a column-major order to store arrays in memory. A performance gain up to 14.4% is observed in this mode.

In most of the PIC codes, particles are stored randomly in computer memory. This is critical for the program performance since random memory accesses are very inefficient. To avoid this problem, we have implemented a counting sort algorithm to sort the particles according to their positions. Despite its cost, the sorting allows increasing the performance of the most time-consuming methods, namely `accl` and `setrho`, by respectively factors of 3.25 and 1.44. Furthermore, the particle sorting also allows for the vectorization of these methods, which is otherwise not possible due to their indirect addressing. A performance gain up to 25% is observed thanks to vectorization for the case considered, with an average of eight particles per grid cell. Furthermore, we have shown that with an increased number of particles per grid cell, the vectorization performance further increases. For the CPU, the gain is up to 30% while it is around 10% for the MIC. The lower gain increase for the MIC compared to the CPU is explained by the poor vectorization due to the small number of particles per grid cell and the wide 512-bits vector registers used by the MIC that can vectorize up to eight doubles compared to four for the CPU.

The `setrho` method is a challenging candidate for the parallelization because of the race conditions. In the `PIC_ENGINE`, we have implemented different parallelization methods based either on an atomic update of the memory or data safe algorithms that completely avoid these race conditions.

The use of a shared memory programming language allows for a reduction of the memory contention problem by limiting data replication thus improving the performance. Indeed, it is observed that the pure OpenMP version of the code runs up to 2.15 times faster than the pure MPI version.

Multi-node experiments were done to test the `PIC_ENGINE` in more realistic conditions and to time the MPI communications. The total multi-node timings are almost twice as slower than the single-node results mainly because of the `pmovez` method used to transfer the particles across the subdomains. Indeed, for more complex applications, the timings of `psort` and `pmovez` will remain the same in absolute value, but the timings of the other methods will be much higher due to the more complex physics and/or higher order schemes. Therefore,

the relative timings of `psort` and `pmovez` is expected to be much smaller in such applications.

We have been able to test the performance of the `PIC_ENGINE` on the MIC architectures. This is possible because the same programming model is used for both CPUs and MICs in native mode. Unfortunately, the MIC performance is poor compared to the CPU, which we interpret as resulting from poor vectorization. However, a recently published study [115] shows that with the appropriate low-level optimizations, the Intel MIC can be up to 1.6 times faster than an Intel eight-core CPU. In a future work, we will consider their approach and try to apply it to the `PIC_ENGINE`.

Finally, the knowledge acquired from the `PIC_ENGINE` has allowed us to implement optimized versions of the ORB5 main parts. As an example, the charge deposition method optimization of ORB5 has been presented. First, the effect of the sorting on the original deposition method has been studied. It is shown that using a full sorting increases the performance by  $\sim 10\%$ . Then, three different OpenMP parallelization algorithms of the charge deposition are tested. Usually, it is observed that the optimal performance depends on the clone/thread configuration and no general rules can be drawn. The color scheme, which is the most efficient algorithm, leads to a timing improvement of  $\sim 60\%$ . The main drawback of this scheme is that it requires various testings to find the optimal number of clones/threads and the correct grid sub-division.



## 4 Improved hybrid electron model

### 4.1 Introduction

Magnetically confined fusion plasmas often exhibit large temperature and density gradients which are the source of free energy for a multitude of drift-wave-type micro-instabilities that span over different spatial and temporal scales. For low- $\beta$  plasmas, anomalous transport of particle, heat, and momentum is attributed to the turbulence caused by ITG, TEM, and ETG modes. The ITG-driven instability, which is responsible for the anomalous ion transport, has been extensively studied using gyrokinetic simulations since the pioneering work of Lee and Parker [116, 117] in the late 1980s. In first approximation, an adiabatic electron model is sufficient to describe these modes. If a kinetic electron response is included, the TEM and ETG modes, which are responsible for the anomalous electron heat and particle transport, can also be described. While a partial kinetic electron response including only the trapped electrons allows one to simulate TEM, the ETG instability requires a fully kinetic electron model. At higher  $\beta$  values, the ITG perturbations are stabilized and electromagnetic modes such as the kinetic ballooning mode (KBM) [118], micro-tearing [119], and Alfvén ITG modes [120] are destabilized. In addition to those drift-wave-type micro-instabilities, the kinetic electron model is also needed to study internal kink modes [121], and interaction between turbulence and energetic particles [94].

The main drawback of the full kinetic electron model is the computational cost that is required to solve the fast electron parallel dynamics. For example, the so-called  $\omega_H$  mode [116], which is the electrostatic limit of the kinetic Alfvén wave, severely limits the simulation timestep as its frequency,  $\omega_H = (k_{\parallel}/k_{\perp})\sqrt{m_i/m_e}\Omega_{ci}$ , becomes comparable to the ion cyclotron frequency at long perpendicular wavelengths. Note also that in this limit, the  $\omega_H$  mode is not consistent with the gyrokinetic ordering presented in Section 2.2, for which the ratio  $\omega/\Omega_{ci}$  is considered to be small, where  $\omega$  is the typical frequency of the mode of interest. Two approaches can be used to avoid the  $\omega_H$  mode while still being able to simulate TEM. A finite- $\beta$  value can be used as it quickly decreases the frequency of the shear Alfvén wave [122]. However, this requires a code able to simulate electromagnetic fluctuations and usually these are more challenging runs to carry out in terms of numerical resources. The other possibility is

to use a hybrid electron model in which trapped electrons have a kinetic response and passing electrons are adiabatic. This was the former hybrid electron model implemented in ORB5 [41]. Although this model is satisfactory in the linear and collisionless limit, major issues are observed in the nonlinear regime and/or in the presence of collisions. Indeed, the ambipolar condition is not satisfied as no contribution from the passing electrons is accounted for in the flux-surface-averaged gyrokinetic Poisson equation, which also leads to a non-conservation of the canonical toroidal angular momentum. This is because the trapping/detrapping processes, which can only occur because of the parallel nonlinearity, collisions, and radial drifts, are not well accounted for<sup>1</sup>. In the above-mentioned hybrid model, particles that are (de)trapped are suddenly treated as (adiabatic) kinetic. This generates a spurious source of density, energy and momentum. Other hybrid electron models using bounce-averaged trapped electrons [123] or fluid-kinetic electrons [124] have been developed but they suffer from the same problems.

More recently, an improvement to the kinetic trapped hybrid electron model has been proposed [56]. It allows one to solve the major issues of the previously described model while avoiding the  $\omega_H$  mode. In this corrected model, the gyrokinetic Poisson equation is first decomposed into non-axis and axisymmetric toroidal Fourier modes,  $n \neq 0$  and  $n = 0$ . The non axisymmetric part is similar to the former hybrid model in ORB5, i.e. trapped electrons are kinetic while passing electrons have an adiabatic response. For the  $n = 0$  part, there is no distinction between passing and trapped electrons in the Poisson equation; they both contribute kinetically. Once the electrostatic potential is obtained by solving the quasi-neutrality equation, a filter is applied to the axisymmetric part of the potential to keep only the poloidal Fourier mode  $m = 0$ . This is done to eliminate the  $\omega_H$  mode, since finite  $k_{\parallel}$  contributions are filtered out. This correction allows the model to ensure the ambipolarity condition, the conservation of the toroidal angular momentum, and the trapped/passing boundary is well accounted for. However, because of the filtering of the  $n = 0$  component of the total electrostatic potential, the GAM frequency is changed, which can be problematic for comparison with experiments [125, 9].

In this work, we propose a further enhancement to the model proposed in [56], which allows one to correctly capture the GAM frequency. The main idea behind this upgraded hybrid electron model is to account only for a flux-surface-averaged (FSA) kinetic contribution of the passing electrons to the quasi-neutrality equation. The non flux-surface-averaged contribution is assumed to have an adiabatic response. For this reason, we will refer to this model as the “FSA” hybrid electron model. The weak form of the electron contribution to the gyrokinetic Poisson equation is given by

$$\epsilon_{\delta} \int dV \alpha_P n_{e0} \frac{e}{T_e} (\phi_1 - \bar{\phi}_1) \hat{\phi}_1 + \int dV n_{e0} \hat{\phi}_1 + \int_{\text{trapped}} d\Omega f_e \hat{\phi}_1 + \int_{\text{passing}} d\Omega f_e \bar{\hat{\phi}}_1, \quad (4.1)$$

---

<sup>1</sup>Note that in collisionless flux-tube codes, the parallel nonlinearity is generally neglected and background profiles are assumed to be constant. Within these approximations, the trapped/passing hybrid electron model is correct



where  $\overline{\hat{\phi}_1}$  is the flux-surface average of the test function  $\hat{\phi}_1$ . The first term represents the adiabatic contribution of the passing electron fraction  $\alpha_P$  given by

$$\alpha_P(\psi, \theta^*) = 1 - \frac{\int_{\text{trapped}} d^3\nu}{\int d^3\nu} = 1 - \sqrt{1 - \frac{B(\psi, \theta^*)}{B_{\text{max}}(\psi)}}. \quad (4.2)$$

The second term is the background contribution, which is assumed to cancel with the ion background contribution, the third term is the trapped electron gyro density, and the last term is the flux-surface-averaged contribution of the passing electrons. Note that the “trapped” and “passing” subscripts mean that the integrals in the velocity space are done over the trapped and passing portions of phase space respectively, which are defined using Eq. (1.14).

As a first approximation, the flux-surface-averaged kinetic contribution of the passing electrons can be computed by filtering out all  $m \neq 0$  kinetic contributions of the passing electrons to the  $n = 0$  mode, i.e. approximating the FSA by taking only the  $n = m = 0$  component into account. This can be written as

$$\epsilon_\delta \int dV \alpha_P n_{e0} \frac{e}{T_e} (\phi_1 - \overline{\phi}_1) \hat{\phi}_1 + \int dV n_{e0} \hat{\phi}_1 + \int_{\text{trapped}} d\Omega f_e \hat{\phi}_1 + \int_{\text{passing}} d\Omega f_e \hat{\phi}_1^{00}, \quad (4.3)$$

where  $\hat{\phi}_1^{00}$  is the  $n = m = 0$  component of  $\hat{\phi}_1$ . Note that this approximation is exact if the Jacobian  $J_{s\theta^*\varphi}$ , related to the spatial coordinates, depends only on the radial coordinate which corresponds to a cylindrical system.

The main difference between the correction presented in [56] and both the FSA and its  $n = m = 0$  approximation is that in the former, the axisymmetric part of the whole self consistent electrostatic field  $\phi$  is filtered to retain only the  $m = 0$  mode, thus also affecting the ion response. In the FSA and  $n = m = 0$  models, the filtering is done only on the axisymmetric contribution of the passing electrons without modifying the ion or trapped electron response. By doing so, the GAM properties are not affected.

the next section of this chapter, the validation of the upgraded hybrid electron model is presented. It is done using a Rosenbluth-Hinton (RH) test [43] to validate the zonal properties, i.e. the GAM frequency and the convergence toward the correct zonal flow residual level. Finally, the ambipolar condition is verified using nonlinear simulations. In Section 4.3, a nonlinear simulation of the TCV tokamak showing interesting features is presented.

## 4.2 Validation of the upgraded hybrid electron model

### 4.2.1 Linear validation: Rosenbluth-Hinton test

The setup used for the RH test presented in this section is mainly inspired from [56] which is a modified CYCLONE-like setup [29]. It consists of a deuterium plasma in a circular concentric magnetic equilibrium with  $\rho_* = 1/150$ ,  $R_0/a = 2.79$ , a quadratic safety factor profile  $q(\rho/a) = 0.85 + 2.18(\rho/a)^2$ , and a magnetic field amplitude at the magnetic axis of  $B_0 = 1.91\text{T}$ . The

## Chapter 4. Improved hybrid electron model

**Table 4.1** : Numerical parameters used for the RH test with different electron models. From top to bottom: adiabatic electrons, the former ORB5 hybrid electron model not ensuring ambipolarity, the correction proposed in [56], the upgraded model with the flux-surface average passing kinetic contribution, its  $n = m = 0$  approximation, and the fully kinetic electron model with a small  $\beta = 10^{-4}$  value.

	Grid resolution ( $N_s \times N_{\theta^*} \times N_{\varphi}$ )	Timestep [ $\Omega_{ci}^{-1}$ ]	Number of markers [ $\cdot 10^6$ ]
Adiabatic	$180 \times 256 \times 32$	30	10 (ions only)
Former	$180 \times 256 \times 32$	5	10 (both species)
Idomura2016	$180 \times 256 \times 32$	5	10 (both species)
FSA	$180 \times 256 \times 32$	5	10 (both species)
$n = m = 0$	$180 \times 256 \times 32$	5	10 (both species)
Kinetic	$180 \times 256 \times 64$	0.5	10 (both species)

surface of reference is located at  $\rho/a = 0.5$ , both the density and temperature profiles are chosen to be flat over the whole radial domain  $\rho/a \in [0, 1]$ , and the ion to electron mass ratio is  $m_i/m_e = 100$ . For the RH test, an initial perturbation on the deuterium weights is applied while the electron weights are initialized to zero. The radially-dependent perturbation is of the form

$$\delta f(\rho/a) = A \cos(\pi k_r \rho/a) f_0(\rho/a), \quad (4.4)$$

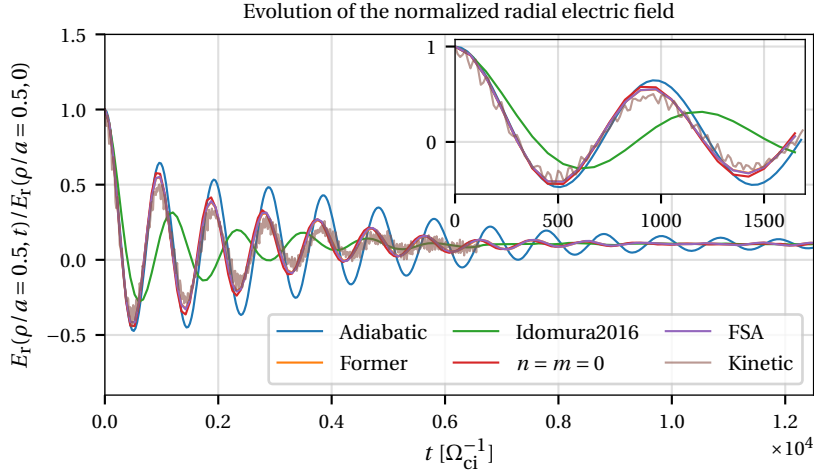
where  $A = 10^{-5}$  is the amplitude and  $k_r = 1$  is the radial wave number harmonic. This perturbation is such that the radial electric field  $E_r$  is maximum at  $\rho/a = 0.5$ . Only the axisymmetric modes, for which  $n = 0$ , are retained in the Fourier filter of the solver. Various electron models are compared in this validation. Their numerical parameters are summarized in Table 4.1. Note that the grid resolution and number of particles are set for the most restrictive case and they are used for all the electron models (except for the kinetic electrons), even though they could be reduced in some cases. The benefit of using the hybrid model is illustrated here, as it allows one to increase the timestep by a factor of 10 compared to the kinetic electron model, which in this case actually uses a small  $\beta$  value to decrease the  $\omega_H$  mode frequency. A similar ratio is also observed for the nonlinear simulations presented in the following sub-sections.

The results of the RH test are shown in Figure 4.1. The radial electric field defined by

$$E_r(\rho/a, t) = -\frac{\partial \bar{\phi}(\rho/a, t)}{\partial(\rho/a)}, \quad (4.5)$$

is normalized to its initial value, i.e.  $E_r(\rho/a, t)/E_r(\rho/a, 0)$  and plotted as a function of time at the center of the radial domain,  $\rho/a = 0.5$ , where  $\rho$  and  $a$  are respectively a geometric radial coordinate and the minor radius. The figure shows an initial, oscillatory phase corresponding

## 4.2. Validation of the upgraded hybrid electron model



**Figure 4.1** : Rosenbluth-Hinton test realized for different electron models. The normalized radial electric field at  $\rho/a = 0.5$  is plotted against time. The subplot in the top-right corner is a zoom of the time interval  $t\Omega_{ci} \in [0, 1500]$

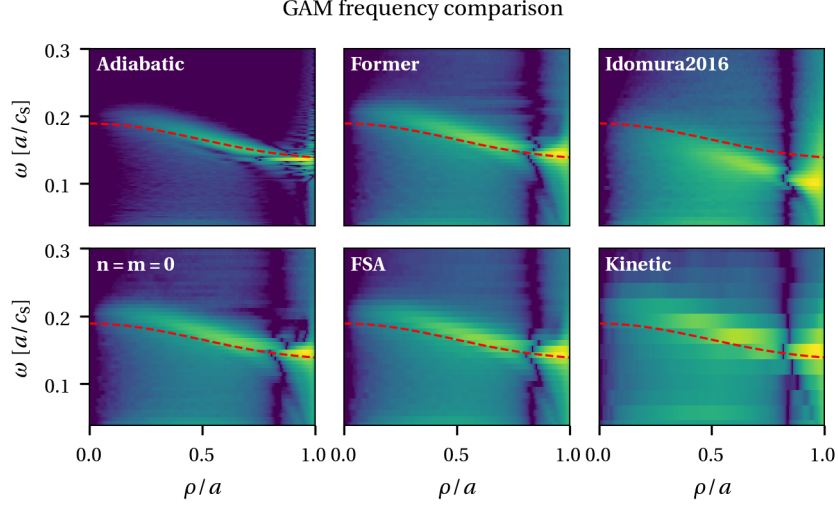
to the GAM followed by a saturated phase corresponding to the ZF residual. All the simulations converge toward the same residual, which is non-zero in the collisionless limit. The GAM frequencies and damping rates differ however between the different models. The adiabatic electron model has the lowest damping rate. The former ORB5 hybrid model, the FSA model and its approximation  $n = m = 0$ , and the fully kinetic electrons have a similar damping rate. Finally, the hybrid model from [56] has the highest damping rate. Note that the damping rate from the adiabatic electron model can be recovered if the real ion to electron mass ratio is used.

The GAM frequency is computed from a Fourier transform in time of the radial electric field  $E_r$  defined by Eq. (4.5). The GAM spectrum as a function of the radial position is shown in Figure 4.2. An analytical estimate of the frequency,  $\omega_{ana}$  [45], is also plotted in dashed red line, where

$$\omega_{ana}^2 = \frac{5 - \kappa}{4} \left[ \omega_{GAM,0}^2 + \frac{1}{q^2 \omega_{GAM,0}^2} \left( \frac{23}{2} + 8\tau_e + 2\tau_e^2 \right) \right] \left( \frac{v_{th,i}}{R} \right)^2, \quad (4.6)$$

$$\omega_{GAM,0}^2 = \frac{7}{2} + 2\tau_e, \quad (4.7)$$

with  $\tau_e = T_e/T_i$ ,  $R$  the major radius,  $v_{th,i}$  the ion thermal velocity, and  $\kappa$  the elongation. Note that another analytical computation of the GAM frequency [46] has been used for comparison and gave very similar estimates as the one presented above. As mentioned in [56], the Idomura2016 model significantly underestimates the GAM frequency, especially for  $\rho/a > 0.5$ .



**Figure 4.2** : GAM spectrum computed from the normalized radial electric field amplitude for the different electron models. The red dashed line is an analytical prediction, Eq. (4.6), of the GAM frequency.

The other models have similar GAM frequency estimates. They are particularly close to the analytical prediction for  $\rho/a > 0.5$ , however they tend to overestimate the frequencies for  $\rho/a < 0.5$ . Both the FSA model and its  $n = m = 0$  approximation show a similar frequency evaluation as the former ORB5 hybrid electron model and as the kinetic electron model.

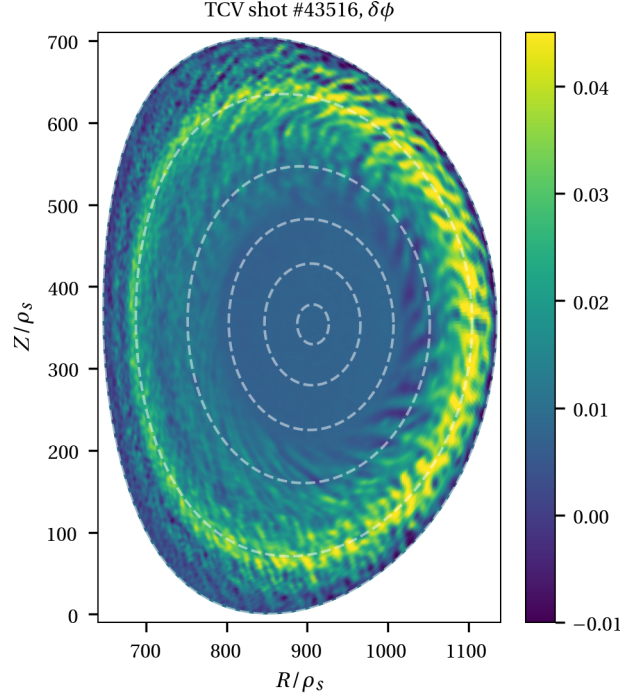
#### 4.2.2 Nonlinear verification

We now turn to the nonlinear verification of the ambipolar condition. The case considered here is a deuterium plasma without impurities and it uses an ideal MHD equilibrium reconstructed from TCV shot #43516 with an aspect ratio of 3.64, an elongation of 1.44, and a triangularity of 0.20, Figure 4.3. The safety factor profile is quadratic,  $q(s) = q_0 + (q_{\text{edge}} - q_0)s^2$ , with  $q_0 = 0.78$  and  $q_{\text{edge}} = 3.29$ . The quantities are normalized with respect to the  $s_0 = 1.0$  reference magnetic surface and ions play the role of reference species for the normalization. It follows that  $\rho_*(s_0) = \rho_i(s_0)/a = 1/245$ —at mid-radius  $\rho_*$  is around  $1/100$ .

The temperature and density profiles are inspired from [49] that found electron profiles exhibiting constant logarithmic gradients in the core region but excluding the sawtooth inversion radius, and a pedestal region—even in L-mode plasmas—with constant gradients. In ORB5, the temperature profile is parametrized as follows, Figure 4.4:

$$T(\rho_V) = \begin{cases} \min\{T_0, T_{\text{ped}} \exp[-\kappa_T(\rho_V - \rho_{\text{ped}})]\} & \text{for } \rho_V \leq \rho_{\text{ped}}, \\ T_1 - \mu_T \cdot (\rho_V - 1) & \text{for } \rho_V > \rho_{\text{ped}}, \end{cases} \quad (4.8)$$

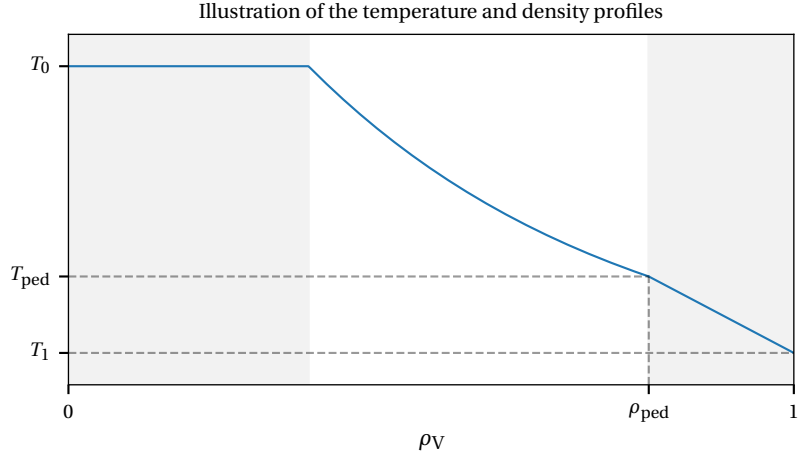
where  $T_{\text{ped}} = T_1 + \mu_T(1 - \rho_{\text{ped}})$ , and the radial variable is defined as  $\rho_V = \sqrt{V(\psi)/V(\psi_{\text{edge}})}$  with



**Figure 4.3 :** Illustration of the TCV equilibrium based on shot #43516. Poloidal cut of the simulated potential fluctuation  $\delta\phi$  during the turbulent phase of the run. The dashed lines represent magnetic surfaces.

$V(\psi)$  being the volume enclosed by the magnetic surface  $\psi = \text{const}$  and  $\psi_{\text{edge}}$  is the value of  $\psi$  at the radial outer edge. The parameters  $T_0$ ,  $T_1$ ,  $\kappa_T$ , and  $\mu_T$  are input parameters defining the profile. A similar definition is used for the density profile with parameters  $n_0$ ,  $n_1$ ,  $\kappa_n$ , and  $\mu_n$ . Note that the profiles defined by Eq. (4.8) are used to qualitatively reproduce the experimental profiles found in [49]. From the point of view of the ORB5 gyrokinetic model, they are not consistent as the instability mechanisms taking place in the sawtooth inversion region are not correctly described.

If not stated otherwise, for all the nonlinear TCV simulations presented in this chapter, the following parameters are used. The temperature and density profiles are defined by  $\rho_{\text{ped}} = 0.8$ ,  $T_0^i = 5$ ,  $T_1^i = 1$ ,  $\kappa_T^i = 2.3$ ,  $\mu_T^i = 6$ ,  $n_0^i = 5$ ,  $n_1^i = 1$ ,  $\mu_n^i = 5$ ,  $\kappa_n^i = 2.3$  and  $T_0^e = 10$ ,  $T_1^e = 1$ ,  $\kappa_T^e = 2.5$ ,  $\mu_T^e = 10$ ,  $n_0^e = 5$ ,  $n_1^e = 1$ ,  $\mu_n^e = 5$ ,  $\kappa_n^e = 2.3$ . Ions and electrons have the same density profile and different temperature profiles except at the radial outer edge where  $T_i(s_0) = T_e(s_0)$ . The ion to electron mass ratio is set to  $m_i/m_e = 200$ . The numerical resolution for these runs is  $N_p = 256 \cdot 10^6$  markers for each species and  $\Delta t = 2 [\Omega_{ci}^{-1}]$ . The grid resolution is  $N_s \times N_{\theta^*} \times N_{\varphi} = 256 \times 512 \times 256$ . The field-aligned Fourier filter is set to keep only modes such



**Figure 4.4 :** Illustration of the temperature and density profiles used in this work. They are composed of three regions. From left to right, the first region mimics the sawtooth inversion region where gradients are zero. Then, a region with constant logarithmic gradients  $\kappa_T$  spans up to  $\rho_{\text{ped}}$ . The outer radial region is the pedestal, characterized by constant gradients defined by  $\mu_T$ .

that  $0 \leq n \leq 64$  and satisfying  $m = nq(s) \pm 5$ . Finally, the modified Krook operator conserving density, parallel momentum and ZF, with a rate  $\gamma_K = 4.2 \cdot 10^{-4} [\Omega_{\text{ci}}]$  is set for both species. It is used as a noise control but also as a heat source to perform temperature-gradient-driven simulations. Note that for the rest of this work, only the upgraded hybrid electron model in the  $n = m = 0$  approximation is used. Corresponding results considering the full FSA model have not yet been obtained.

To verify that the upgraded hybrid model does indeed satisfy the ambipolarity condition, we will look at the density profile evolution and the particle fluxes. In ORB5, the particle diagnostics (profiles, fluxes, etc.) are usually done on flux-surface averaged quantities, leading to 1D radial profiles. The gyrocenter density profile is defined as

$$\overline{n(\psi)} = \overline{\int f d^3 v}, \quad (4.9)$$

where the overline symbolizes the flux-surface average which is defined as

$$\overline{A(\psi)} = \frac{\int \int d\theta^* d\varphi J_{s\theta^*\varphi}(\psi, \theta^*) A(\psi, \theta^*, \varphi)}{2\pi \int d\theta^* J_{s\theta^*\varphi}(s, \theta^*)}. \quad (4.10)$$

Numerically, the flux-surface average consists of a radial binning of the markers. For example,

## 4.2. Validation of the upgraded hybrid electron model

the gyrocenter density is computed as

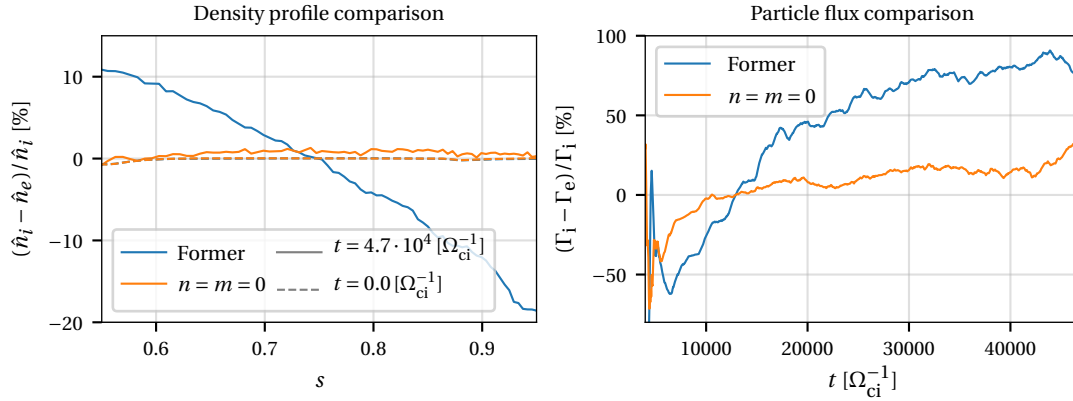
$$n(\psi_i) \approx \frac{V_{\text{tot}}}{N_{\text{tot}}} \frac{\sum_{p|\psi_p \in [\psi_i, \psi_{i+1}]} [f_0(\psi_p, \theta_{*p}, E_{\text{kin},p}) \Omega_p + w_p]}{2\pi \int_{\psi_i}^{\psi_{i+1}} \int d\psi d\theta^* J_{s\theta^* \varphi}(\psi, \theta^*)}, \quad (4.11)$$

where the sum is taken over all the markers within a bin  $[\psi_i, \psi_{i+1}]$ ,  $V_{\text{tot}} = 2\pi \int \int d\psi d\theta^* J_{s\theta^* \varphi}$  is the total torus volume, and  $N_{\text{tot}}$  is the total number of markers. Note that the use of the poloidal magnetic flux coordinate  $\psi$  for binnings is motivated by statistics properties. Indeed, radial bins defined by an equidistant partition in  $\psi$  have roughly the same volume as  $\psi \sim r^2$ , where  $r$  is the minor radius, which ensures a similar number of markers per bin. Similarly, the radial gyrocenter particle flux density is defined by

$$\Gamma_\sigma = \frac{\nabla \psi}{|\nabla \psi|} \cdot \int f_\sigma \dot{\mathbf{X}} d^3 v = \frac{1}{|\nabla \psi|} \int f_\sigma \dot{\psi} d^3 v, \quad (4.12)$$

where  $\dot{\mathbf{X}}$  is the gyrocenter characteristics defined by Eq. (2.16) and  $d\psi/dt = \nabla \psi \cdot d\mathbf{X}/dt$ .

In Figure 4.5 (left), the ion and electron density profiles are compared between the former ORB5 hybrid electron model and the  $n = m = 0$  model. In the former case, the difference



**Figure 4.5 :** The left plot shows the relative difference between the ion and electron density profiles at the beginning (dashed lines) and at the end (plain lines) of the simulation,  $t = 4.7 \cdot 10^4 [\Omega_{\text{ci}}^{-1}]$ . The densities are normalized with the averaged density,  $\hat{n} = n/\bar{n}$ . The right plot shows the difference between ion and electron particle fluxes, i.e.  $\Gamma_i - \Gamma_e$  at the radial position  $s \approx 0.6$ . In the two figures, the former ORB5 hybrid electron model and the  $n = m = 0$  model are compared against each other.

between the ion and electron densities goes up to  $\sim 20\%$  and the two profiles are markedly different on the whole radial domain considered. On the other hand, with the  $n = m = 0$  model, both ion and electron density profiles are essentially identical, with a maximum relative difference of  $\sim 1\%$ . Let us recall that the ORB5 density diagnostic actually computes

the *gyrocenter* density and not the particle density. It is then expected that the ion and electron density profiles are slightly different since the ion polarization density and FLR effects are not accounted for in the density diagnostics. On the right plot, the relative difference in particle flux between ions and electrons evaluated at  $s = 0.6$  is shown for both models. For the former model, the relative difference between ion and electron particle flux rapidly goes up to  $\sim 80\%$ . On the other hand, for the  $n = m = 0$  case, the difference is bounded by  $\sim 30\%$  which is more than 2.5 times smaller than in the former case.

The use of the  $n = m = 0$  model has greatly improved the ambipolarity condition which is confirmed by ORB5 density and particle flux diagnostics. Some differences are nevertheless observed between ion and electron densities and fluxes and are attributed to two main reasons. First, the code actually diagnoses gyrocenter density and particle fluxes, while in the quasi-neutrality condition both ion polarization and FLR effects are accounted for. Second, the  $n = m = 0$  is an approximation of the flux-surface average operation which is not exact in this case because of the toroidicity and shaping of the magnetic equilibrium.

### 4.3 Effect of the $n = m = 0$ hybrid electron model on turbulence properties

In the previous section, the  $n = m = 0$  hybrid model was verified on the following points. Linear RH tests showed a correct representation of the GAM frequency and damping rate, as well as correct ZF residual. Nonlinearly, the ambipolarity condition was shown to be satisfied on average in time. In this section, we shall compare some turbulence properties between the upgraded model and the former hybrid model. To this end, heat fluxes and  $E \times B$  shearing rates will be compared between the two models.

In ORB5, the heat flux density  $q_{\sigma,H}$  is defined as

$$q_{\sigma,H} = q_{\sigma,\text{kin}} + q_{\sigma,\text{pot}} - \frac{5}{2} m_{\sigma} v_{\text{th}}^2 \Gamma_{\sigma}, \quad (4.13)$$

$$q_{\sigma,\text{kin}}(\psi) = \frac{1}{|\nabla\psi|} \int \frac{1}{2} m_{\sigma} v^2 f_{\sigma} \psi d^3 v, \quad (4.14)$$

$$q_{\sigma,\text{pot}}(\psi) = \frac{1}{|\nabla\psi|} \int f_{\sigma} q_{\sigma} \phi_1 \psi d^3 v, \quad (4.15)$$

where  $\Gamma_{\sigma}$  is the gyrocenter particle flux defined by Eq. (4.12). Neglecting the off-diagonal terms of the transport matrix, we then define an effective heat diffusivity  $\chi_{\sigma}$  such that

$$q_{\sigma,H} = -n \chi_{\sigma} \nabla T, \quad (4.16)$$

and which is assumed to be constant on a flux surface. Taking the flux-surface average of



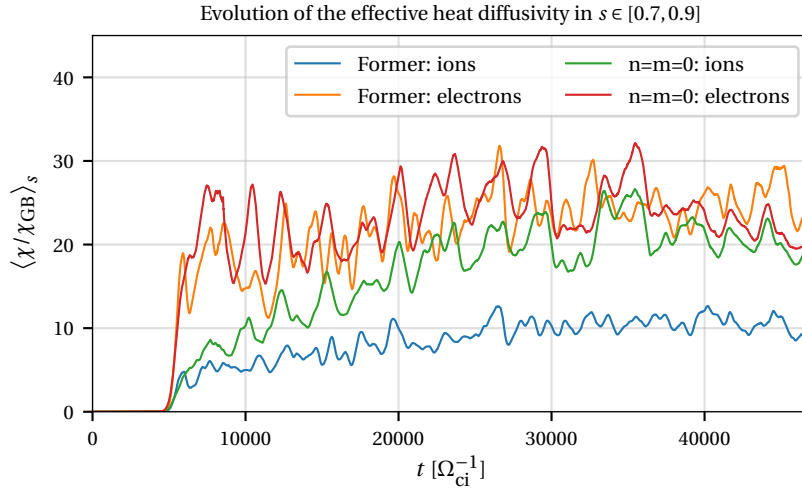
### 4.3. Effect of the $n = m = 0$ hybrid electron model on turbulence properties

Eq. (4.16) leads to the definition of the effective heat diffusivity:

$$\chi_\sigma = -\frac{q_{\sigma,H} \overline{|\nabla\psi|}}{n dT/d\psi \overline{|\nabla\psi|^2}}. \quad (4.17)$$

This definition is used throughout this work and is further normalized in gyro-Bohm units, i.e. with respect to  $\chi_{GB} = \rho_{s0}^2 c_{s0} / a$ , where  $\rho_{s0}$  and  $c_{s0}$  are respectively the ion sound Larmor radius and ion sound velocity of the reference species taken at the reference radial position  $s_0$ .

In Figure 4.6, the time evolution of the radially-averaged effective heat diffusivity is plotted for both the former ORB5 hybrid electron model and the  $n = m = 0$  hybrid model. Using



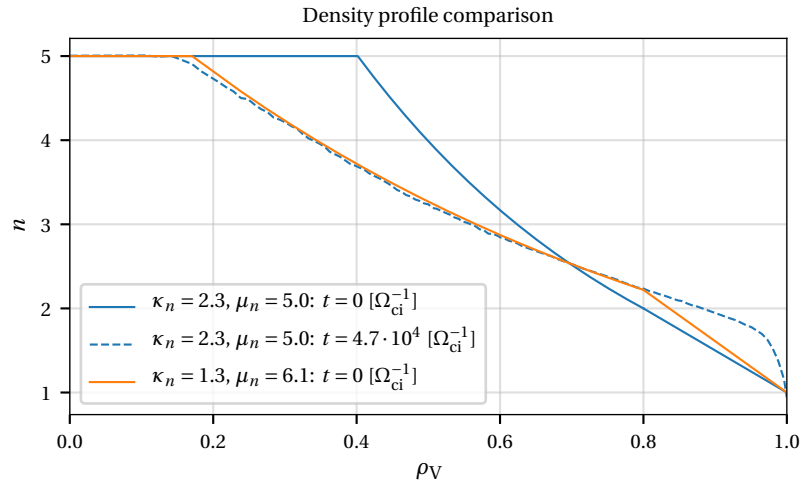
**Figure 4.6 :** Time trace of the radially-averaged effective heat diffusivities. The radial average is taken over  $s \in [0.7, 0.9]$ . Both ion and electron diffusivities are shown for the former ORB5 hybrid electron model as well as the upgraded model  $n = m = 0$ .

the former hybrid electron model, the simulation is TEM-dominated and during the whole simulation the effective electron heat diffusivity is  $\sim 2.5$  times higher than the ion diffusivity. With the  $n = m = 0$  hybrid electron model, the electron heat diffusivity is not significantly changed and its magnitude corresponds to the level obtained with the former hybrid model. On the other hand, the ion heat diffusivity is now growing linearly in time until reaching the same level as that of the electrons. They are then both closely related in magnitude and fluctuations. This result is rather surprising. Indeed, a priori one does not expect the ion transport to be significantly modified given that the correction to the hybrid electron model only directly affects the way passing electrons are treated; the inclusion of a kinetic passing electron response seems to drive ion heat transport. Two hypotheses can be used to explain this observation. First, from a linear point of view, the relaxation of the density

profiles observed in Figure 4.5 leads to larger  $\eta_i = L_n/L_T$  which could destabilize an ITG mode. Second, a nonlinear interaction could change the ZF amplitude thus leading to modified ion transport properties.

To test the first hypothesis, two linear scans are made to compute the contributions to the linear growth rate  $\gamma_{\text{lin}}$  of unstable modes coming from both ions and electrons. The parameters used for these two scans are inspired by the nonlinear TCV case presented above. Note that the numerical parameters are slightly adapted for the linear runs, namely  $\Delta t = 1 [\Omega_{\text{ci}}^{-1}]$ , the grid resolution is changed to  $N_s \times N_{\theta^*} \times N_\varphi = 256 \times 512 \times 512$ , the number of markers is set to  $20 \cdot 10^6$  for the ions and  $48 \cdot 10^6$  for the electrons, and the Krook noise control is turned off.

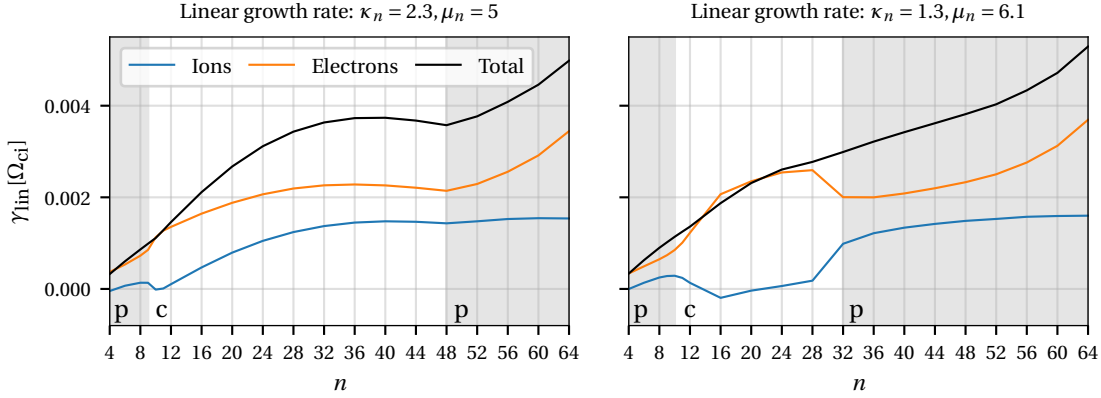
The first scan uses the same physical setup as the nonlinear run, i.e. a density profile defined by  $\kappa_n = 2.3$ ,  $\mu_n = 5$ . On the other hand, the second linear scan uses a density profile which corresponds to the relaxed density profile from the former run, i.e. defined by  $\kappa_n = 1.3$ ,  $\mu_n = 6.1$ . These values of  $\kappa_n$  and  $\mu_n$  for the second scan are obtained by fitting the final density profile of the nonlinear run assuming the same functional form as defined in Eq. (4.8), see Figure 4.7.



**Figure 4.7 :** Illustration of the density fitting procedure. The first run is initialized with a density profile defined by  $\kappa_n = 2.3$  and  $\mu_n = 5.0$  (blue plain line). At the end of the run, the final density profile (blue dashed line) is fitted in such a way to obtain the  $\kappa_n$  coefficient. Note that we have assumed the same pedestal width and edge density for both runs.

Both scans are presented in Figure 4.8. They are composed of the linear growth rate for toroidal mode numbers  $n$  ranging from 4 up to 64, which is the highest value solved in the nonlinear case. In these scans, the total linear growth rate is shown, as well as the ion and electron contributions. In both scans, there are three distinct toroidal-mode-number intervals that can be assigned to different radial regions where the most unstable modes grow. In the

### 4.3. Effect of the $n = m = 0$ hybrid electron model on turbulence properties



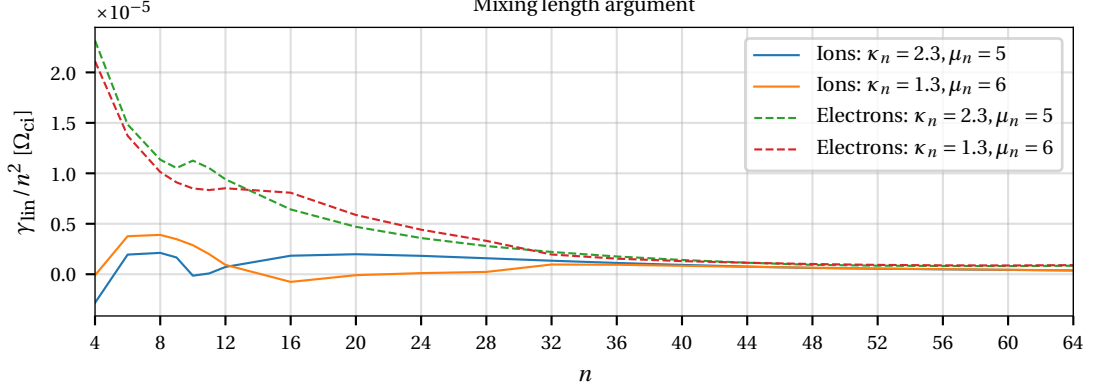
**Figure 4.8** : Linear growth rates as a function of the toroidal mode number  $n$ . The maximum  $n = 64$  value corresponds to  $k_\theta \rho_i \sim 0.85$ . The left plot shows the results for the case with  $\kappa_n = 2.3$ ,  $\mu_n = 5$  and the right plot shows the case with  $\kappa_n = 1.3$ ,  $\mu_n = 6.1$ . In both plots, the ion (blue) and electron (orange) contributions are shown as well as the total linear growth rate (black). The three distinct toroidal-mode-number intervals  $[4, 9]$ ,  $[10, 48]$ , and  $[49, 64]$  for the left plot and  $[4, 10]$ ,  $[11, 32]$ , and  $[33, 64]$  for the right plot denote regions where the instability is developing in the pedestal (p) (shaded) or in the core (c).

intervals marked with “p”, the instabilities develop in the pedestal, i.e.  $0.8 \leq \rho_V \leq 1.0$ , whereas for intervals marked with “c”, the instabilities develop in the core, i.e.  $0.4 \leq \rho_V \leq 0.8$ . For the case with the initial density profile (left plot), i.e. with the parameters  $\kappa_n = 2.3$ ,  $\mu_n = 5$ , we have an ITG-TEM mix even though the electron growth rates always dominate over the ion ones, e.g. for  $n = 36$ , the ion contribution represents  $\sim 40\%$  of the total growth rate. In the right plot, the results for the case with relaxed density, i.e.  $\kappa_n = 1.3$ ,  $\mu_n = 6.1$  are shown. The striking difference comes from the modes  $n \in [10, 32]$  developing in the core region. The ion contribution to the instability drive is negligible. For higher toroidal-mode numbers,  $n \in [33, 64]$ , there is no significant influence of the density profile on the growth rates. For low toroidal-mode numbers,  $n \in [4, 10]$ , the ion contribution to the growth rate is significantly increased by an average of  $\sim 70\%$  for the relaxed density case compared to the initial density profile case. Even though all the ion contribution to the growth rate is decreased in the core region, the low- $n$  contributions are destabilized which leads to an increased nonlinear ion heat transport. This can be illustrated using a mixing-length argument, where the anomalous heat diffusion coefficient can be estimated, using a random walk model, as

$$\chi \sim \frac{\gamma}{k_\perp^2} \sim \frac{\gamma}{k_\theta^2} \sim \frac{\gamma}{(nq(r)/r)^2}, \quad (4.18)$$

where we have used  $k_\perp \approx k_\theta$  and  $m \approx nq$  which is valid for both ITG and TEM modes, and  $r$  is the geometrical radial coordinate. Note that geometric coefficients are neglected for the sake

of simplifying the expression. In Figure 4.9, the growth rate is normalized with  $n^2$  following Eq. (4.18) to show, qualitatively, the contribution of each mode to the heat transport. For

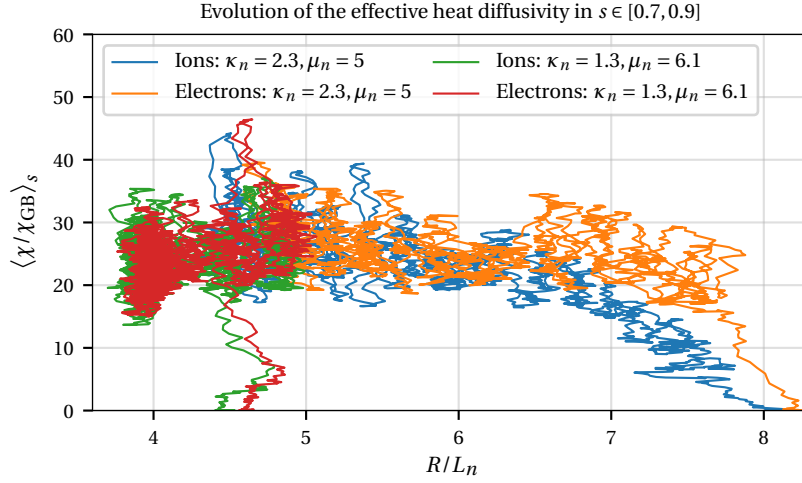


**Figure 4.9** : Estimation of the toroidal mode number diffusion coefficient using a mixing length argument. The estimation of the ion coefficient for both the case with  $\kappa_n = 2.3$ ,  $\mu_n = 5$  (blue) and the case with  $\kappa_n = 1.3$ ,  $\mu_n = 6.1$  (orange) are plotted.

$n \leq 12$ , the estimated ion heat diffusivities are approximately two times higher for the relaxed density case. On the other hand, we see that the electron heat diffusivities are slightly reduced. Around  $n = 12$  and above, the tendency is inverted. Ion diffusivities are reduced in the relaxed density case while electron diffusivities are increased. The enhanced ion heat transport is thus only partly explained through this mixing length argument.

The former nonlinear simulation is now compared to a second nonlinear run initialized with the relaxed density profile with  $\kappa_n = 1.3$ ,  $\mu_n = 6.1$ , Figure 4.7. The main motivation for doing a second run instead of continuing the former simulation is that the density relaxation becomes so large in the former run that  $|\delta f|$  becomes large. As a result, the numerical noise increases and it would be difficult to pursue the simulation. Indeed, let us recall that ORB5 relies on a control variate PIC approach in which the condition  $|\delta f| \ll f_0$  has to be satisfied to effectively reduce the numerical noise. Since the background distribution  $f_0$  is constant in time, all the profile relaxations are accounted for in the perturbed distribution  $\delta f$ . Restarting the simulation with a relaxed density profile allows one to transfer the profile relaxations to the new background distribution and restart with a clean run; this would correspond to a step-wise approximation of a more continuous adaptive  $f_0$  approach [68]. The radially-averaged effective heat diffusivities as functions of the characteristic density gradient length are plotted in Figure 4.10 for the cases with density profiles defined by  $\kappa_n = 2.3$ ,  $\mu_n = 5$  and  $\kappa_n = 1.3$ ,  $\mu_n = 6.1$ , respectively. The former simulation starts at around  $R/L_n = 8$ . The electron diffusivity reaches almost immediately its quasi-steady state level while the ion diffusivity grows slowly up to the electron level at  $R/L_n \sim 6.5$ . At this point, both diffusivities stay at their saturated level  $\langle \chi / \chi_{GB} \rangle_s \sim 25$ . The second simulation with  $\kappa_n = 1.3$ ,  $\mu_n = 6.1$  starts at

### 4.3. Effect of the $n = m = 0$ hybrid electron model on turbulence properties

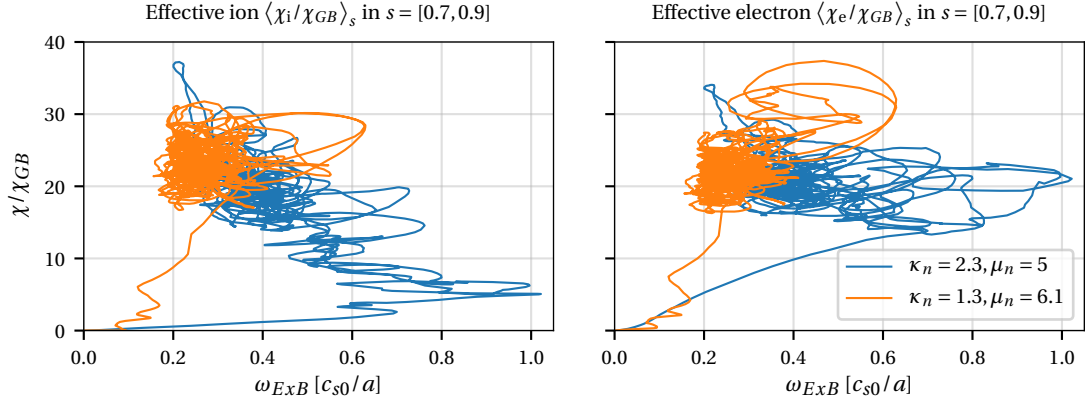


**Figure 4.10** : Radially-averaged effective heat diffusivities as a function of the density characteristic gradient length  $R/L_n$ . The radial average is made over  $s \in [0.7, 0.9]$ . Both ion and electron heat diffusivities are shown for the case with  $\kappa_n = 2.3, \mu_n = 5$  and the case with  $\kappa_n = 1.3, \mu_n = 6.1$ .

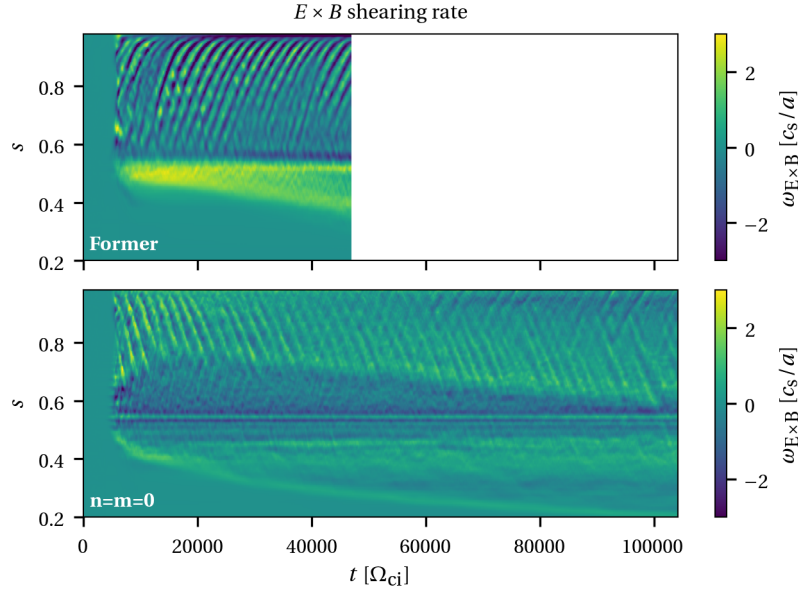
$R/L_n \sim 4.5$ . In this case, both ion and electron diffusivities quickly reach their saturated level which correspond to the one of the previous run. Two preliminary conclusions can be drawn from this figure. First, restarting the simulation using the final state of the previous run as initial condition gives consistent results. Second, we see that the density relaxation for the second case is much smaller than in the first case, even though both runs have the same physical duration. This suggests that the simulation state is approaching the zero-particle-flux condition [126, 127, 128, 129].

We now look at the radially-averaged effective heat diffusivities as a function of the  $E \times B$  shearing rate, Figure 4.11. In the left plot is shown the ion effective heat diffusivity. As the density profile relaxes, the  $E \times B$  shearing rate amplitude decreases leading to an increase in ion heat transport. On the other hand, for the electrons (right plot), the shearing rate does not significantly affect the transport properties. Indeed, for ITG-driven turbulence it is well established that ZFs provide the main saturation mechanism that quenches turbulence, but for TEM-driven turbulence the situation is not so clear and depends on various parameters such as  $T_e/T_i$  and  $\nabla T_e$  [40]. In this case, the TEM-driven turbulence is not significantly affected by zonal flow quenching.

In Figure 4.12, the spatio-temporal evolution of the  $E \times B$  shearing rate  $\omega_{E \times B}$  is shown for both the former hybrid and the  $n = m = 0$  models. It is interesting to note that with the  $n = m = 0$  model, there is a long-lived sharp radial structure located at  $s_1 \sim 0.5$  which corresponds to the rational surface with  $q(s_1) = 1$ . This structure is further shown in Figure 4.13 (top), where the radial profiles of  $\omega_{E \times B}$  are time-averaged over  $t \in [1, 4.6] \cdot 10^4 [\Omega_{ci}]$  for both the



**Figure 4.11** : Radially-averaged effective heat diffusivities as a function of  $E \times B$  shearing rate. The radial average is made over  $s \in [0.7, 0.9]$ . Both ion (left) and electron (right) heat diffusivities are shown for the case with  $\kappa_n = 2.3, \mu_n = 5$  and the case with  $\kappa_n = 1.3, \mu_n = 6.1$ .

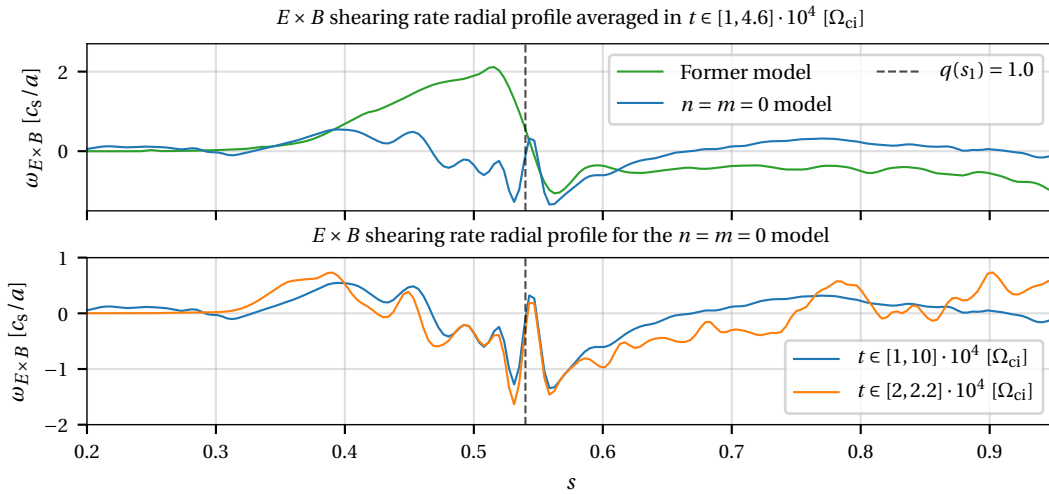


**Figure 4.12** :  $E \times B$  shearing rate evolution for using the former ORB5 hybrid model (top) and the upgraded  $n = m = 0$  model (bottom).

### 4.3. Effect of the $n = m = 0$ hybrid electron model on turbulence properties

former model and the  $n = m = 0$  model. In the former case, a broader structure is observed with a sharp transition from positive to negative shearing rate. The other significant difference between the two models is the direction of the  $\omega_{E \times B}$  avalanches; their frequency, however, is left unchanged. With the former model, avalanches propagate outward and they are present on the whole second half of the radial domain, i.e.  $s \in [0.6, 1.0]$ . With the  $n = m = 0$  model, avalanches mainly propagate inward, especially during the early simulation stages. At longer times,  $t > 4 \cdot 10^4 [\Omega_{ci}^{-1}]$ , there is a mix of inward and outward going avalanches. Note also the presence of a radial interval, around  $s \in [0.55, 0.70]$ , with outward propagating avalanches of small amplitude, and which disappears with time as the inward going avalanches penetrate it. Note that, the different avalanche directions observed in Figure 4.12 is correlated with the sign of the average shearing rate [130]. For the former hybrid model, a negative averaged shear rate is observed in  $s \in [0.54, 0.95]$  which corresponds to outward propagating avalanches. For the  $n = m = 0$  model, three distinct regions delimited by  $s \in [0.54, 0.66]$ ,  $s \in [0.66, 0.91]$ , and  $s \in [0.91, 0.95]$  are observed with respectively negative, positive, and negative averaged shear rate, which corresponds to mainly outward, inward, and outward propagating avalanches.

In Figure 4.13 (bottom), the shearing rate profile of the  $n = m = 0$  model is shown averaged over the whole turbulent phase and averaged over an avalanche period. In both cases, there is a large, fine structure around  $s_1 \sim 0.5$ . On other lowest mode rational surfaces, e.g.  $q = 2$  at  $s = 0.9$ , no such sharp structure is found. Simulations with higher resolution should be performed to confirm this observation.



**Figure 4.13 :** Radial profile of the  $E \times B$  shearing rate. The top plot shows a comparison between the former ORB5 model (green line) and the  $n = m = 0$  model (blue line) averaged in  $t \in [1, 4.6] \cdot 10^4 [\Omega_{ci}]$ . The bottom plot shows the  $n = m = 0$  model. The blue line is a time average over the whole turbulent phase and the orange line shows a short-term average over one avalanche period around  $t = 2 \cdot 10^4 [\Omega_{ci}^{-1}]$ . The vertical dashed line at  $s_1 = 0.54$  represents the radial position where  $q = 1$ .

#### 4.4 Core and pedestal TEM turbulent transport

Experimental investigations made in the TCV tokamak [49] showed that the plasma profiles exhibit a pedestal region—even in L-mode discharges—in which gradients are constant and a core region, excluding the sawtooth inversion region, in which the logarithmic gradients are constant. Various numerical studies using the ORB5 code [131, 132] have then tried to reproduce qualitatively the conclusions of TCV experiments. A set of simulations of ITG turbulence—even though the considered TCV discharges showed a TEM-dominant regime—using the same setup as the simulations presented in Section 4.2.2, but with the adiabatic electron model, were performed. The conclusions of these studies are in qualitative agreement with the experiments and show that the core/pedestal transport exhibit non-local effects that seem related to avalanches mediated by ZFs. Also, transport in the core is much stiffer than in the pedestal if measured as a function of the logarithmic temperature gradient but not so much if measured as a function of the linear gradient. Finally, it was shown that the nonlinear critical temperature logarithmic gradient is higher in the pedestal than in the core. On the other hand, the nonlinear critical temperature linear gradients were observed to be comparable in the core and the pedestal.

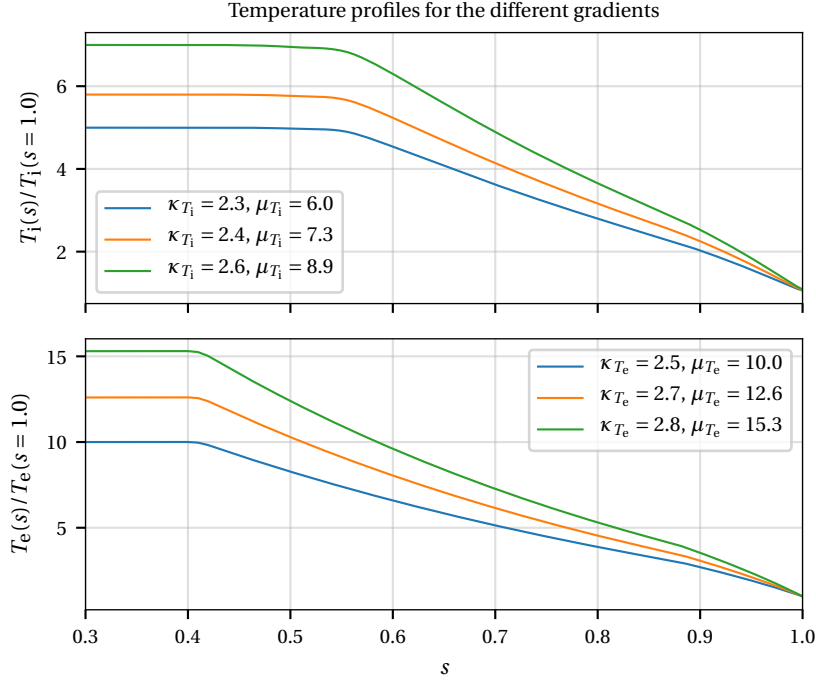
In this section, these numerical studies are extended to a TEM-dominant regime using the upgraded  $n = m = 0$  hybrid electron model. To this end, the TEM simulation presented in Section 4.2.2 is used as a reference case, from which a gradient scan is made. Similarly to [131], different values of  $\kappa_{T\sigma}$  and  $\mu_{T\sigma}$  are chosen. In this case, the pedestal gradient is increased by steps of  $\sim 20\%$  and the core logarithmic gradient by steps of  $\sim 5\%$ . The three simulation parameters are given in Table 4.2 and the corresponding initial temperature profiles are shown in Figure 4.14.

**Table 4.2 :** Summary of the ion and electron gradient parameters used to define the temperature profiles for the gradient scan. The first run is the reference simulation.

Run #	$T_{0i}$	$\kappa_{Ti}$	$\mu_{Ti}$	$T_{0e}$	$\kappa_{Te}$	$\mu_{Te}$
1	5.0	2.3	6.0	10.0	2.5	10
2	5.8	2.4	7.3	12.6	2.7	12.6
3	7.0	2.6	8.9	15.3	2.8	15.3

Characterization of the core/pedestal transport requires sufficiently long simulations to obtain a meaningful statistical average. Indeed, the temperature gradients do vary slightly during the simulation even though we use a Krook operator to maintain them as close as possible to the initial ones. This is particularly critical when studying stiffness problems because a small variation in temperature gradient leads to a large variation in the computed heat power. The time trace of the radially averaged heat diffusivities of the reference case is shown in Figure 4.15. The heat diffusivities seem to be in a quasi-steady state in  $t \in [2, 6] \cdot 10^4 [\Omega_{ci}^{-1}]$ , followed by a slow downward trend which is attributed to the particle losses at the

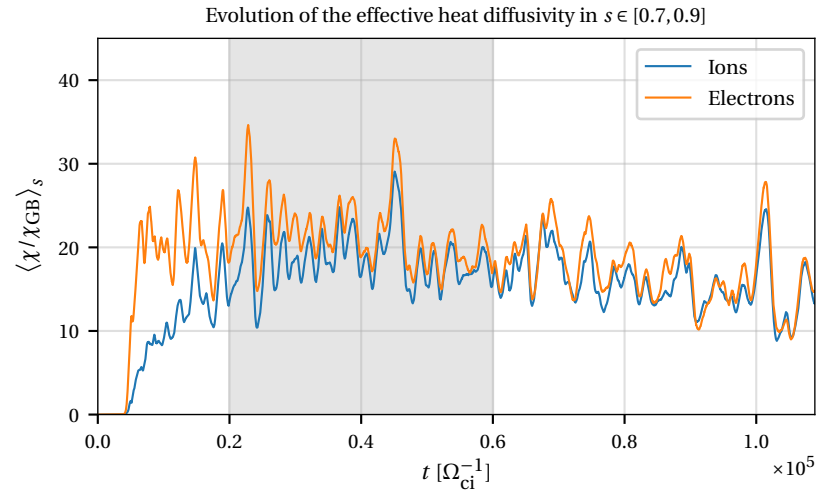




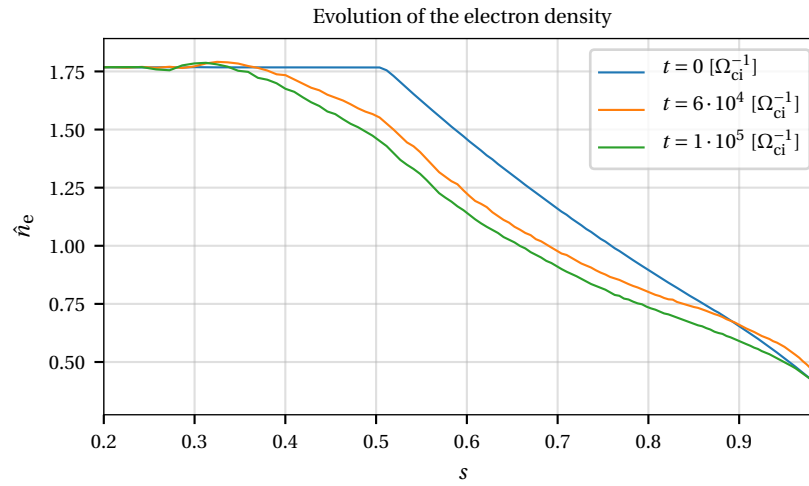
**Figure 4.14 :** Illustration of the ion and electron temperature profiles for the gradient scan. Note that the radial domain extends up to  $s = 0$  but is not shown here for convenience.

edge. Indeed, during the first half of the run, the density profile quickly relaxes, Figure 4.16. As it approaches the state of a zero-particle-flux condition, the relaxation speed decreases. On a long time scale, density losses at the edge become important. We recall that the markers leaving the radial domain are re-injected back into the plasma with a zero weight, which corresponds to a particle—and other higher moments—source/sink. During the second half of the simulation, from  $t = 6 \cdot 10^4 [\Omega_{ci}^{-1}]$  to  $t = 1 \cdot 10^5 [\Omega_{ci}^{-1}]$ , particles exit at the outer radial edge and the density loss propagates into the core. Similar conclusions can also be drawn for the ion density profile. This overall relaxation contributes to a reduction of the transport level.

It is also important to note that with such relaxed profiles, the condition  $|\delta f| \ll f_0$ , required for the  $\delta f$  PIC method to operate at a reduced noise level is not ensured. Indeed, in the simulation presented in Figure 4.16, the difference between the initial density profile and the final one can be up to 30%. Furthermore, the markers also drift radially into regions of phase space that were not initially sampled, leading to a sampling dilution. Both of these issues contribute to a progressive drop in signal-to-noise ratio during the simulation; it approaches a value of 10 at the end of the run but always stays above it. The density relaxation due to the radial transport and particle losses at the radial edge, together with the noise issue, make long simulations in quasi-steady state difficult to carry out. For this reason, all the following results



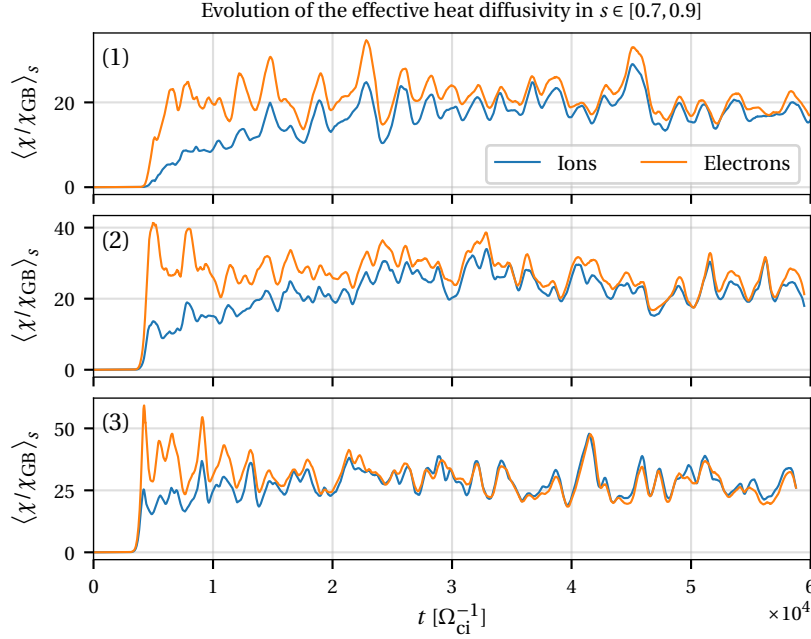
**Figure 4.15 :** Reference simulations: time trace of the effective heat diffusivities radially averaged over  $s \in [0.7, 0.9]$ .



**Figure 4.16 :** Time evolution of the electron density profile.

will be obtained by averaging the quantities over the first temporary quasi-steady state, i.e.  $t \in [2, 6] \cdot 10^4 [\Omega_{ci}^{-1}]$ . Longer quasi-steady simulations could, in principle, be achieved using a particle source to compensate for the edge losses and using an adaptive  $f_0$  method, such as re-sampling the phase space using the relaxed density profile as it was done in Section 4.3.

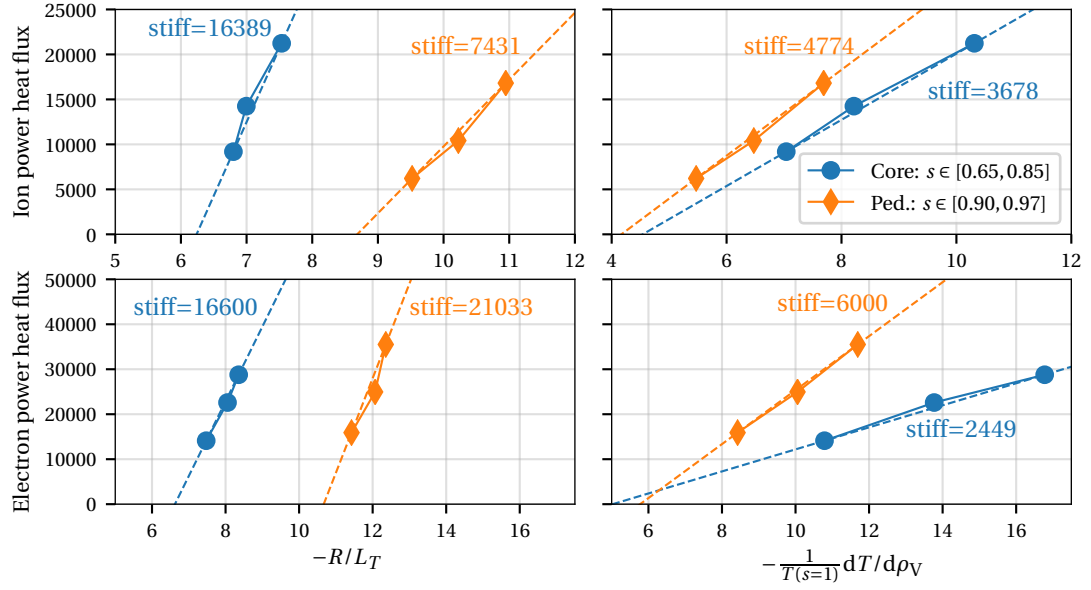
The time evolution of the radially averaged effective heat diffusivities is shown in Figure 4.17. It is interesting to note that in all the cases, the ion heat diffusivity increases to join



**Figure 4.17** : Time traces of radially averaged effective heat diffusivities. The radial average is made in  $s \in [0.7, 0.9]$ . The simulation number is shown in the top left corner of the plot and the corresponding gradient values can be found in Table 4.2.

the electron diffusivity level, as already discussed in Section 4.3. As expected, the steeper the gradients, the higher the heat transport. Indeed, averaging the effective heat diffusivities over  $t \in [2, 6] \cdot 10^4 [\Omega_{ci}^{-1}]$  yields a value of 18.6, 24.0, and 29.0 for the run numbers 1, 2, and 3 respectively, which represents an increase of around 30% and 56% with respect to the reference simulation.

We now take the time and radial averages of the ion and electron heat powers and plot them as a function of the time- and radially-averaged logarithmic gradient  $R/L_T$  and the normalized linear gradient  $(1/T(\rho_V = 1))dT/d\rho_V$ , Figure 4.18. For the ion heat power (top plots), similar conclusions as [132] are found. If measured using  $R/L_T$ , the ion heat transport is much more stiff in the core than in the pedestal; an increase in power of a factor 2.3 leads to



**Figure 4.18 :** Heat power as a function of the temperature logarithmic and linear gradients. Both the power and the gradients are time averaged over  $t \in [2, 6] \cdot 10^4 [\Omega_{ci}^{-1}]$  and radially averaged over  $s \in [0.65, 0.85]$  in the core and  $s \in [0.90, 0.97]$  in the pedestal. The “stiff” parameter is an indicator of the stiffness. It represents the slope of the line joining the first and last points of the scan. The powers are given in units of  $n_i(s=1)m_i\rho_{si}(s=1)c_{si}(s=1)$ .

an increase of around 12% of the logarithmic gradient. On the other hand, if one considers the linear gradient, transport stiffness is comparable between core and pedestal. Here, an increase by a factor of  $\sim 2.5$  in heat power leads to an increase of around 45% of the linear gradient in both the core and pedestal. The nonlinear critical logarithmic gradient is much higher in the pedestal than in the core, while similar nonlinear critical linear gradients are observed for both regions. Let us now consider the electron heat transport (bottom plots). In this case, both the core and pedestal seem to have a comparable stiffness if measured with  $R/L_T$ . It has to be noted that the critical logarithmic gradient is also higher in the pedestal than in the core. Finally, similarly to the ion case, both core and pedestal have similar critical linear gradients.

## 4.5 Summary of the chapter

In this chapter, a correction to the former ORB5 hybrid electron model has been presented. This new model allows us to run nonlinear simulations of electron modes at a lower numerical cost than the fully kinetic electron model while satisfying the ambipolarity condition, preserving the GAM frequency, and avoiding the  $\omega_H$  mode. The model has been validated linearly using Rosenbluth-Hinton tests to evaluate the GAM frequency and the ZF residual.

The new model has a similar GAM frequency as both adiabatic and fully kinetic models and they converge toward the same ZF residual. Using nonlinear simulation, the problem of the ambipolarity condition has been addressed. The new model leads to consistent ion and electron radial density profiles while the use of the former model resulted in radically different profiles. Similarly, the difference between the ion and electron particle fluxes is more than 2.5 times smaller using the corrected model. The non-zero difference has been attributed to two factors. First, the ORB5 diagnostics only diagnose the gyrocenter moments, leaving out the ion polarization and FLR effects. Second, the corrected model uses an approximation of the flux-surface average.

A nonlinear simulation using TCV-like parameters with the corrected hybrid electron model showed that the ion heat transport was radically affected by the inclusion of a partial kinetic response from passing electrons. Indeed, in this simulation, the heat transport went from a TEM-dominant regime to an ITG-TEM mixed regime. This was attributed to the relaxation of the density profiles that lead to the destabilization of low- $n$  modes. Using a mixing-length argument, it was shown that those modes actually dominate the heat transport. Further investigations using a diagnostic able to quantitatively estimate the contribution to the heat/particle flux for each toroidal mode will be done in a future work to confirm those conclusions. In addition, it is observed that the density relaxation leads to the reduction of the  $E \times B$  shearing rate thus increasing the ion heat transport. The reasons for this reduction are not understood yet.

A comparison of the zonal properties between the former ORB5 hybrid electron and the  $n = m = 0$  models is shown. Using the upgraded model, a sharp, long-lived radial structure appears at the lowest-order mode rational surface  $q = 1$ . The presence of such a structure was attributed to the non-adiabatic response of passing electrons in previous works [133, 134, 135, 136, 137, 54]. Furthermore, it was shown that the zonal radial modulations were created through nonlinear coupling of non-zonal modes. In our case, only a kinetic response from the zonal component of the passing electron is accounted for. However, all the passing electrons are pushed using the electric field consistent with the quasi-neutrality equation. Non-zonal modes can still induce a zonal response through self-coupling of modes such that  $m \neq 0$ . A future work proposal to confirm this hypothesis is to implement a Fourier diagnostics to study the interaction between passing electron zonal and non zonal modes. The comparison between the two hybrid electron models also showed different  $\omega_{E \times B}$  properties. The direction of propagation of the avalanches is changed and a radial zone of incoherent fluctuations is created.

Finally, a short characterization of the core/pedestal heat transport is made using a gradient scan of TEM-dominant simulations. It is shown that the ion heat transport is much more stiff in the core than in the pedestal region. For the electrons, a comparable stiffness between the core and pedestal regions is observed and the nonlinear critical temperature logarithmic gradients are much higher in the pedestal than in the core. Further investigations are still

needed. Longer simulations could be achieved using a particle source and an adaptive  $f_0$  method. This study considered only three different values of gradients in the scan. More data points could be added, especially around marginal stability. Also, using a fully kinetic electron model and/or including collisions, which are known to play an important role near the edge, could allow for more quantitative results. A similar study can be done by doing a power scan instead of a gradient scan by using flux-driven simulations.

## 5 Flux driven simulations

As presented in Chapter 3, the ORB5 code is based on a low-noise  $\delta f$  PIC approach consisting of separating the full distribution function  $f$  into a constant-in-time background part  $f_0$  and a fluctuating part  $\delta f$ . The background distribution function is then typically set to be an equilibrium distribution function  $f_{\text{eq}}$  known analytically and only the fluctuating part of the distribution function is evolved using numerical particles, so-called markers. In addition to generating less numerical noise than the standard full PIC method for a similar number of markers, provided that the ordering  $|\delta f| \ll f_0$  is satisfied, this approach allows one to run so-called temperature *gradient-driven* (GD) simulations in which the temperature profiles are kept close to their initial value while the other moments of the total distribution function are left free to evolve. This is typically done by relaxing the  $\delta f$  part using a modified Krook operator effectively acting as an heat-only source/sink.

The main advantage of the GD approach is that it allows one to simulate a plasma having temperature profiles close to predefined values and at a reduced numerical cost because the system quickly reaches a quasi-steady state. On the other hand, this approach suffers from two main problems making comparison with experiments difficult. First, it does not reflect the experimental situation in which heat fluxes are applied by various localized heat sources. Second, experimental plasmas are often found to exhibit profile stiffness: above a certain critical temperature gradient, a small increase of gradient leads to a large increase of heat flux. This is particularly problematic for GD simulations because a small uncertainty on the prescribed temperature gradients leads to large uncertainties on the computed heat fluxes, and thus the effective heat source required for maintaining profiles in GD runs end up being far from the actual experimental ones, both in their spatial localization and in the level of injected power.

To avoid these drawbacks, the so-called *flux-driven* (FD) approach can be used. In FD simulations, the heat sources and associated fluxes are given as an input and the temperature profile is left free to relax toward an equilibrium. However, since the profile relaxation is done over transport time scales, FD simulations are very demanding in terms of numerical resources, especially for large machines,  $\rho_* \ll 1$ . Indeed, using a simple random walk model

for turbulent transport, the transport time scale can be estimated to scale with the square of the normalized minor radius,  $t_{\text{transp}}(\nu_{\text{th},i}/a) \sim 1/\rho_*^2$ .

Modern gyrokinetic codes, e.g. GYSELA[24], GT5D[19], GENE[18], and XGC[12] now implement the FD approach. In this chapter, we present how the ORB5 code has been adapted to run FD simulations. This begins by a short summary of the heat sources/sinks implemented in the code followed by a description of the non-conventional “mode switching” consisting of starting a GD simulation and pursuing it as a FD run. Then, illustrative simulations of ITG and TEM driven turbulence are presented. Finally, an analysis of the effect of heat sources/sinks on the distribution function is made and the future outlook is presented.

### 5.1 Summary of the different heat sources

The key element of temperature GD or FD simulations is the heating operator. It provides the heat flux necessary to maintain a certain level of turbulence and thus avoid transient decaying simulations where profiles relax and marginal stability is quickly reached. In ORB5, different heat sources are implemented, each having different properties in addition to injecting heat. Let us review the ones used for this study.

#### 5.1.1 Krook operator heat source

The Krook operator, which is primarily used for controlling noise due to marker discreteness, can effectively act as a heat source, allowing for GD simulation by relaxing the energy moment of  $\delta f$  while conserving other moments. As presented in Section 3.1.3.1, the Krook source term can be written as

$$S_K = -\gamma_K \delta f + S_K^{\text{corr}}, \quad (5.1)$$

where the first term on the r.h.s. damps  $\delta f$  with a rate  $\gamma_K$  while the second term is a correction defined by Eq. (3.50) which allows one to control separately the conservation, up to machine precision, of density, parallel momentum, energy, and zonal flows (ZF) in radial bins. This approach is very interesting as it offers the possibility to run temperature GD simulations where only the temperature profile is constrained while the other moments evolve self-consistently with the gyrokinetic Vlasov-Maxwell system of equations. This is of particular interest for simulating tokamak discharges in which all the particle fueling occurs at the boundary and where the zero particle-flux condition is observed in steady-state. However, this implementation of a heat source based on a modified Krook operator is not suitable for FD simulations as the resulting injected heat flux cannot be prescribed.



### 5.1.2 Fixed heat source

The second heat source, hereby called *fixed heat source* (FHS) is given by

$$S_{\text{FHS}} = \gamma_{\text{FHS}} S_r(s) S_E(E_k) = \gamma_{\text{FHS}} S_r(s) \frac{1}{T} \left( \frac{E_k}{T} - \frac{3}{2} \right) f_L(n, T, E_k), \quad (5.2)$$

where  $S_r(s)$  is the radial profile of the heat source—read from an external file—and usually consists of a sum of analytical Gaussian shapes defined by  $\exp[-(s - s_{\text{pos}})^2 / \Delta s^2]$ , where  $s_{\text{pos}}$  is the position of the peak and  $\Delta s$  is the Gaussian standard deviation. The value of the profile is then linearly interpolated at the particle position. Note that the radial profile  $S_r(s)$  is normalized so that  $\max_s S_r(s) = 1$  and the source amplitude is set by the  $\gamma_{\text{FHS}}$  constant. The energy-dependent part of the FHS,  $S_E(E_k)$ , is given by the derivative of the local Maxwellian  $f_L$  with respect to the temperature,  $S_E \sim \partial f_L / \partial T$ , which aims at decreasing the number of sub-thermal particles and increasing the number of supra-thermal particles, Figure 5.1 (a). Furthermore,  $S_E(E_k)$  involves an associated Laguerre polynomial of order 1 of the normalized energy,  $L_1^{1/2}(E_k / T)$ , which ensures that, in the limit of an infinite number of markers, there is no injection of density or momentum. For example, the conservation of density can be proven using a simple argument. The density, which is defined as

$$n = \int f_L d^3 v \neq n(T), \quad (5.3)$$

does not depend on the temperature. Thus,

$$0 = \frac{\partial n}{\partial T} = \int \frac{\partial f_L}{\partial T} d^3 v = \int S_E d^3 v, \quad (5.4)$$

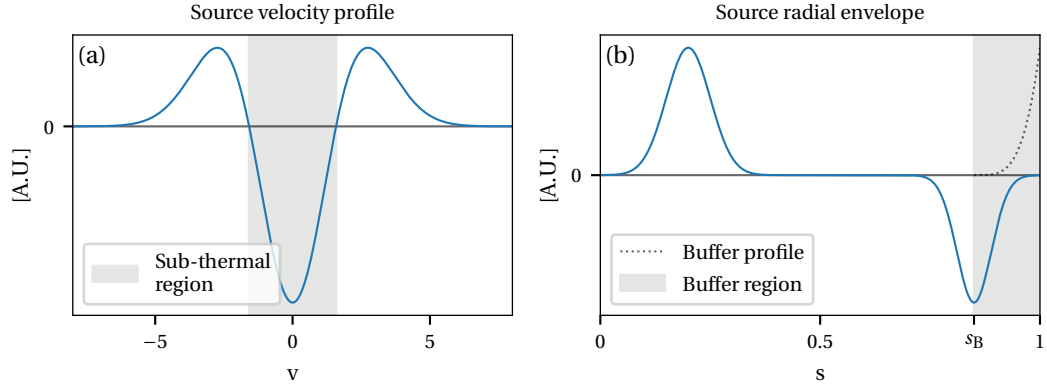
i.e. as long as the integration is done exactly, the heat source conserves density. Practically, however, the conservation of density, parallel momentum, and zonal flows is further enforced in radial bins by a similar conservation scheme as the one used for the Krook noise control, Eq. (3.50).

Other functional forms of the heat source are possible in Eq. (5.2). For example, the FHS can be defined to be proportional to the difference between a Maxwellian at the plasma temperature,  $f_P$ , and a Maxwellian at the targeted temperature,  $f_T$ :

$$S \propto S_r(s) (f_P - f_T). \quad (5.5)$$

However, it can be shown that using a first order Taylor series in  $(T_P - T_T)$ , both forms are equivalent [138].

To achieve a statistical steady state, all the injected heat must be consistently dissipated by some mechanism. In this work, two methods have been tested. First, a heat sink is placed at the outer radial edge and consists of a negative profile for the FHS, Figure 5.1 (b). Usually, the



**Figure 5.1** : Illustration of the source profiles. (a) Velocity dependence of the fixed heat source. The particles in the sub-thermal region, grey area, are depleted in favor of those in the supra-thermal region. (b) Radial envelope of the source. Two methods are available for extracting the heat: either we use a sink which is the FHS with a negative profile or a buffer (grey area) in which a Krook operator heavily damps  $\delta f$  with an amplitude following a polynomial profile (grey dotted line).

amplitude of the sink is determined such that the total injected power is zero, i.e. such that

$$P_{\text{tot}} = 2\pi \int_0^{2\pi} \int_{s_{\min}}^{s_{\max}} J_{s\theta^*} \varphi(s, \theta^*) S_r(s) ds d\theta^* = 0, \quad (5.6)$$

where the radial profile of the source  $S_r(s)$  is composed of the source and sink. This heat extraction method already reveals a few drawbacks. First, the heat sink is always active and with a fixed rate set to evacuate all the injected heat. This may lead to critical situations where the total distribution function actually becomes negative. This would be typically the case at the very beginning of a simulation where the turbulence is not yet well established to transport all the injected heat to the sink. Second, the sink amplitude satisfying Eq. (5.6) is computed externally, using a grid that does not necessarily correspond to the simulation grid. The numerical integration errors are then added to the linear interpolation errors in ORB5 which may lead to a greater error on Eq. (5.6). In other words, a non-zero total heat flux is injected in the system, which makes it impossible to reach a quasi-steady state. This is however mitigated by the fact that the error on the total injected power is still relatively small and its effect is only visible on time scales longer than the typical simulation time.

An alternative solution to the fixed heat sink is to use a buffer region in which all the perturbations are dissipated. In ORB5, this is obtained by using a Krook operator inspired from [139]:

$$S_B = -\gamma_B(s) \delta f, \quad (5.7)$$

where the buffer rate profile  $\gamma_B(s)$  is zero outside the buffer region and is a normalized fourth order polynomial,  $(s - s_B)^4 / (s_{\max} - s_B)^4$ , inside, ie for  $s_B \leq s \leq s_{\max}$ . The boundaries of the buffer region are set by the  $s_B$  parameter up to the outer radial edge at  $s_{\max}$ , Figure 5.1 (b). This effectively adaptive heat extraction method has the advantage of relaxing the distribution function to the background distribution function  $f_0$  which also ensures that it never becomes negative. However, the damping rate  $\gamma_B$  must be set to sufficiently high values to effectively relax  $\delta f$  in the narrow buffer region and it may affect the turbulence outside the buffer through non-local effects.

### 5.1.3 Adaptive Krook heat source

Finally, a last heating operator has been implemented which consists of a minor modification of the Krook heat source presented in Section 5.1.1, which was unable to inject an, *a priori*, predicted heat power. In this variant, the adaptive Krook heat source, hereby denoted with the subscript AK, automatically computes the rate required to obtain a certain targeted heat power  $P_T$ . For that, the total injected heat power  $P_{\text{in}}$  for a Krook heat source with rate  $\gamma = 1$  is first computed:

$$P_{\text{in}} = 2\pi \int_0^{2\pi} \int_{s_{\min}}^{s_{\max}} J_{s\theta^* \varphi}(s, \theta^*) S_K S_r(s) ds d\theta^* \quad (5.8)$$

$$= 2\pi \int_0^{2\pi} \int_{s_{\min}}^{s_{\max}} J_{s\theta^* \varphi}(s, \theta^*) (-\delta f + S_K^{\text{corr}}) S_r(s) ds d\theta^*. \quad (5.9)$$

Note that in this case, a radial profile is provided for the Krook heat source. Then, the required rate can be easily computed as a rescaling, i.e.

$$\gamma_{\text{AK}} = \frac{P_T}{P_{\text{in}}}, \quad (5.10)$$

leading to the adaptive Krook heat source term

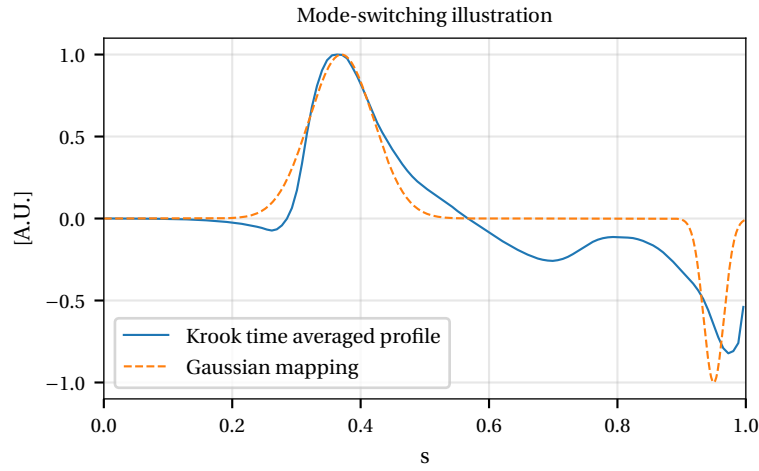
$$S_{\text{AK}} = [-\gamma_{\text{AK}} \delta f + S_{\text{AK}}^{\text{corr}}] S_r(s). \quad (5.11)$$

To ensure that the adaptive Krook heat source rate is within acceptable values, it is furthermore ensured that  $0 \leq \gamma_{\text{AK}} \leq \gamma_{\max}$ , where  $\gamma_{\max}$  is the maximum allowed rate and is provided as an input parameter.

## 5.2 Switching from gradient- to flux-driven simulations

As a first step towards FD simulations, we adopted a non-conventional approach, hereby called *mode switching*. It consists of first starting a GD simulation using the Krook operator acting as noise control and heat source, i.e. conserving  $n$ ,  $v_{\parallel}$ , and ZF. The effective heat source profile associated to the Krook operator is then time-averaged and radially parametrized using

a Gaussian shape, which is used as a radial envelope  $S_r(s)$  for the heat source. The goal of this operation is to obtain a localized radial profile—the Krook effective heat source profile may be non-zero everywhere in the radial domain—in such a way to inject the same level of heat as with the time-averaged Krook. For the heat dissipation at the outer edge, either the buffer or the heat source with a negative Gaussian shape acting as a heat sink can be used. In any case, only the heat source is mapped, in shape and in amplitude, to the Krook profile. The dissipation operator—buffer or sink—is placed as close as possible to the outer radial edge in order to dissipate all the injected heat and allow for the largest source-free region possible where turbulence can evolve unaffected by the source and heat dissipation mechanism (sink or buffer). An illustration of this procedure using a heat sink can be found in Figure 5.2. Note



**Figure 5.2 :** Illustration of the mode-switching approach. The effective heat source profile obtained from the Krook operator (blue plain line) is mapped using a Gaussian shape (orange dashed line). In this sketch, the dissipation mechanism is a heat sink consisting of a negative Gaussian shape.

that an exact matching between the Krook heat source profile and the Gaussian mapping is not expected. The goal is to define a radial envelope for the heat source that is localized and that allows the source to inject the same total amount of heat as the Krook operator. Once the radial envelope has been defined, the GD simulation is continued in FD mode using heat source Eq. (5.2) or Eq. (5.11). For this part of the run, the Krook operator, Eq. (5.1), is only used as a noise control, i.e.  $S_K^{\text{corr}}$  is set up to conserve  $n$ ,  $v_{\parallel}$ ,  $E_k$ , and ZF.

The mode-switching method is adopted here for two main reasons. First, we start from an already defined quasi-steady state and impose the *a priori* necessary heat flux to keep the system in this state. This allows the simulation to reach a quasi-steady turbulent state more quickly as the plasma profile relaxation is minimized. Furthermore, it also allows one to keep the numerical benefits of the control variate  $\delta f$  method used in ORB5. Indeed, the condition

for this method to effectively reduce the numerical noise is that  $|\delta f| \ll f_0$  is satisfied. If the final quasi-steady state obtained in FD mode is far from the equilibrium imposed during the first GD phase, the amplitude of  $\delta f$  may become comparable to  $f_0$ , thus deteriorating the simulation quality.

The mode-switching approach can nonetheless pose a certain number of problems. First, the radial heat source profile obtained from the Krook may be totally unrealistic and not at all localized. In this case, it may be difficult to find a correct mapping that leads to a similar quasi-steady state as for the GD run. This difficulty would break the main advantage of this method. Second, during a GD run, the system is somehow artificially forced to stay in a certain state. Indeed, as the perturbed distribution  $\delta f$  grows, the Krook operator effect is stronger to ensure that the total distribution function stays close to the equilibrium defined as input, i.e.  $f \sim f_0$ . This is especially the case during the first moments of the nonlinear phase where large heat flux bursts are observed. This constraint may lead to a “bifurcation” of the system toward a state it would never have reached if it had evolved freely. Despite the above mentioned issues of the mode-switching approach, we think that it is a good candidate to develop the FD abilities of ORB5 and explore the effects of the heat sources on the plasma distribution functions. Note, however, that ORB5 is also capable of running full FD simulations without using a preparatory GD simulation.

### 5.2.1 Illustration of the mode-switching approach with ITG and TEM driven simulations

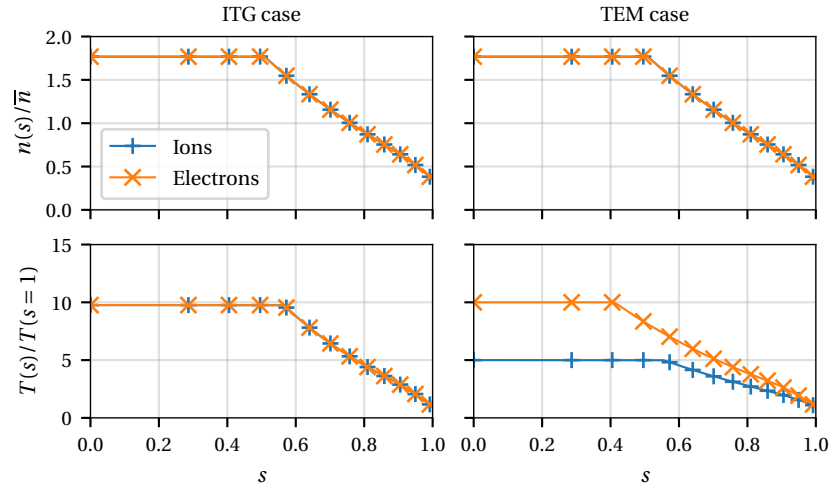
To illustrate the mode-switching method described in the previous section, two test-case simulations—an ITG- and a TEM-dominant one—are now presented. Note that the TEM-dominant simulation is the same as the one already shown in Chapter 4 but is recalled here for convenience. Both cases use the same axisymmetric ideal MHD equilibrium computed by the CHEASE code and based on TCV shot #43516 parameters [49]. This equilibrium, Figure 4.3, has an aspect ratio of 3.64, an elongation of 1.44 and a triangularity of 0.20 at the last closed flux surface. The safety factor profile is quadratic,  $q(s) = q_0 + (q_{\text{edge}} - q_0)s^2$ , with  $q_0 = 0.78$  and  $q_{\text{edge}} = 3.29$ . The reference magnetic surface for the normalization is  $s_0 = 1.0$ , ions play the role of reference species for the normalization, and  $\rho_*(s_0) = \rho_i(s_0)/a = 1/245$ . All the simulations assume a deuterium plasma without impurities.

If not stated otherwise, the ITG-dominated case is defined as follows. The ion temperature profile is defined by equation

$$T(\rho_V) = \begin{cases} \min \{T_0, T_{\text{ped}} \exp[-\kappa_T(\rho_V - \rho_{\text{ped}})]\} & \text{for } \rho_V \leq \rho_{\text{ped}}, \\ T_1 - \mu_T \cdot (\rho_V - 1) & \text{for } \rho_V > \rho_{\text{ped}}, \end{cases} \quad (5.12)$$

with  $T_{\text{ped}} = T_1 + \mu_T(1 - \rho_{\text{ped}})$  which mimics qualitatively the experimental profiles obtained in [49]. The radial coordinate is given by  $\rho_V = \sqrt{V(\psi)/V(\psi_e)}$  with  $V(\psi)$  being the volume enclosed by the magnetic surface  $\psi = \text{const}$  and  $\psi_e$  is the value of  $\psi$  at the radial outer edge.

The temperature at the top of the pedestal is given by  $T_{\text{ped}} = T_1 + \mu_T(1 - \rho_{\text{ped}})$ . The density profile definition is based on the same equation and the following parameters are used:  $\rho_{\text{ped}} = 0.8$ ,  $T_0^i = 10$ ,  $T_1^i = 1$ ,  $\kappa_T^i = 2.3$ ,  $\mu_T^i = 12$ ,  $n_0^i = 5$ ,  $n_1^i = 1$ ,  $\mu_n^i = 5$ ,  $\kappa_n^i = 3.1$ , Figure 5.3 (left). The electrons have the same temperature and density profiles as the ions and are assumed to have an adiabatic response. Note that the density and temperature profiles chosen for these simulations are only in qualitative agreement with the experiments. No quantitative agreement was attempted for this study. The numerical resolution for these runs is  $N_p = 128 \cdot 10^6$  markers and  $\Delta t = 20 [\Omega_{ci}^{-1}]$ . The grid resolution is  $N_s \times N_{\theta^*} \times N_{\varphi} = 256 \times 512 \times 256$ . The field-aligned Fourier filter is set to keep only modes such that  $0 \leq n \leq 64$  and satisfying  $m = nq(s) \pm 5$ . Finally, the Krook rate is set to  $\gamma_K = 1.4 \cdot 10^{-4} [\Omega_{ci}]$ , which is about 10% of the maximum linear ITG growth rate.

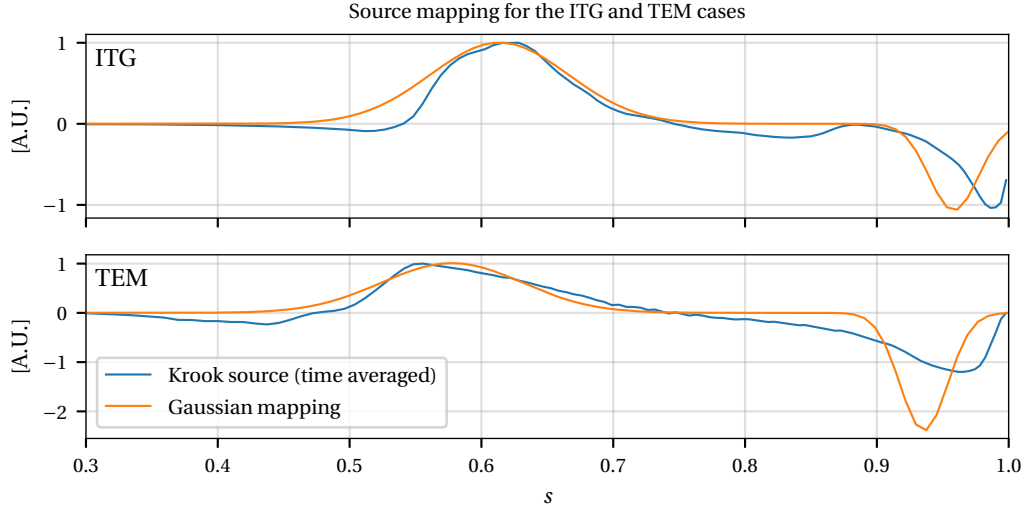


**Figure 5.3** : Background density (top row) and temperature (bottom row) profiles for both ITG-dominated (left column) and TEM-dominated (right column) cases.

Unless stated otherwise, the TEM simulations were carried out with the upgraded hybrid electron model. The considered ion and electron temperature and density profiles are defined similarly as the ITG-dominated case with the following parameters:  $\rho_{\text{ped}} = 0.8$ ,  $T_0^i = 5$ ,  $T_1^i = 1$ ,  $\kappa_T^i = 2.3$ ,  $\mu_T^i = 6$ ,  $n_0^i = 5$ ,  $n_1^i = 1$ ,  $\mu_n^i = 5$ ,  $\kappa_n^i = 2.3$  and  $T_0^e = 10$ ,  $T_1^e = 1$ ,  $\kappa_T^e = 2.5$ ,  $\mu_T^e = 10$ ,  $n_0^e = 5$ ,  $n_1^e = 1$ ,  $\mu_n^e = 5$ ,  $\kappa_n^e = 2.3$ , Figure 5.3 (right). Ions and electrons have the same density profile and different temperature profiles except at the edge where  $T_i(s_0) = T_e(s_0)$ . The ion to electron mass ratio is set to  $m_i/m_e = 200$ . The numerical resolution for these runs is  $N_p = 256 \cdot 10^6$  markers for each species and  $\Delta t = 2 [\Omega_{ci}^{-1}]$ . The grid resolution is  $N_s \times N_{\theta^*} \times N_{\varphi} = 256 \times 512 \times 256$ . The field-aligned Fourier filter is set to keep only modes such that  $0 \leq n \leq 64$  and satisfying  $m = nq(s) \pm 5$ . Finally, the Krook rate is set to  $\gamma_K = 4.2 \cdot 10^{-4} [\Omega_{ci}]$  for both species, which is about 10% of the maximum linear TEM growth rate..

## 5.2. Switching from gradient- to flux-driven simulations

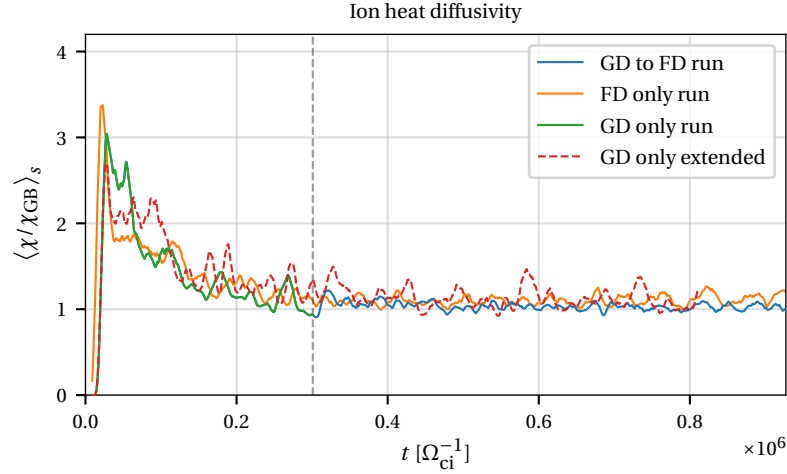
For both ITG and TEM cases, the FD heat source radial profile is defined using the methodology presented in Section 5.2, Figure 5.4.



**Figure 5.4 :** Presentation of the source radial profiles obtained from the mapping of the effective GD heat source due to the Krook operator. In the ITG case, the Krook contribution to the heat source is averaged over time in  $t \in [2.5, 3] \cdot 10^5 [\Omega_{ci}^{-1}]$ . In the TEM case, the time average is made over  $t \in [3.5, 4] \cdot 10^4 [\Omega_{ci}^{-1}]$ .

Let us begin by presenting the results for the ITG case. In Figure 5.5, the ion heat diffusivity is plotted as a function of time for three different cases: a GD-only simulation, a simulation using the mode-switching approach presented previously, i.e. a continuation of the GD run in FD using a source which is an analytical mapping of the Krook effective heat source profile from the GD simulation, and a FD-only run that uses the same source profile. For the FD simulations, the FHS is used, Eq. (5.2). The vertical dashed line represents the time at which the switching takes place. Note that the duration of the longest simulations,  $\sim 10^6 [\Omega_{ci}^{-1}]$ , corresponds to a typical measured TCV energy confinement time, i.e.  $\sim 10$  ms [140, 49, 141]. The switching between the GD run and its FD continuation occurs smoothly and the FD statistical steady state is very close to the GD one. Comparing now the run using the mode switching technique with the FD-only run, we see that they both converge toward a similar steady state, on average, with a 7% relative difference which lies in the fluctuation error. We also note that during the first GD phase, the FD-only run is comparable to the GD run. Concerning the heat transport properties, beginning the run in GD and switching to FD after the first bursts does not affect the final steady state compared to a FD-only run.

In Figure 5.6, the time trace of the radially averaged  $E \times B$  shearing rate magnitude is plotted for the three cases presented in the previous paragraph. Here again, the switching

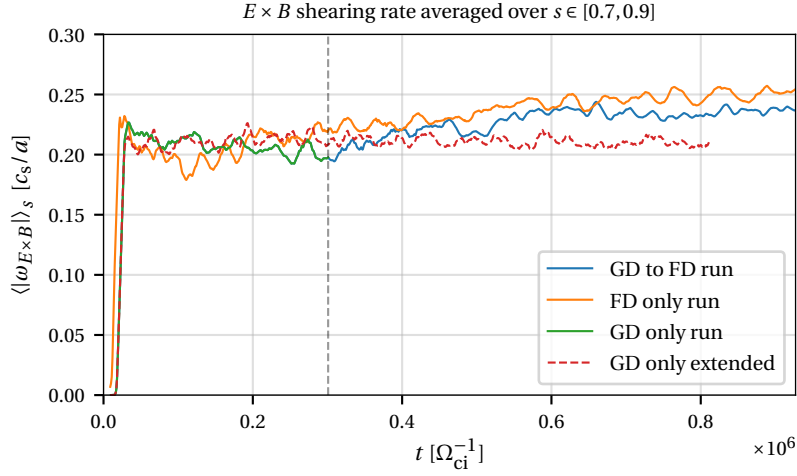


**Figure 5.5 :** ITG case. Time trace of the radially averaged ion heat diffusivity in gyro-Bohm units. The signal is further time averaged using a rolling average with a window of  $t = 1 \cdot 10^4 [\Omega_{ci}^{-1}]$ . The radial average is made over  $s \in [0.7, 0.9]$ . Three simulations are presented: a GD-only (green/red dashed), a FD-only (orange), and a simulation using the mode switching that is started as GD and continued FD. The vertical dashed line at  $t \sim 3 \cdot 10^5 [\Omega_{ci}^{-1}]$  marks where the switching from GD to FD is made. Note that the two curves labeled “GD only run” and “GD only extended” are in fact simulations made using the same input file but with different code versions. Because of the turbulent nature of the problem, it cannot be expected to obtain the exact same results even though they are equivalent on a statistical average.

between GD and FD occurs smoothly. During the GD part of the run, both the GD and the FD runs are comparable on average. After the switching, the shearing rate levels increase and they both converge toward a similar steady state. The FD-only run has an averaged shearing rate magnitude that is systematically  $\sim 5\%$  higher than the run using mode switching. This difference is slightly higher than the fluctuation error. In Figure 5.7 is plotted the radial profile of the  $E \times B$  shearing rate averaged over two different time intervals. The upper figure shows the radial profile where the time average is taken during the GD part of the run, i.e.  $t \in [1, 3] \cdot 10^5 [\Omega_{ci}^{-1}]$ , while the lower figure shows the radial profile for the FD part of the run, i.e.  $t \in [7, 9] \cdot 10^5 [\Omega_{ci}^{-1}]$ . During the early phase of the simulation, Figure (a), the only difference between the runs is the heat source. The mode-switched run is heated using the Krook operator which always adapts to the state of the system while the FD-only run is heated using the fixed heat source which injects the equivalent heat as the time averaged Krook operator. Although different instantaneously, the heat sources are similar in shape and amplitude on average over time. The heat sinks, however, are significantly different for both cases since for the FD-only run the sink is arbitrarily located near the outer edge with an amplitude such that all the injected heat is evacuated. This difference induces different developments of the  $E \times B$



## 5.2. Switching from gradient- to flux-driven simulations



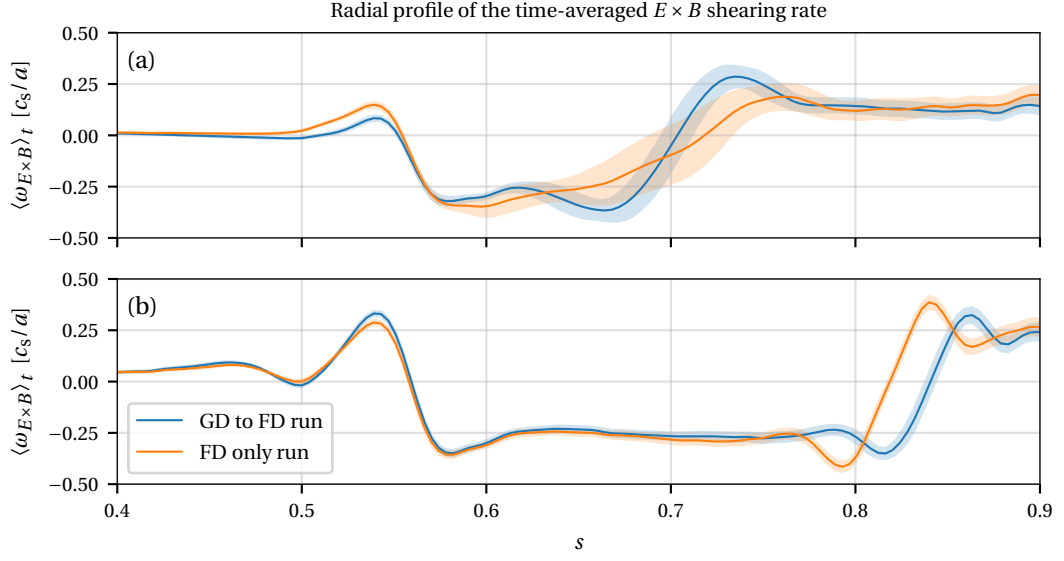
**Figure 5.6 :** ITG case. Time evolution of the radially averaged  $E \times B$  shearing rate magnitude. The radial average is done over  $s \in [0.7, 0.9]$  and a rolling time average with a window of  $t = 1 \cdot 10^4 [\Omega_{ci}^{-1}]$  is further applied. Three simulations are presented: a GD-only (green/red dashed), a FD-only (orange), and a simulation using the mode switching that is started as GD and continued FD. The vertical dashed line at  $t \sim 3 \cdot 10^5 [\Omega_{ci}^{-1}]$  marks where the switching from GD to FD is made. Same remark as Figure 5.5 for the GD curves.

shearing rate's radial structures. Both runs have similar radial profiles in  $0.4 \leq s \leq 0.64$  and in the outer part of the domain but the profiles are different in the source/sink-free region of the domain. During the later times of the simulation, Figure (b), both runs use the same heat source/sink with the same parameters. The radial profiles of the  $E \times B$  shearing rate are similar in  $0.4 \leq s \leq 0.78$  but differ in the outer region. This is a consequence of the early differences in the development of the radial structures. It has to be noted that the difference in magnitude between the two runs observed in Figure 5.6 comes from the different profiles seen in the outer region.

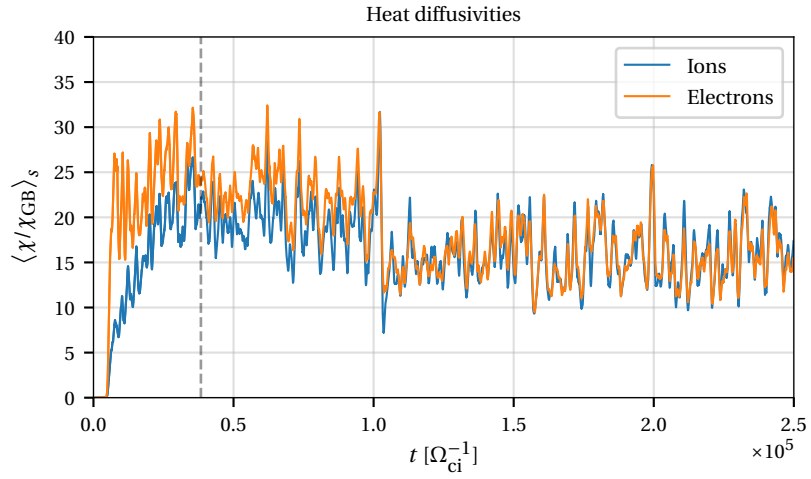
Similar results are obtained for the TEM-dominant case. In Figures 5.8 and 5.9, the mode switching is illustrated considering the time evolution of the ion and electron heat diffusivities. As for the ITG-dominant case presented in the last paragraphs, the transition occurs smoothly from GD to FD for both species. Up to  $t = 1 \cdot 10^5 [\Omega_{ci}^{-1}]$ , both simulations converge toward a similar statistical steady state. Note that during the first GD phase,  $t < 4 \cdot 10^4 [\Omega_{ci}^{-1}]$ , both simulations are identical<sup>1</sup>.

In Figure 5.10, the  $E \times B$  shearing rate magnitude evolution is compared between a purely GD run and the mode-switched simulation. Before the switching, both runs coincide exactly

<sup>1</sup>The different instantaneous fluctuations are observed because different version of the code were used to perform those simulations. Due to the turbulent nature of the system, exact matching cannot be expected even though both simulations must (and do) converge toward a similar statistical steady state.

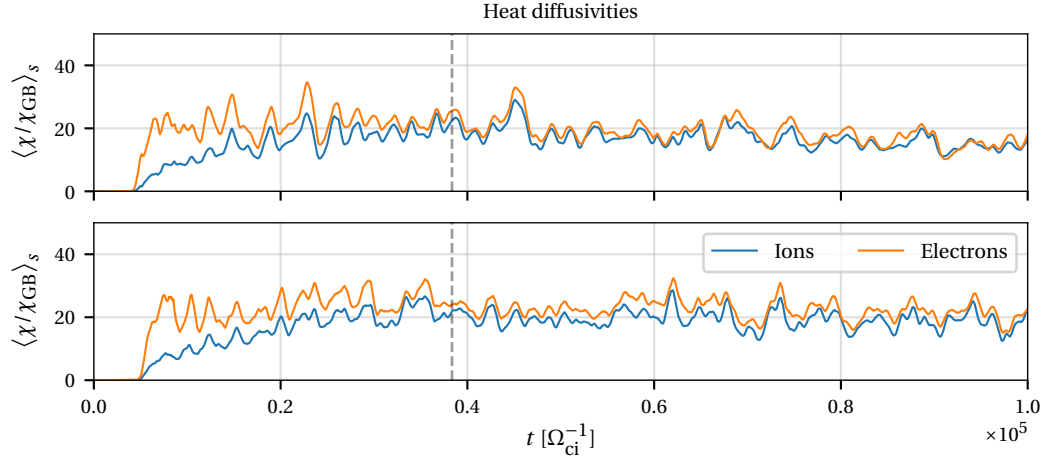


**Figure 5.7** : ITG case. radial profile of the  $E \times B$  shearing rate. Figure (a) shows the radial profile time averaged over the initial GD part,  $t \in [1, 3] \cdot 10^5 [\Omega_{ci}^{-1}]$ , and figure (b) shows the radial profile time averaged during the FD part of the run,  $t \in [7, 9] \cdot 10^5 [\Omega_{ci}^{-1}]$ . The blue curve shows the profile of the mode-switched run and the orange curve shows the results of the FD-only simulation. The shaded areas show the standard deviation.

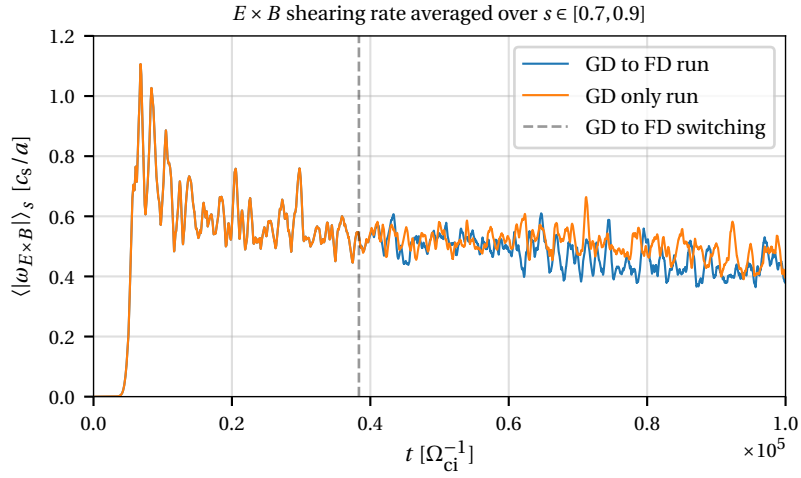


**Figure 5.8** : TEM case. Time trace of the radially averaged ion heat diffusivity in gyro-Bohm units. The radial average is made over  $s \in [0.7, 0.9]$ . The vertical dashed line at  $t \sim 4 \cdot 10^4 [\Omega_{ci}^{-1}]$  marks where the switching from GD to FD is made.

## 5.2. Switching from gradient- to flux-driven simulations



**Figure 5.9** : TEM case. Similar to Figure 5.8. Comparison between GD (top) and GD to FD (bottom).



**Figure 5.10** : TEM case. Evolution of the  $E \times B$  shearing rate magnitude radially averaged over  $s \in [0.7, 0.9]$ . The comparison is made between a purely GD run and a simulation using the mode switching. The vertical dashed line at  $t \sim 4 \cdot 10^4 [\Omega_{ci}^{-1}]$  marks where the switching from GD to FD is made.

because they are the same—the first part of the mode-switched simulation is a GD run. After the switching, their evolutions differ but they converge toward a similar time-averaged steady state with a relative difference of  $\sim 6\%$ , which is within the fluctuation error.

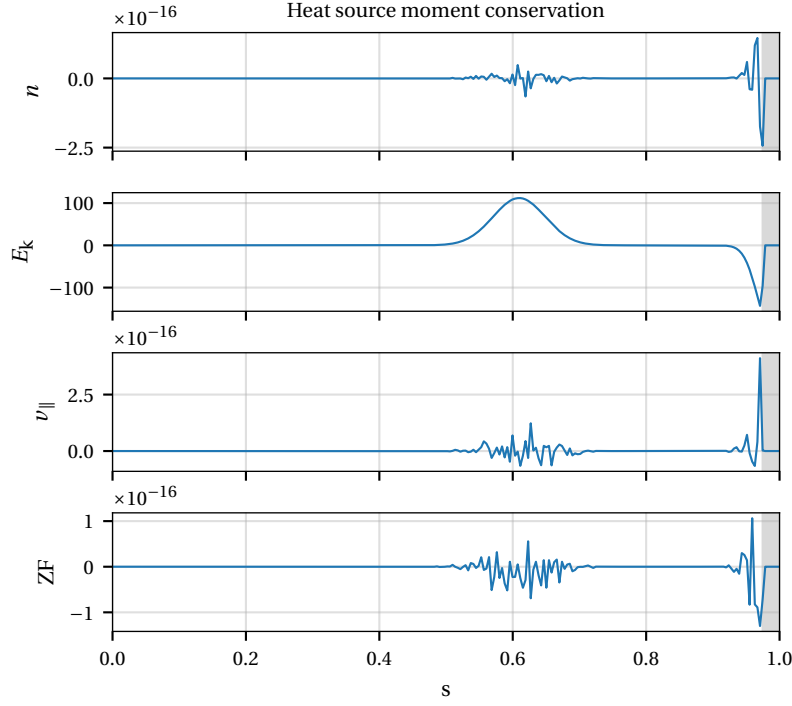
A conclusion from the ITG/TEM test cases is that the mode-switching approach is robust. It gives similar results for the transport properties and zonal flow shearing rates as the FD-only simulation using the same source. Constraining the system during the early strong bursts does not affect the final state except for a hysteresis effect observed on the evolution of the  $E \times B$  shearing rate radial structures.

### 5.3 Heat source effect on the phase space

The FHS used in the previous runs to inject heat into the system inevitably modifies the distribution function in velocity space. This can create spurious sources/sinks of moments that affect the simulation results in a non-physical way. By definition, the FHS defined by Eq. (5.2) does not inject density or parallel momentum. Further conservation properties are also applied by an ad-hoc correction term that ensures conservation of the perturbed density, parallel momentum, and ZF in radial bins up to machine precision, Figure 5.11.

In the ITG simulation presented in Section 5.2.1, the adiabatic electron model was used. No particle transport is thus expected as the perturbed electrostatic potential and perturbed density are always in phase; the density profiles are therefore intrinsically frozen. However, whenever the FHS is activated, a long-time evolution of the density is observed, Figure 5.12. Around  $s = 0.5$ , there is a nonphysical loss of density located where the density profile gradient becomes negative and where the heat source's radial profile gradient becomes positive. Surprisingly, this difference in density mainly comes from the  $f_0$  contribution that is computed from an integration of the  $p$ -weights and which is supposed to be constant in time. No relevant difference is observed on the  $\delta f$  contribution. This suggests a deviation of the marker trajectories from the physical particle trajectories; the heat source only acts on the marker  $\delta f$  weights. There is also a small density loss at the outer radial edge but this is due to the boundary conditions. Indeed, the markers leaving the domain are re-injected with a zero weight which corresponds to a sink (or source) of moments. Since this ITG run uses adiabatic electrons and it is verified that the heat source only injects heat, it is worrisome to observe such a density deviation. In the case of kinetic electron simulations, it becomes difficult to discriminate between the density variations due to the actual transport and the ones due to the spurious changes from the heat source.

In ORB5, the radial profile of the background density is computed as a flux-surface average



**Figure 5.11** : Moments injected by the FHS. By construction, this source injects heat and conserves density, parallel momentum, and ZF on a flux surface. The ad-hoc correction source term ensures this conservation up to machine precision. The gray area is a cut-off zone in which the heat source is set to zero artificially.

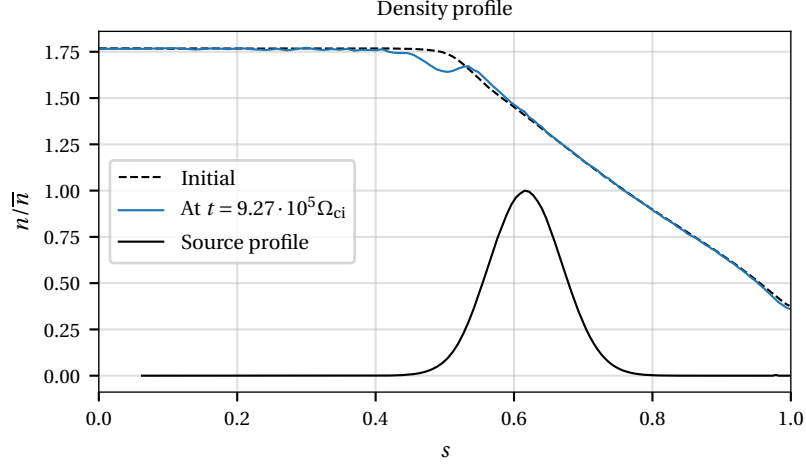
approximated using a radial binning:

$$n_0(s) \approx n(s_i) = \frac{V_{\text{tot}}}{V_{\text{bin}}(s_i) N_p} \sum_{p \in [s_i, s_{i+1})} f_0(s_p) \Omega_p, \quad (5.13)$$

$$V_{\text{tot}} = \int_{s_{\min}}^{s_{\max}} ds \int_0^{2\pi} d\theta^* \int_0^{2\pi} d\varphi J_{s\theta^*\varphi}(s, \theta^*), \quad (5.14)$$

$$V_{\text{bin}}(s_i) = \int_{s_i}^{s_{i+1}} ds \int_0^{2\pi} d\theta^* \int_0^{2\pi} d\varphi J_{s\theta^*\varphi}(s, \theta^*), \quad (5.15)$$

where the interval  $[s_i, s_{i+1})$  denotes a bin  $i$ ,  $N_p$  is the total number of markers,  $V_{\text{tot}}$  is the volume of the torus, and  $V_{\text{bin}}$  is the bin volume. For the binning, the background function  $f_0$  is evaluated at the marker gyro-center position. In the computation of the background density, only the binning part is done using particles; the volumes are computed from the field grid using a Gauss-Legendre quadrature formula. This suggests that the dip in density is coming from a phase-space sampling problem which appears through the elemental phase-space volumes  $\Omega_p$ .



**Figure 5.12** : Density radial profile evolution with the FHS turned on for the ITG run with adiabatic electrons. The heat source radial envelope (black plain line) is placed as a reference to mark the source position.

To investigate this issue, a 2D diagnostic of the distribution function properties is implemented in ORB5. It allows one to reconstruct the marker properties (weight, elemental phase space volume, etc.) by binning these properties in the 2D plane  $(s, v^2)$ , where  $v^2 = v_{\parallel}^2 + \mu B$  is the marker velocity modulus squared. More particularly, we are interested in the evolution of the volume  $V_{ij}$  of a bin  $[s_i, s_{i+1}) \times [v_j^2, v_{j+1}^2)$  in the velocity space, i.e.

$$V_{ij} = \frac{1}{V_{\text{bin}}(s_i)} \int_{s_i}^{s_{i+1}} ds \int_0^{2\pi} d\theta^* \int_0^{2\pi} d\varphi \int_{v_j \leq |v| < v_{j+1}} d^3 v J_{s\theta^* \varphi}(s, \theta^*) = \int_{v_j}^{v_{j+1}} d^3 v. \quad (5.16)$$

This equation can be evaluated numerically using a Monte-Carlo (MC) approach:

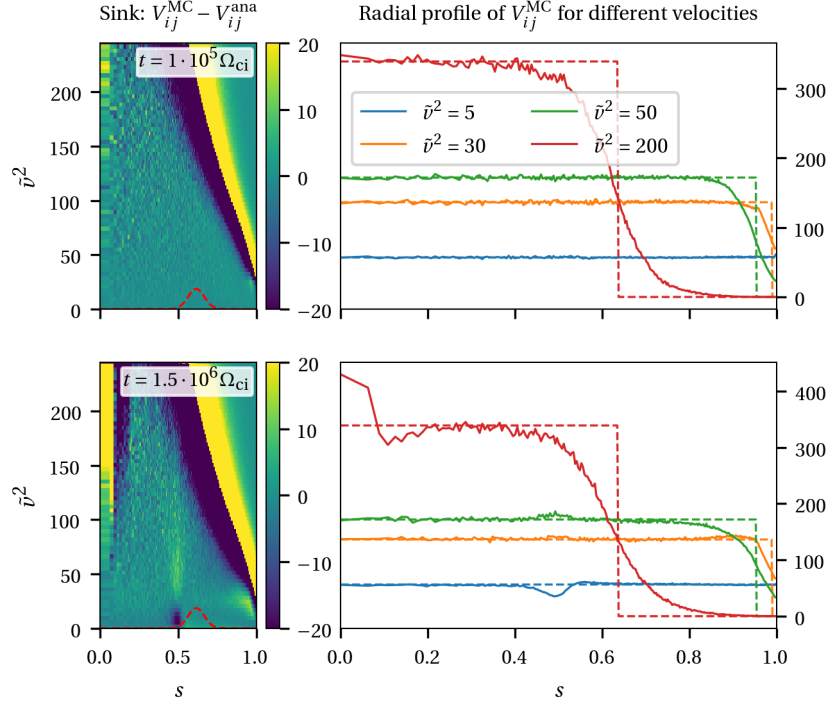
$$V_{ij}^{\text{MC}} = \frac{V_{\text{tot}}}{N_p} \sum_{p \in [s_i, s_{i+1}) \times [v_j, v_{j+1})} \Omega_p, \quad (5.17)$$

or analytically:

$$V_{ij}^{\text{ana}} = \frac{4\pi}{3} \left( v_{j+1}^{\frac{3}{2}} - v_j^{\frac{3}{2}} \right). \quad (5.18)$$

In the following, we start from the reference ITG case presented above. Variants with different heat source/sink and parameters will also be considered. In Figure 5.13, the sampling diagnostic results for the reference ITG case are presented. The first row shows the difference  $V_{ij}^{\text{MC}} - V_{ij}^{\text{ana}}$  in the 2D plane  $(s, v^2)$  and the radial profile of the bin volume compared to its

analytical value for different velocities at  $t = 10^5 \Omega_{ci}$ . The second row shows the same quantities at  $t = 1.5 \cdot 10^6 \Omega_{ci}$ . Note that, for the two time snapshots considered, the FHS is activated. The



**Figure 5.13** : Evolution of the bin volume in velocity phase space for the reference ITG case. The left column shows the difference between the numerical estimation and the analytical value while the right column shows the radial profile of the bin volume in velocity space for different velocity values. The dashed lines represent the analytical volumes. The velocity scale is normalized with  $\tilde{v}^2 = [v/v_{th}(s=1)]^2$ . The rows of the plot represent two snapshots of the system at  $t = 10^5 [\Omega_{ci}^{-1}]$  (upper) and  $t = 1.5 \cdot 10^6 [\Omega_{ci}^{-1}]$  (lower). The red dashed line in the 2D plots represents the heat source position; the amplitude is arbitrary.

2D plots are used to compare the evolution of the sampling. At the beginning of a simulation, the markers are loaded following a uniform distribution in the radial and in the  $v^2$  directions. In the velocity space, the markers are loaded such that

$$v^2 \leq v_{\max}^2 = \kappa_v^2 v_{th}^2 = \kappa_v^2 \frac{T_i}{m_i}, \quad (5.19)$$

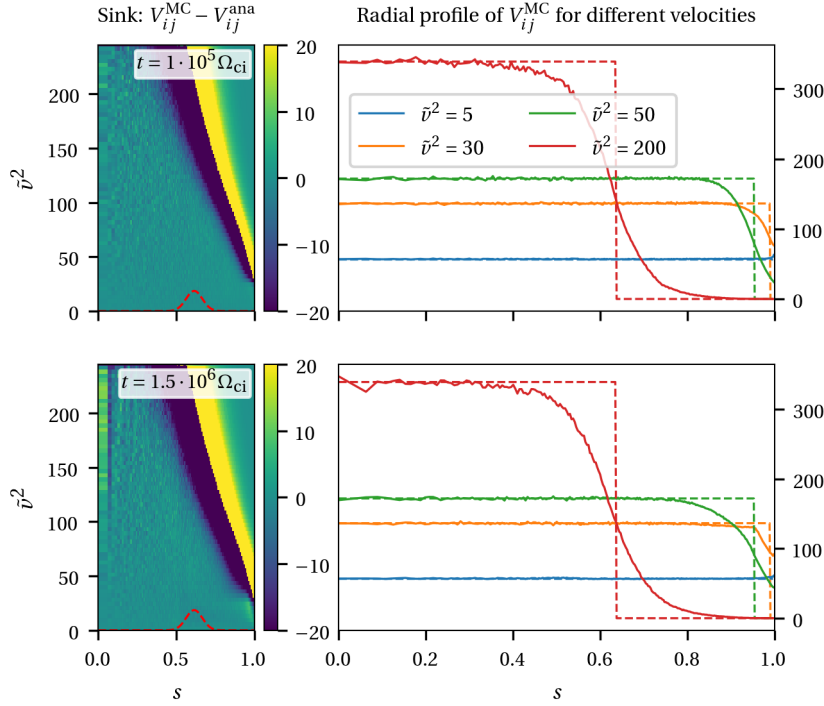
where  $\kappa_v$  is an input parameter and is equal to 5 for the simulations presented in this section. In other words, the limit of initialization in velocity space follows proportionally to the thermal velocity. The same limit is applied on the analytical values for a better comparison, i.e.  $V_{ij}^{ana} = 0$  if  $v_j^2 > \kappa_v^2 T_i / m_i$ . The radial profiles of the volumes in velocity space presented in the right

column are always taken at  $\tilde{v}^2 = 5, 30, 50$ , and  $200$ , where  $\tilde{v} = v/v_{th}(s=1)$  is a normalized velocity, and are compared with the corresponding analytical value in dashed line. The absence of sampling in certain regions of phase space is shown by the numerical estimation of the velocity phase space volume going to zero at certain radial positions that depend on the particular velocity considered. The transition is however continuous due to a radial advection of the markers. This drift is mainly caused by the fact that, in this region, the distribution function of the markers is artificially set to zero for velocities greater than the sampling limit velocity, Eq. (5.19).

Let us now consider the results at the end of the simulation,  $t = 1.5 \cdot 10^6 [\Omega_{ci}^{-1}]$  (lower row). Three main sampling deteriorations can be observed. First, at  $s \approx 0$  and for  $\tilde{v}^2 > 100$ , there is an over-estimation of the phase space volume due to markers having drifted radially towards the magnetic axis. Note that in this region the statistics are usually very poor due to the small bin volumes. Furthermore, the main differences are at very high velocities where the particle distribution functions are exponentially close to zero, minimizing the effect of this poor sampling. The second region where the sampling is affected is near the outer radial edge where there is also an accumulation of markers near  $\tilde{v}^2 \approx 25$ . This is certainly due to the boundary conditions applied to the leaving markers, i.e. their weight is set to zero and there could be a marker accumulation near the radial edge. Note that these two regions where the sampling deteriorates are also observed for GD simulations where no density problem is observed. They could become problematic at some point but we do not expect them to have a significant influence on the density loss. Finally, the sampling is clearly affected in the center of the radial domain where the FHS is active. The error in the marker distribution follows the functional form of the FHS, i.e. the derivative of the Maxwellian with respect to the temperature, see Figure 5.1 (a) for an example. The transition between a negative and positive volume difference is at  $\tilde{v}^2 \approx 25$  which corresponds to the local thermal ion velocity  $v/v_{th}(s=0.5) = 1$ . We recall that the purpose of the FHS is to heat the plasma by transferring the particles from sub-thermal to supra-thermal regions. From the marker's perspective, this corresponds to decreasing the weight in sub-thermal regions and increasing it in supra-thermal regions. The problem in evaluating the background density is that when multiplied by  $f_0$ , Eq. (5.13), which decays exponentially with the velocity, the region where there is an over sampling—the supra-thermal region—is under evaluated and does not compensate for the region with an under sampling—the sub-thermal region—even though the total velocity volume is verified to be conserved. This is further amplified by the fact that  $f_0$  is supposed constant in time and does not follow the evolution of the temperature. This suggests to use an adaptative- $f_0$  scheme. To further verify that sampling is indeed the cause of the problem, the same simulation has been done but with four times more particles, i.e. with  $512 \cdot 10^6$  markers. As can be seen in Figure 5.14, the sampling problem is mitigated for similar simulation duration. Even though increasing resolution has apparently solved the sampling problem, the mechanisms that lead to the undersampling of low velocity regions around the heat source radial position is still not understood and the



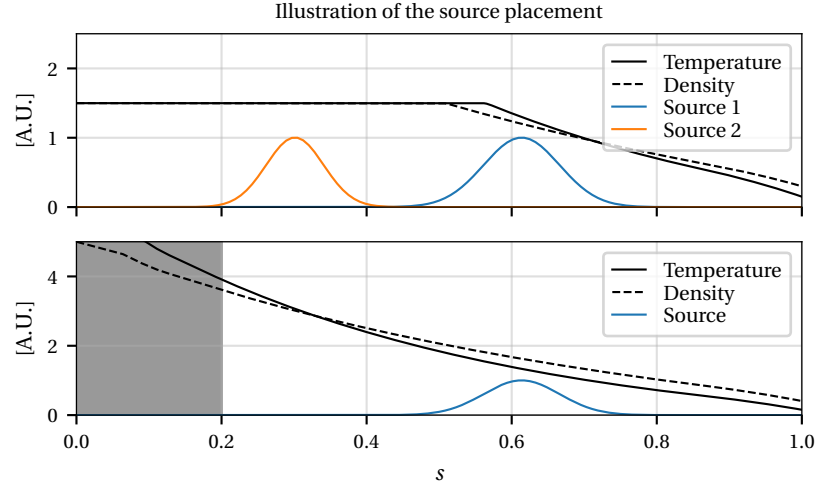
problem may be pushed to later simulation times.



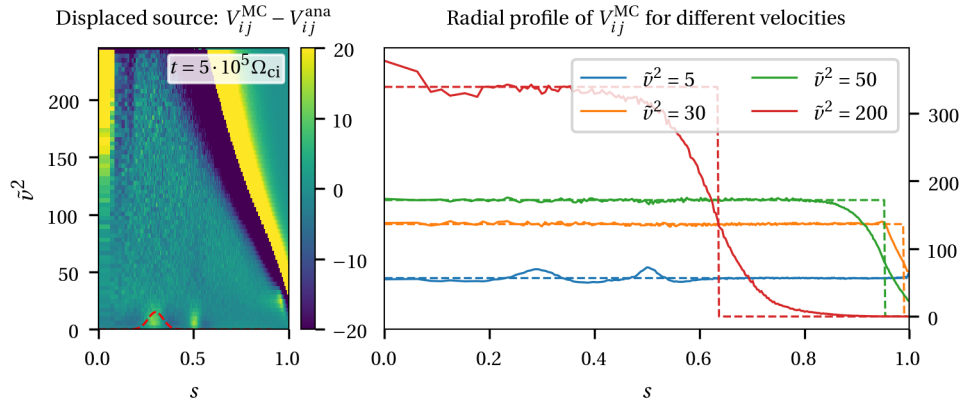
**Figure 5.14** : Same as Figure 5.13, but with  $512 \cdot 10^6$  markers instead of  $128 \cdot 10^6$ .

In the previous run, the FHS was placed so that part of it lay in the region where there are no density and temperature gradients, i.e. where no turbulence develops, and the majority of it was in the region with logarithmically-constant gradients, see “Source 1” in Figure 5.15 (top). Another simulation is carried out, in which the FHS position is changed so that the source is entirely in the region where both density and temperature have no gradients, see “Source 2” in Figure 5.15 (top). This is done to see the influence of the source in regions where there are no heat transfer mechanisms. Note that this kind of profile with flat regions mimicking the sawtooth inversion region is anyway not consistent with the physical model implemented in ORB5 since no MHD modes are described. The results of the simulations with “Source 2” are presented in Figure 5.16. Having shifted the heat source into the gradient-free region has clearly affected the simulation. There is now an overestimation of the phase space volume at the source location,  $s \approx 0.3$ , and also where the density gradient becomes non zero,  $s \approx 0.5$ . They both lead to an over estimation of the background density at those positions. From the previous two simulations, the conclusion is that the gradient-free profiles imposed to mimic the sawtooth inversion region are incompatible with the FHS as implemented in ORB5.

For the next simulation, the flat-profile is removed and the radial domain is now taken

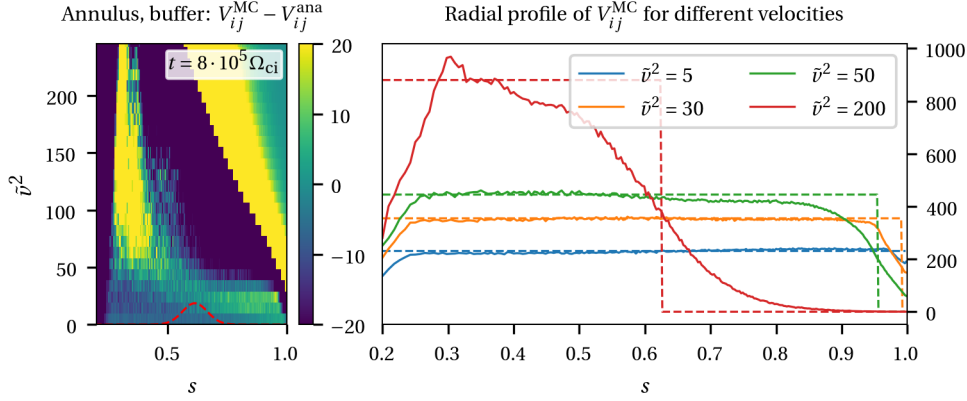


**Figure 5.15 :** Illustration of the heat source radial position for the different runs. The top row shows the original position, presented in Figure 5.4, corresponding to the GD Krook mapping (Source 1) and the position in the gradient-free region (Source 2). The bottom plot shows the case of the annular simulation. The source position is the same as “Source 1” from the top plot and the density and temperature profiles are adapted so that there is no zero-gradient region. The radial interval  $s \in [0.0, 0.2]$ , in grey, is omitted from the annular simulation. Note that all the profiles are arbitrarily rescaled to fit in the plot.



**Figure 5.16 :** Similar as Figure 5.13, but for the case of the FHS entirely placed in the region where there are no density nor temperature gradients (“Source 2” in Figure 5.15).

to be an annulus ranging from  $s = 0.2$  to  $s = 1.0$ , Figure 5.15 (bottom). Homogeneous Dirichlet boundary conditions are applied for the fields at both radial edges. The results of this

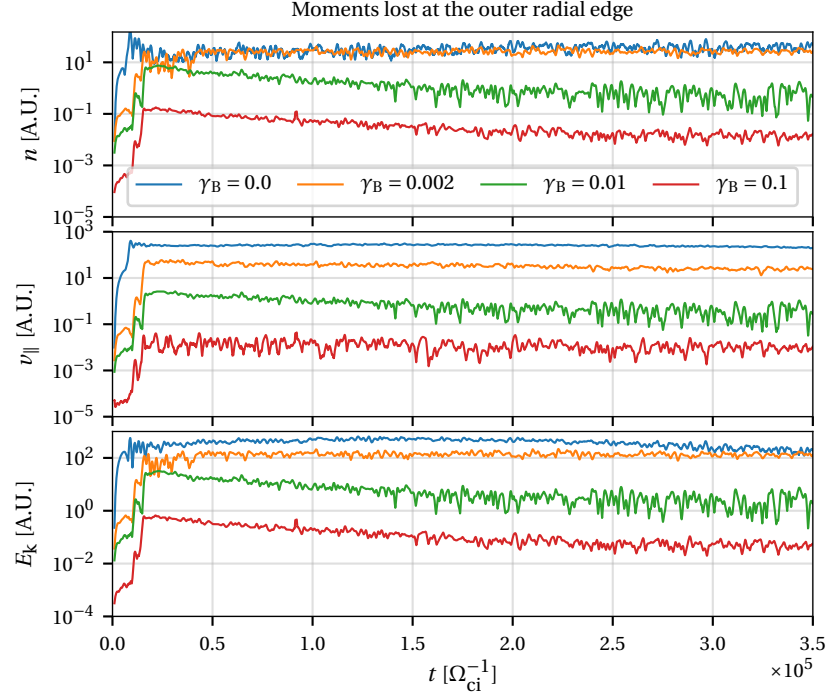


**Figure 5.17** : Similar as Figure 5.13, but for the case of an annular radial domain without flat profile region, Figure 5.15 (lower).

simulation are found in Figure 5.17. The loss of sampling between  $s = 0.2$  and  $s = 0.3$  is due to the boundary condition applied to the markers leaving the annulus through the inner radial boundary, similarly to the situation at the outer boundary,  $s = 1.0$ . This obviously also affects the density (and other moments) profile but it was checked that the accumulated sum of the density loss at  $s = 0.2$  is equal to the missing density near the inner radial edge. In this simulation, the FHS does not influence the sampling properties nor the density profile, which confirms the conclusion of the previous experiment that those background density and temperature profiles containing zero-gradient regions are not compatible with FD simulations especially for the case of a fixed-in-time  $f_0$  background distribution function.

One now wants to study a different heat evacuation mechanism, namely the Krook buffer. To this end, a similar simulation as the original case is carried out, i.e. the density and temperature profiles are those presented in Figure 5.15 with the “Source 1” radial profile for the heat source. The heat sink is substituted by the Krook buffer described by Eq. (5.7), which is less prone to critical issues such as producing negative distribution functions. In this simulation, the buffer region extends from  $s = 0.9$  to  $s = 1.0$ . First, a scan in the buffer rate  $\gamma_B$  is made to find a sufficiently high value to absorb the injected power by damping the distribution fluctuations. To verify the damping efficiency, the density, parallel flows, and kinetic energy that exit the radial domain at the outer edge are plotted in Figure 5.18 for the different values of  $\gamma_B$ . The lowest, non-zero  $\gamma_B$  value of the scan corresponds to roughly 10 times the value of the Krook noise control; the buffer is thus strongly damping the perturbations. For reference, a case without buffer is also shown ( $\gamma_B = 0$ ).

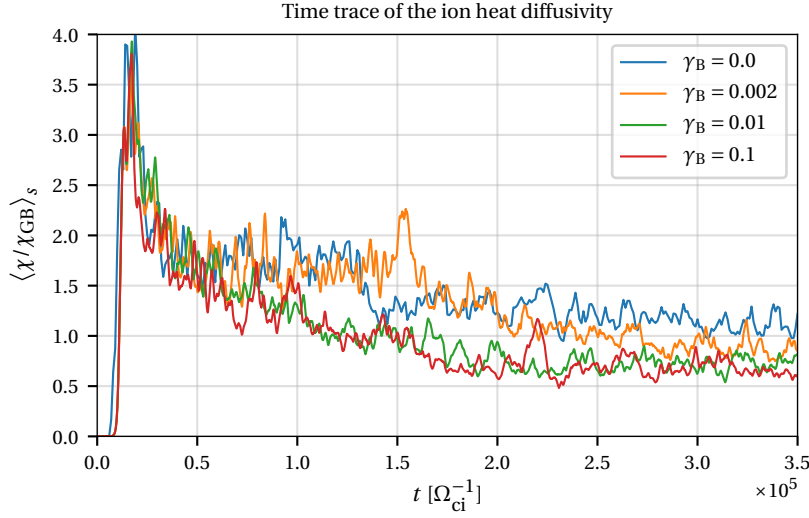
The effect of the buffer on the density and kinetic energy is small for  $\gamma_B = 0.002$ . For higher values, there is an effective reduction of all the edge moment losses. Keeping in mind



**Figure 5.18 :** Density, parallel flows and kinetic energy exiting the domain per unit time at the outer radial edge for different values of  $\gamma_B$ . The diagnostic computes the corresponding moments from the markers leaving the domain at the outer radial boundary. The buffer region extends from  $s = 0.9$  to  $s = 1.0$ .

that the buffer is damping the perturbed distribution function  $\delta f$ , which could artificially change the plasma properties, the aim of this scan is to find a value of  $\gamma_B$  sufficiently high to evacuate all the injected heat within the buffer while not affecting too much the turbulence outside the buffer region. To verify this point, the effective ion heat diffusivity as a function of time is plotted in Figure 5.19. The radial average of the heat diffusivity is done in such a way as to exclude the buffer region, i.e. considers  $s \in [0.7, 0.9]$ . As expected, the buffer affects the transport properties in the core through non-local effects [142]. Indeed, the heat diffusivities are reduced compared to the reference case ( $\gamma_B = 0$ ) by 20% for  $\gamma_B = 0.002$  and 40% for  $\gamma_B = 0.01$  and  $\gamma_B = 0.1$ . These estimations are made based on a time average of the diffusivities for  $t \in [2, 3.5] \cdot 10^5 [\Omega_{ci}^{-1}]$ . There also seems to be a critical value for the buffer rate above which no significant influence on the core heat transport is observed, as already pointed out in Ref. [132] for a similar experiment. The main difference between this work and Ref. [132] is that the buffer considered here also damps the zonal flows. An extension to these simulations would be to change the buffer to be flow-preserving to see if there is a shift in the critical value of the buffer rate due to the nonlinear stabilization of the modes by the zonal

flows. For the remainder of this section, all the simulations where the buffer is used are done



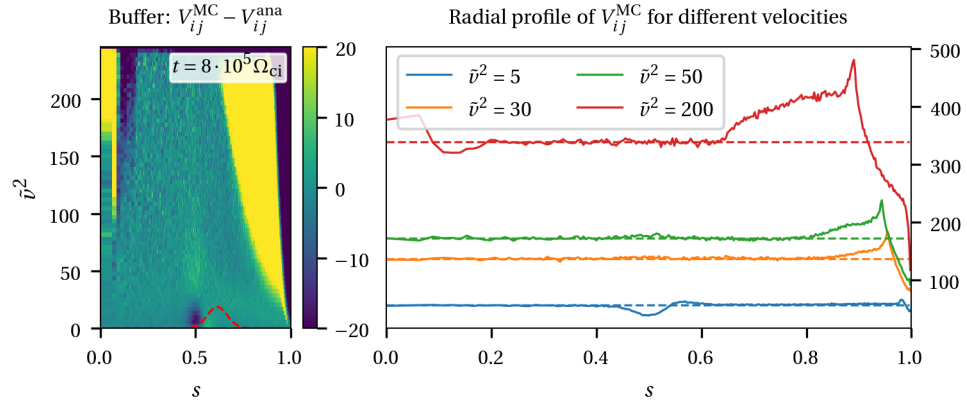
**Figure 5.19** : Effective ion heat diffusivity in gyro-Bohm units versus time. The heat diffusivity is averaged radially over  $0.7 \leq s \leq 0.9$ . The lines show the results with different buffer rates and with the same buffer region from  $s = 0.9$  to the edge.

with the maximum rate tested in the previous scan, i.e.  $\gamma_B = 0.1$ .

The effect of the FHS in combination with the buffer on the phase space sampling is plotted in Figure 5.20. In this simulation, a global value  $\nu_{th}$  corresponding to  $s = 0$  is used for the marker loading, i.e. for all the radial positions, the markers are loaded up to  $\nu_{max}^2 = \kappa_v^2 T_i(s = 0) / m_i$ . This is done to make sure that the phase space is homogeneously sampled throughout the plasma minor radius. Indeed, with the type of temperature profiles that we use, the velocity sampling limit is typically  $\sim 3$  times higher in the core than in the pedestal. Markers can then drift radially in regions where there is no sampling thus degrading the overall sampling by “dilution”. The presence of the buffer does not affect the sampling properties in the edge. The large change of volume that can be observed in the outer part of the radial domain and for high velocities is due to the marker boundary conditions as for the previous cases. It is however amplified because of the global loading of the markers. Indeed, particles with higher velocities have larger radial excursions and thus tend to exit the domain from deeper within the core plasma. It has been checked that similar observations are obtained without the buffer.

## 5.4 Summary of the chapter

In this chapter, the methodology used to carry out FD simulations using the ORB5 code has been presented. All the tested heat sources and sinks are summarized and different nonlinear simulations of a TCV-like setup are made to test them.



**Figure 5.20** : Similar as Figure 5.13, but for the case of the FHS with a buffer at the radial edge to evacuate the injected heat and using the “Source 1” radial profile for the heat source, Figure 5.15.

The so-called fixed heat source is proportional to the derivative of the Maxwellian with respect to the background temperature. The functional form of this heat source as well as an ad-hoc correction term to address sampling errors ensure that the FHS only injects heat and conserves density, parallel momentum, and zonal flows in radial bins up to machine precision. Flux-driven simulations are made with the FHS using the mode-switching method which consists of making a FD continuation of a GD run using a fitting of the effective Krook heat source radial profile from the first GD run as a radial profile for the FHS. It has been tested that switching occurs smoothly and that the results are consistent with FD-only simulations using the same radial heating profile. The use of the FHS evidenced a sampling problem leading to errors in the evaluation of the background density. It is shown that density and temperature profiles presenting zero-gradient regions are incompatible with FD simulations in ORB5. It is then concluded that the heat source should not be placed in zero-gradient regions. Indeed, the absence of a transport mechanisms to evacuate the injected heat and the constant-in-time background distribution function may be the cause. The cause of the density problem is an undersampling of the low velocity regions due to a drift in velocity of the markers. However, the cause of such a drift is still not understood. As a future work, we propose to implement an adaptive- $f_0$  method.

Two heat sinks are tested. The first consists of a FHS with negative rate, which is set such that the total injected heat is zero. The second is a buffer region using a simple Krook operator to damp the perturbed distribution function. The fixed heat sink has a major issue as it does not ensure the positivity of the total distribution function. Furthermore, it is difficult to make sure that the injected heat is actually totally evacuated by the sink due to discretization errors. We think it is safer and more convenient to use a buffer at the outer radial region even if it slightly affects the core transport due to non-local effects.

Finally, an adaptive Krook heat source is developed. It consists of a Krook heating operator

automatically adjusting its rate to inject a fixed power. It is a variant of the standard Krook operator that is used as a heat source for GD simulations but, for which, the input power is not known *a priori* since it depends on the value of  $\delta f$ . A first non conclusive test has been made but further investigations are left for a future work.





## 6 Conclusion and Outlook

### 6.1 Summary

The main goal of this thesis was to enable the ORB5 code for flux-driven simulations in view of profile stiffness investigation and comparison with experiments performed on the TCV tokamak. In particular, to reproduce numerically the shots presented in [49] and study why TCV shots with different heating powers presented a core region where temperature and density profiles exhibit a constant logarithmic gradient and a pedestal region—even in L-mode—with constant profile gradients. In view of this study, three main objectives have been defined for this thesis.

Flux-driven driven simulations are very demanding in terms of computational resources, especially with kinetic electrons required to simulate the TEM-dominant TCV shots. The first goal of this work consisted in a complete refactoring of the ORB5 code followed by an optimization procedure to improve the code performance and port it to multicore CPUs, adding multithreading to the existing MPI parallelization using OpenMP. The refactoring process consisted of a code cleaning and modularization as well as the implementation of an automated testing setup that allows for the testing of the whole code in a few minutes. For the optimization process, we first developed a test bed called `PIC_ENGINE` that contains only the key elements of the PIC algorithm. This code has allowed us to quickly and easily implement various algorithms and parallelization schemes, and make various parameter scans to determine the optimal configurations. The more promising schemes have then been implemented in ORB5. As an example, we presented the color scheme, which is an efficient method for the charge/current deposition step, and which improves the performance by a factor 2.4 compared to the ORB5 original deposition routine.

The TCV shots presented in [49] are TEM-dominant. To simulate them, a model including at least a kinetic response of the trapped electrons is required—a model with fully kinetic electrons can also be used but is generally more expensive in numerical resources. Such a hybrid model, in which passing electrons have an adiabatic response and trapped electrons are kinetic, was already implemented in ORB5. However, this model is not valid in the nonlinear regime. Indeed, it does not satisfy the ambipolarity condition, which also leads to a non-

conservation of the toroidal momentum. This is because it creates spurious sources/sinks of particles, energy, and momentum due to the (de)trapping processes that are not well described. A corrected version has been implemented in the code. The main change compared to the original ORB5 hybrid electron model is that it accounts for a kinetic response from the flux-surface average of the passing electrons. As a first approximation, a model in which the flux-surface average is replaced by a Fourier filter retaining only the  $n = m = 0$  contribution from the passing electrons is implemented. In addition to solving the two main issues of the original model, this improved version also has a correct description of the GAM frequency and damping rate. In the nonlinear regime, it was observed that fine structures of the  $E \times B$  shearing rate are formed around  $q = 1$  surface. Although not fully understood yet, we suspect that passing electrons with  $n \neq 0$ , which are not taken into account kinetically in the quasineutrality equation interact with the  $n = m = 0$  contribution through nonlinear coupling. Furthermore, simulations using TCV parameters have shown a transition from TEM-dominant transport to an ITG-TEM mix. This was attributed to the density profile relaxation that linearly destabilizes low- $n$  ion modes and nonlinearly reduces the level of zonal flow shearing rate, thus increasing ion heat transport.

Finally, a heat source able to inject a fixed amount of power while conserving, up to machine precision, other moments such as density and parallel momentum has been implemented in ORB5, thus enabling flux-driven simulations. To avoid long simulations of the order of the profile relaxation time, which are expensive numerically and can be problematic with respect to the  $\delta f$  PIC method used in ORB5, we have tested a “mode switching” approach, which consists of initializing a gradient-driven simulation from which we define an effective heat source profile that is used as an input for the flux-driven simulation. The simulations using mode switching have been compared to gradient-driven-only and flux-driven-only runs and they all showed similar statistical quasi-steady states. It was noted, however, that  $E \times B$  shearing rate structures appearing in the early phase of the gradient-driven run persisted even in the flux-driven continuation thus slightly modifying transport properties compared to a flux-driven-only run. The fixed heat source implemented in this work showed density conservation problems for very long simulations if the order of the confinement time. Indeed, even though it was checked that no density is injected by the source up to machine precision, a radially localized non-conservation of density coming from the  $f_0$  contribution was observed. It was attributed to a sampling problem coming from the fact that the fixed heat source was used in combination with background profiles that are not consistent with the gyrokinetic model implemented in ORB5. These profiles, inspired by experiments shown in [49], have a flat region supposed to mimic the sawtooth activity. In this region, we suspect that accumulation of numerical errors in the marker orbits lead to an under sampling in some regions of the phase space. It has to be noted that the problem occurred only on the density in our simulations. However, it may well also appear on higher-order moments for even longer simulations. Various types of heat sources and sinks have also been tested but none of them

showed satisfactory conclusions with respect to the sampling problem. Simulations where the problematic flat regions were artificially removed were also carried out and they showed no density conservation problem, thus confirming the conclusion that one should avoid injecting heat power in a region of the plasma where the model used does not include a mechanism for transporting this heat out of the injection region.

## 6.2 Outlook

### 6.2.1 Refactoring and optimization of ORB5

The optimization of the ORB5 code was made in such a way to obtain the best performance with as little complexity as possible. The main reason for this choice was the portability of the code between HPC platforms but also between scientists that do not have the same knowledge in computational sciences. Even though the major parts of the code have been optimized, there are still a few of them to be further refactored/optimized. For example, the sorting routine has allowed us to use various algorithms, such as the color scheme, that improve the code performance. However, it was noted that the sorting cost is, in general, comparable to the time gain, leading to no significant overall performance improvement. Worse, in some cases, the sorting cost can even be greater than the overall time gain. More efficient sorting algorithms or a different approach, in which particles are sorted only when they have actually moved to another grid cell, may be tested. Also, it has clearly been observed that the optimal setup heavily depends on various parameters such as the number of clones, subdomains, OpenMP threads, etc. In order to simplify as much as possible the use of the code, an automatic performance scheduler can be implemented. It would consist of automatically testing various parallelization configurations for a few timesteps and then selecting the best one for the rest of the production run.

The other main future task is the porting of ORB5 to GPUs<sup>1</sup>. To this end, the diagnostic routines have to be modularized. Furthermore, the in/output routines used to write diagnostic data to disk may be parallelized. Indeed, for the moment only one MPI task is writing the data. This is not problematic as the time spent for writing is negligible compared to the computational time, but with further optimizations and the porting to GPUs, this writing time will become an important fraction of the total time.

Finally, concerning the refactoring and testing of the code, two main future lines of action can be identified. First, the current way of creating checkpoint files for a simulation restart depends heavily on the parallelization setup and the HPC platforms. This makes impossible to continue simulations on other machines or with different parallelization, which can lead to the loss of some runs. To avoid that, the routine creating the checkpoint files should be changed in order to drop those aforementioned constraints. Second, the current testing setup runs various scaled-down physically-relevant simulations and checks for changes with the

---

<sup>1</sup>At the time of writing, most of the ORB5 code has been ported to GPU [101], in a work carried out as part of another Ph.D. project

previous version of the code. Even though it is a very useful warning signal, this testing method requires someone to analyze the eventual differences and decide if they are expected and justified or not. Furthermore, when implementing new physics into the code, it can become difficult to discern the differences coming from the change in physical model or coming from implementation bugs. To address this issue, more robust tests should be implemented. Going toward unit-testing, these tests should be able to check independently small portions of the program and in a univocal way.

### 6.2.2 Hybrid electron model

The correction to the former hybrid electron model in ORB5 has given satisfactory results with respect to the linear and nonlinear properties. However, only the version approximating the flux-surface average by a  $n = m = 0$  Fourier filter was tested nonlinearly. Extended nonlinear tests for the flux-surface-average version of the corrected model should be carried out.

Further investigation of the formation of the fine radial structures around  $q = 1$  will be carried out in future work. In particular, we will study how the nonlinear coupling through the electric field between adiabatic and kinetic passing electrons acts on those radial structures. Comparison with runs using fully kinetic electrons are also envisaged.

### 6.2.3 Flux-driven simulations

The flux-driven setup presented in this work showed a non-conservation problem of the various moments, in particular the density. Even though some understanding of the causes of this problem was achieved during this work, a complete understanding of what are the exact causes is still missing. Further investigations are required.

The flux-driven capabilities of ORB5 are limited by the  $\delta f$  PIC method with constant background  $f_0$ . Indeed, the final state of the system must not be too far from the initial one if one wants to keep the numerical noise as low as possible. This constraint could be lifted by implementing a  $\delta f$  PIC method with evolving background schemes [68].

Currently, the heat buffer is placed on the radial outer part of the domain. In and in the vicinity of this small region, transport properties are artificially affected by the naive Krook operator used in the buffer. This can make studies of the pedestal profiles difficult. One of the possible extensions of the code is to artificially extend the radial domain beyond the last closed flux surface and put all the buffers in this region.

### 6.2.4 General outlook

The ORB5 code is now fully equipped to make flux-driven simulations using hybrid electrons relevant for the study of profile stiffness as presented in [49]. Various power scans, similar to the gradient scans presented in [131], can be carried out to study how the background profile gradients evolve as a function of the heat power input.

# A Parallelization algorithms

In this appendix, we present the different parallelizations methods we have implemented in the PIC\_ENGINE for the sorting and the charge deposition already briefly presented in Sections 3.3.2.2 and 3.3.2.3, respectively. The algorithms are presented in Fortran-like pseudo-code that we have intentionally simplified to keep only the relevant information. They may not work as they are but only the parallelization technique is emphasized here.

## A.1 Sorting

We have seen in Algorithm 1 in Section 3.3.2.2 that the counting sort used in the PIC\_ENGINE consists mainly in three loops, two over the particles to create a histogram and put them in their right place and one over the buckets for the prefix sum. Note that in all versions, the prefix sum is never parallelized because it takes a negligible time in our application. Here are the two different methods discussed in this paper:

**psort\_1:** In this method, the first loop is parallelized using OpenMP reduction and the third loop is done with a OpenMP do loop and atomic directives to avoid race conditions. Furthermore, in the first loop, the displs array is calculated as the count array of particles in each bucket. It is then overwritten by the *inclusive* scan in the second loop.

```
1 displs(:)=0
2 !$Omp Parallel &
3 !$Omp Private(ind1,ind2,ind)
4 !$Omp Do Reduction(+:displs)
5 Do i=1,np
6   ! Compute the particle grid-cell
7   ! index
8   indX=Int(part_att(i,dirX)/dx)+1
9   indY=Int(part_att(i,dirY)/dy)+1
10  ind=(indY-1)*nbX+indX
11  ! Create the histogram
```

## Appendix A. Parallelization algorithms

---

```
12     displs(ind)=displs(ind)+1
13 End Do
14 !$Omp End Do
15
16 !$Omp Single
17 ! Prefix sum
18 Do i=2,nbX*nbY
19     displs(i)=displs(i)+displs(i-1)
20 End Do
21 !$Omp End Single
22
23 !$Omp Do Private(is)
24 Do i=np,1,-1
25     ! Compute the particle grid-cell
26     ! index
27     indX=Int(part_att(i,dirX)/dx)+1
28     indY=Int(part_att(i,dirY)/dy)+1
29     ind=(indY-1)*nbX+indX
30     ! Update atomically to avoid race
31     ! conditions
32     !$Omp Atomic Capture
33     is=displs(ind)
34     displs(ind)=displs(ind)-1
35     !$Omp End Atomic
36     part_att_temp(is,:)=part_att(i,:)
37 End Do
38 !$Omp End Do
39 !$Omp End Parallel
```

psort\_2: The second sorting method is slightly different from the first one. During the first loop, we compute the index of each particle in the sorted array, and race conditions are avoided with OpenMP atomic. By doing so, the third loop is trivially parallelized with OpenMP do loop.

```
1  !$Omp Parallel &
2  !$Omp Private(is,ind1,ind2,tmp)
3  !$Omp Do
4  Do i=1,np
5      ! Compute the particle grid-cell
6      ! index
```

```

7     indX=Int(part_att(i,dirX)/dx)+1
8     indY=Int(part_att(i,dirY)/dy)+1
9     ind(i)=(indY-1)*nbX+indX
10    is=ind(i)
11    !$Omp Atomic Capture
12    tmp=counts(is)
13    ! Create the histogram
14    counts(is)=counts(is)+1
15    !$Omp End Atomic
16    indices(i)=tmp
17  End Do
18  !$Omp End Do
19
20  !$Omp Single
21  displs(1)=counts(1)
22  Do i=2,nb1*nb2
23    displs(i)=displs(i-1)+counts(i)
24  End Do
25  !$Omp End Single
26
27  !$Omp Do
28  Do i=1,np
29    is=ind(i)
30    part_att_temp(displs(is)- &
31      & indices(i),:)=part_att(i,:)
32  End Do
33  !$Omp End Do
34  !$Omp End Parallel

```

## A.2 Charge assignment

Similarly to the sorting, we present here different charge assignment methods implemented in the PIC\_ENGINE. The four versions of setrho are:

**setrho\_1:** In this first method, we mimic the OpenMP reduction with an in-house algorithm inspired by the OpenMP standard [143].

```

1  !$Omp Parallel &
2  !$Omp Private(tid,ix,iy,iz,wx,wy,wz)
3  ! Get the thread number
4  tid=omp_get_thread_num()

```

## Appendix A. Parallelization algorithms

---

```
5
6  !$Omp Do
7  Do ip=1,np
8      ! Compute the particle relative position
9      ix=Int(part_att(ip,1)-xgrid(0))
10     wx=part_att(ip,1)-xgrid(ix)
11     ! Do similarly for y and z
12     ...
13
14     ! Deposit the charge according to
15     ! the relative positions
16     rho_loc(ix,iy,iz,tid)= &
17         & rho_loc(ix,iy,iz,tid)+ &
18         & (1.0-wx)*(1.0-wy)*(1.0-wz)
19     ! Do similarly for the other seven
20     ! grid points
21     ...
22 End Do
23 !$Omp End Do
24
25 ! Perform the reduction
26 !$Omp Do Collapse(3)
27 Do iz=0,nz
28     Do iy=0,ny
29         Do ix=0,nx
30             rho(ix,iy,iz)= &
31                 & Sum(rho_loc(ix,iy,iz,:))
32         End Do
33     End Do
34 End Do
35 !$Omp End Do
36 !$Omp End Parallel
```

setrho\_2: The second method uses the OpenMP `atomic` directive to safely update the charge array.

```
1  !$Omp Parallel &
2  !$Omp Private(ix,iy,iz,wx,wy,wz,ip)
3  !$Omp Do
4  Do ip=1,np
```



```

5      ! Compute the particle relative position
6      ix=Int(part_att(ip,1)-xgrid(0))
7      wx=part_att(ip,1)-xgrid(ix)
8      ! Do similarly for y and z
9      ...
10
11     ! Deposit the charge according to
12     ! the relative positions
13     !$Omp Atomic Update
14     rho(ix,iy,iz)=rho(ix,iy,iz)+ &
15         & (1.0-wx)*(1.0-wy)*(1.0-wz)
16     ! Do similarly for the other seven
17     ! grid points
18     ...
19 End Do
20 !$Omp End Do
21 !$Omp End Parallel

```

setrho\_3: Similarly to the first method, we use this time the OpenMP reduction to avoid race conditions when updating the charge array.

```

1  !$Omp Parallel Private(ix,iy,iz)
2  !$Omp Do Private(wx,wy,wz,inb,ic) &
3  !$Omp Reduction(+:rho)
4  Do ip=1,np
5      ! Compute the particle relative position
6      ix=Int(part_att(ip,1)-xgrid(0))
7      wx=part_att(ip,1)-xgrid(ix)
8      ! Do similarly for y and z
9      ...
10
11     ! Deposit the charge according to
12     ! the relative positions
13     rho(ix,iy,iz)=rho(ix,iy,iz)+ &
14         & (1.0-wx)*(1.0-wy)*(1.0-wz)
15     ! Do similarly for the other seven
16     ! grid points
17     ...
18 End Do
19 !$Omp End Do

```

## Appendix A. Parallelization algorithms

---

```
20 !$Omp End Parallel
```

setrho\_4: Finally, in this last method we use the finite support of the CIC method. Indeed, here each particle will only contribute to the eight nearest grid points.

```
1  !$Omp Do Parallel Collapse(3) &
2  !$Omp Private(ix,iy,iz,d,ibin) &
3  !$Omp Private(jbin,wx,wy,wz)
4  Do iz=0,nz
5    Do iy=0,ny
6      Do ix=0,nx
7        d=0.0
8        Do jbin=Max(1,iy),Min(ny,iy+1)
9          Do ibin=Max(1,ix),Min(nx,ix+1)
10         bin=(jbin-1)*nx+ibin
11         Do ip=displs(bin)+1, &
12           & displs(bin+1)
13           wx=part_att(ip,1)-xgrid(ix)
14           wy=part_att(ip,2)-ygrid(iy)
15           wz=part_att(ip,3)-zgrid(iz)
16           d=d+Min(1.0-wx,1.0+wx)* &
17             & Min(1.0-wy,1.0+wy)* &
18             & Min(1.0-wz,1.0+wz)
19         End Do
20       End Do
21     End Do
22     rho(ix,iy,iz) = d
23   End Do
24 End Do
25 End Do
26 !$Omp End Parallel Do
```

## Bibliography

- [1] UN, “United Nations Development Programme,” 2018. <http://hdr.undp.org/en/content/human-development-index-hdi>
- [2] IEA, “The World Bank,” 2014. <https://data.worldbank.org/indicator/EG.USE.PCAP.KG.OE>
- [3] IEA, “Energy snapshot,” 2017. <https://www.iea.org/newsroom/energysnapshots/oecd-electricity-production-by-source-1974-2016.html>
- [4] J. D. Lawson, “Some criteria for a power producing thermonuclear reactor,” *Proc. Phys. Soc. Sect. B*, vol. 70, no. 1, pp. 6–10, jan 1957. <http://stacks.iop.org/0370-1301/70/i=1/a=303?key=crossref.e1697fd64b8a71aa295438657a972463>
- [5] F. Troyon, R. Gruber, H. Saurenmann, S. Semenzato, and S. Succi, “MHD-Limits to Plasma Confinement,” *Plasma Phys. Control. Fusion*, vol. 26, no. 1A, pp. 209–215, jan 1984. <http://stacks.iop.org/0741-3335/26/i=1A/a=319?key=crossref.cae719754c2ff814c52af095f5977ea4>
- [6] M. Greenwald, “Density limits in toroidal plasmas,” p. 201, aug 2002. <http://stacks.iop.org/0741-3335/44/i=8/a=201?key=crossref.10b223c9bfedf85db5ed007f2fdae8a5>
- [7] J. Candy, R. E. Waltz, and W. Dorland, “The local limit of global gyrokinetic simulations,” *Phys. Plasmas*, vol. 11, no. 5, pp. L25–L28, may 2004. <http://aip.scitation.org/doi/10.1063/1.1695358>
- [8] B. F. McMillan, S. Jolliet, A. Bottino, P. Angelino, T. M. Tran, and L. Villard, “Rapid Fourier space solution of linear partial integro-differential equations in toroidal magnetic confinement geometries,” *Comput. Phys. Commun.*, vol. 181, no. 4, pp. 715–719, apr 2010. <http://linkinghub.elsevier.com/retrieve/pii/S0010465509003920http://www.sciencedirect.com/science/article/pii/S0010465509003920>
- [9] G. Merlo, S. Brunner, Z. Huang, S. Coda, T. Görler, L. Villard, A. B. Navarro, J. Dominski, M. Fontana, F. Jenko, L. Porte, and D. Told, “Investigating the radial structure of axisymmetric fluctuations in the TCV tokamak with local and global

- gyrokinetic GENE simulations,” *Plasma Phys. Control. Fusion*, vol. 60, no. 3, p. 034003, mar 2018. <http://stacks.iop.org/0741-3335/60/i=3/a=034003?key=crossref.520dac020c8041b000ba9fe2457e3aa9>
- [10] E. Lanti, N. Ohana, N. Tronko, A. Bottino, B. F. Mcmillan, A. Mishchenko, A. Scheinberg, A. Biancalani, P. Angelino, S. Brunner, J. Dominski, P. Donnel, C. Gheller, R. Hatzky, A. Jocksch, S. Joliet, J. P. M. Collar, I. Novikau, T. Vernay, and L. Villard, “Orb5 : a global electromagnetic gyrokinetic code using the PIC approach in toroidal geometry,” *Comput. Phys. Commun.*, no. Submitted, pp. 1–50, 2019.
- [11] Y. Idomura, S. Tokuda, and Y. Kishimoto, “Global gyrokinetic simulation of ion temperature gradient driven turbulence in plasmas using a canonical Maxwellian distribution,” *Nucl. Fusion*, vol. 43, no. 4, pp. 234–243, apr 2003. <http://iopscience.iop.org/article/10.1088/0029-5515/43/4/303http://stacks.iop.org/0029-5515/43/i=4/a=303?key=crossref.d169703164fcc8d437dddf6f6a74be33>
- [12] S. Ku, C. S. Chang, R. Hager, R. M. Churchill, G. R. Tynan, I. Cziegler, M. Greenwald, J. Hughes, S. Parker, M. F. Adams, E. D’Azevedo, and P. Worley, “A fast low-to-high confinement mode bifurcation dynamics in the boundary-plasma gyrokinetic code XGC1,” *Phys. Plasmas*, vol. 25, no. 5, p. 056107, may 2018. <http://aip.scitation.org/doi/10.1063/1.5020792>
- [13] Z. Lin, “Turbulent Transport Reduction by Zonal Flows: Massively Parallel Simulations,” *Science (80-. )*, vol. 281, no. 5384, pp. 1835–1837, sep 1998. <http://science.sciencemag.org/content/281/5384/1835.abstract>
- [14] W. X. Wang, T. S. Hahm, W. W. Lee, G. Rewoldt, J. Manickam, and W. M. Tang, “Nonlocal properties of gyrokinetic turbulence and the role of ExB flow shear,” *Phys. Plasmas*, vol. 14, no. 7, p. 072306, jul 2007. <http://aip.scitation.org/doi/10.1063/1.2750647>
- [15] W. X. Wang, P. H. Diamond, T. S. Hahm, S. Ethier, G. Rewoldt, and W. M. Tang, “Nonlinear flow generation by electrostatic turbulence in tokamaks,” *Phys. Plasmas*, vol. 17, no. 7, p. 072511, jul 2010. <http://aip.scitation.org/doi/10.1063/1.3459096>
- [16] T. Korpilo, A. Gurchenko, E. Gusakov, J. Heikkinen, S. Janhunen, T. Kiviniemi, S. Leerink, P. Niskala, and A. Perevalov, “Gyrokinetic full-torus simulations of ohmic tokamak plasmas in circular limiter configuration,” *Comput. Phys. Commun.*, vol. 203, pp. 128–137, jun 2016. <https://www.sciencedirect.com/science/article/pii/S0010465516300364?via%3Dihub>
- [17] A. M. Dimits, T. J. Williams, J. A. Byers, and B. I. Cohen, “Scalings of ion-temperature-gradient-driven anomalous transport in tokamaks,” *Phys. Rev. Lett.*, vol. 77, no. 1, pp. 71–74, jul 1996. <https://link.aps.org/doi/10.1103/PhysRevLett.77.71>

- 
- [18] F. Jenko, W. Dorland, M. Kotschenreuther, and B. N. Rogers, "Electron temperature gradient driven turbulence," *Phys. Plasmas*, vol. 7, no. 5, p. 1904, apr 2000. <http://aip.scitation.org/doi/10.1063/1.874014><http://scitation.aip.org/content/aip/journal/pop/7/5/10.1063/1.874014><https://aip.scitation.org/doi/10.1063/1.874014>
  - [19] Y. Idomura, M. Ida, T. Kano, N. Aiba, and S. Tokuda, "Conservative global gyrokinetic toroidal full-f five-dimensional Vlasov simulation," *Comput. Phys. Commun.*, vol. 179, no. 6, pp. 391–403, sep 2008. <https://www.sciencedirect.com/science/article/pii/S0010465508001409?via%3Dihub>
  - [20] J. Candy and R. E. Waltz, "An Eulerian gyrokinetic-Maxwell solver," *J. Comput. Phys.*, vol. 186, no. 2, pp. 545–581, apr 2003. <https://www.sciencedirect.com/science/article/pii/S0021999103000792><https://www.sciencedirect.com/science/article/pii/S0021999103000792?via%3Dihub>
  - [21] W. Dorland, F. Jenko, M. Kotschenreuther, and B. N. Rogers, "Electron Temperature Gradient Turbulence," *Phys. Rev. Lett.*, vol. 85, no. 26, pp. 5579–5582, dec 2000. <https://link.aps.org/doi/10.1103/PhysRevLett.85.5579>
  - [22] T.-H. H. Watanabe and H. Sugama, "Velocity-space structures of distribution function in toroidal ion temperature gradient turbulence," *Nucl. Fusion*, vol. 46, no. 1, pp. 24–32, jan 2006. <http://stacks.iop.org/0029-5515/46/i=1/a=003?key=crossref.3d7e6f40ce52b566eca20514f822347d>
  - [23] A. G. Peeters and D. Strintzi, "The effect of a uniform radial electric field on the toroidal ion temperature gradient mode," *Phys. Plasmas*, vol. 11, no. 8, pp. 3748–3751, aug 2004. <http://aip.scitation.org/doi/10.1063/1.1762876>
  - [24] V. Grandgirard, J. Abiteboul, J. Bigot, T. Cartier-Michaud, N. Crouseilles, G. Dif-Pradalier, C. Ehrlacher, D. Esteve, X. Garbet, P. Ghendrih, G. Latu, M. Mehrenberger, C. Norscini, C. Passeron, F. Rozar, Y. Sarazin, E. Sonnendrücker, A. Strugarek, and D. Zarzoso, "A 5D gyrokinetic full-f global semi-Lagrangian code for flux-driven ion turbulence simulations," *Comput. Phys. Commun.*, vol. 207, pp. 35–68, oct 2016. <http://linkinghub.elsevier.com/retrieve/pii/S0010465516301230>
  - [25] L. Ye, Y. Xu, X. Xiao, Z. Dai, and S. Wang, "A gyrokinetic continuum code based on the numerical Lie transform (NLT) method," *J. Comput. Phys.*, vol. 316, pp. 180–192, jul 2016. <https://www.sciencedirect.com/science/article/pii/S0021999116300699>
  - [26] H. Lütjens, A. Bondeson, and O. Sauter, "The CHEASE code for toroidal MHD equilibria," *Comput. Phys. Commun.*, vol. 97, no. 3, pp. 219–260, sep 1996. <https://www.sciencedirect.com/science/article/pii/001046559600046X>

- [27] J. P. Graves, “Toroidal drift precession and wave–particle interaction in shaped tokamaks with finite beta and neoclassical equilibrium effects,” *Plasma Phys. Control. Fusion*, vol. 55, no. 7, p. 074009, jul 2013. <http://stacks.iop.org/0741-3335/55/i=7/a=074009?key=crossref.7be268d660557c3df6e0b8a423bfc088>
- [28] A. Burckel, O. Sauter, C. Angioni, J. Candy, E. Fable, and X. Lapillonne, “On the effects of the equilibrium model in gyrokinetic simulations: from s-  $\alpha$  to diverted MHD equilibrium,” *J. Phys. Conf. Ser.*, vol. 260, no. 1, p. 12006, nov 2010. <http://iopscience.iop.org/article/10.1088/1742-6596/260/1/012006http://stacks.iop.org/1742-6596/260/i=1/a=012006?key=crossref.f6b1ebaa153d960a2d4c50609310c96b>
- [29] a. M. Dimits, G. Bateman, M. a. Beer, B. I. Cohen, W. Dorland, G. W. Hammett, C. Kim, J. E. Kinsey, M. Kotschenreuther, a. H. Kritz, L. L. Lao, J. Mandrekas, W. M. Nevins, S. Parker, a. J. Redd, D. E. Shumaker, R. D. Sydora, and J. Weiland, “Comparisons and physics basis of tokamak transport models and turbulence simulations,” *Phys. Plasmas*, vol. 7, no. 3, p. 969, 2000. <http://scitation.aip.org/content/aip/journal/pop/7/3/10.1063/1.873896>
- [30] A. Marinoni, S. Brunner, Y. Camenen, S. Coda, J. P. Graves, X. Lapillonne, A. Pochelon, O. Sauter, and L. Villard, “The effect of plasma triangularity on turbulent transport: modeling TCV experiments by linear and non-linear gyrokinetic simulations,” *Plasma Phys. Control. Fusion*, vol. 51, no. 5, p. 055016, may 2009. <http://stacks.iop.org/0741-3335/51/i=5/a=055016?key=crossref.355c33ed99296dcd501d287b6664f159>
- [31] M. Fontana, L. Porte, S. Coda, O. Sauter, and T. T. Team, “The effect of triangularity on fluctuations in a tokamak plasma,” *Nucl. Fusion*, vol. 58, no. 2, p. 024002, feb 2018. <http://stacks.iop.org/0029-5515/58/i=2/a=024002?key=crossref.67702f45512c4503f416bca563c8f3eb>
- [32] A. Marinoni, M. E. Austin, A. W. Hyatt, M. L. Walker, J. Candy, C. Chrystal, C. J. Lasnier, G. R. McKee, T. Odstřil, C. C. Petty, M. Porkolab, J. C. Rost, O. Sauter, S. P. Smith, G. M. Staebler, C. Sung, K. E. Thome, A. D. Turnbull, L. Zeng, and D.-D. Team, “H-mode grade confinement in L-mode edge plasmas at negative triangularity on DIII-D,” *Phys. Plasmas*, vol. 26, no. 4, p. 042515, apr 2019. <http://aip.scitation.org/doi/10.1063/1.5091802>
- [33] G. Merlo, S. Brunner, O. Sauter, Y. Camenen, T. Görler, F. Jenko, A. Marinoni, D. Told, and L. Villard, “Investigating profile stiffness and critical gradients in shaped TCV discharges using local gyrokinetic simulations of turbulent transport,” *Plasma Phys. Control. Fusion*, vol. 57, no. 5, p. 054010, 2015. <http://stacks.iop.org/0741-3335/57/i=5/a=054010?key=crossref.4a3f89e047ff2e816f6d85c4fece3794>
- [34] S. Brunner, *Waves and Instabilities in Inhomogeneous Plasmas*. EPFL Doctoral School lecture, 2015.

- 
- [35] F. Jenko and W. Dorland, "Prediction of Significant Tokamak Turbulence at Electron Gyroradius Scales," *Phys. Rev. Lett.*, vol. 89, no. 22, p. 225001, nov 2002. <https://link.aps.org/doi/10.1103/PhysRevLett.89.225001>
- [36] B. B. Kadomtsev, O. P. Pogutse, and I. V. Kurchatov, "Trapped particles in toroidal magnetic systems," *Nucl. Fusion*, vol. 11, no. 1, pp. 67–92, jan 1971. <http://stacks.iop.org/0029-5515/11/i=1/a=010?key=crossref.efefb73e27dfcc555a67d6f1a6efb7fa>
- [37] D. R. Ernst, P. T. Bonoli, P. J. Catto, W. Dorland, C. L. Fiore, R. S. Granetz, M. Greenwald, A. E. Hubbard, M. Porkolab, M. H. Redi, J. E. Rice, K. Zhurovich, and A. C.-M. Alcator C-Mod Group, "Role of trapped electron mode turbulence in internal transport barrier control in the Alcator C-Mod Tokamak," *Phys. Plasmas*, vol. 11, no. 5, pp. 2637–2648, may 2004. <http://aip.scitation.org/doi/10.1063/1.1705653>
- [38] T. Dannert and F. Jenko, "Gyrokinetic simulation of collisionless trapped-electron mode turbulence," *Phys. Plasmas*, vol. 12, no. 7, pp. 1–8, jul 2005. <http://aip.scitation.org/doi/10.1063/1.1947447>
- [39] A. G. Peeters, C. Angioni, M. Apostoliceanu, F. Jenko, F. Ryter, and t. A. U. the ASDEX Upgrade team, "Linear gyrokinetic stability calculations of electron heat dominated plasmas in ASDEX Upgrade," *Phys. Plasmas*, vol. 12, no. 2, p. 022505, feb 2005. <http://aip.scitation.org/doi/10.1063/1.1848111>
- [40] J. Lang, S. Parker, and Y. Chen, "Nonlinear saturation of collisionless trapped electron mode turbulence: Zonal flows and zonal density," *Phys. Plasmas*, vol. 15, no. 5, p. 055907, may 2008. <http://aip.scitation.org/doi/10.1063/1.2884036>
- [41] T. Vernay, S. Brunner, L. Villard, B. F. McMillan, S. Jolliet, A. Bottino, T. Görler, and F. Jenko, "Global gyrokinetic simulations of TEM microturbulence," *Plasma Phys. Control. Fusion*, vol. 55, no. 7, p. 074016, jul 2013. <http://stacks.iop.org/0741-3335/55/i=7/a=074016?key=crossref.6e26708d569dcb2dcd6631cc3a584de7>
- [42] S. Brunner, "Global approach to the spectral problem of microinstabilities in Tokamak plasmas using a gyrokinetic model," Ph.D. dissertation, EPFL, 1997. <https://infoscience.epfl.ch/record/32178?ln=en>
- [43] M. N. Rosenbluth and F. L. Hinton, "Poloidal Flow Driven by Ion-Temperature-Gradient Turbulence in Tokamaks," *Phys. Rev. Lett.*, vol. 80, no. 4, pp. 724–727, jan 1998. <http://link.aps.org/doi/10.1103/PhysRevLett.80.724>
- [44] H. Sugama and T. H. Watanabe, "Collisionless damping of zonal flows in helical systems," *Phys. Plasmas*, vol. 13, no. 1, pp. 1–18, jan 2006. <http://aip.scitation.org/doi/10.1063/1.2149311>

- [45] P. Angelino, X. Garbet, L. Villard, A. Bottino, S. Jolliet, P. Ghendrih, V. Grandgirard, B. F. McMillan, Y. Sarazin, G. Dif-Pradalier, and T. M. Tran, "The role of plasma elongation on the linear damping of zonal flows," *Phys. Plasmas*, vol. 15, no. 6, p. 062306, jun 2008. <http://aip.scitation.org/doi/10.1063/1.2928849>
- [46] Z. Gao, "Plasma shaping effects on the geodesic acoustic mode in the large orbit drift width limit," *Phys. Plasmas*, vol. 17, no. 9, p. 092503, sep 2010. <http://aip.scitation.org/doi/10.1063/1.3481464>
- [47] F. Hofmann, J. B. Lister, W. Anton, S. Barry, R. Behn, S. Bernel, G. Besson, F. Buhlmann, R. Chavan, M. Corboz, M. J. Dutch, B. P. Duval, D. Fasel, A. Favre, S. Franke, A. Heym, A. Hirt, C. Hollenstein, P. Isoz, B. Joye, X. Llobet, J. C. Magnin, B. Marletaz, P. Marmillod, Y. Martin, J. M. Mayor, J. M. Moret, C. Nieswand, P. J. Paris, A. Perez, Z. A. Pietrzyk, R. A. Pitts, A. Pochelon, R. Rage, O. Sauter, G. Tonetti, M. Q. Tran, F. Troyon, D. J. Ward, and H. Weisen, "Creation and control of variably shaped plasmas in TCV," *Plasma Phys. Control. Fusion*, vol. 36, no. 12 B, pp. B277–B287, dec 1994. <http://stacks.iop.org/0741-3335/36/i=12B/a=023?key=crossref.1b42d2ef9fb94c7eeb837ce0e2c29250>
- [48] R. Aymar, P. Barabaschi, and Y. Shimomura, "The ITER design," *Plasma Phys. Control. Fusion*, vol. 44, no. 5, pp. 519–565, may 2002. <http://stacks.iop.org/0741-3335/44/i=5/a=304?key=crossref.6afc8c86585ff6436716f752ac5cf8b1>
- [49] O. Sauter, S. Brunner, D. Kim, G. Merlo, R. Behn, Y. Camenen, S. Coda, B. P. Duval, L. Federspiel, T. P. Goodman, A. Karpushov, A. Merle, and T. Team, "On the non-stiffness of edge transport in L-mode tokamak plasmas," *Phys. Plasmas*, vol. 21, no. 5, p. 055906, 2014. <http://scitation.aip.org/content/aip/journal/pop/21/5/10.1063/1.4876612>
- [50] H. Sugama, "Gyrokinetic field theory," *Phys. Plasmas*, vol. 7, no. 2, pp. 466–480, feb 2000. <http://aip.scitation.org/doi/abs/10.1063/1.873832><http://aip.scitation.org/doi/10.1063/1.873832>
- [51] N. Tronko, A. Bottino, and E. Sonnendrücker, "Second order gyrokinetic theory for particle-in-cell codes," *Phys. Plasmas*, vol. 23, no. 8, p. 082505, aug 2016. <http://aip.scitation.org/doi/10.1063/1.4960039>
- [52] N. Tronko, A. Bottino, C. Chandre, and E. Sonnendrücker, "Hierarchy of second order gyrokinetic Hamiltonian models for particle-in-cell codes," *Plasma Phys. Control. Fusion*, vol. 59, no. 6, p. 064008, jun 2017. <http://stacks.iop.org/0741-3335/59/i=6/a=064008?key=crossref.0311bba6987f428be09915498fafb4f3>
- [53] J. R. Cary and A. J. Brizard, "Hamiltonian theory of guiding-center motion," *Rev. Mod. Phys.*, vol. 81, no. 2, pp. 693–738, may 2009. <https://link.aps.org/doi/10.1103/RevModPhys.81.693>



- [54] J. Dominski, B. F. McMillan, S. Brunner, G. Merlo, T. M. Tran, and L. Villard, "An arbitrary wavelength solver for global gyrokinetic simulations. Application to the study of fine radial structures on microturbulence due to non-adiabatic passing electron dynamics," *Phys. Plasmas*, vol. 24, no. 2, 2017. <http://dx.doi.org/10.1063/1.4976120><http://aip.scitation.org/toc/php/24/2>
- [55] E. Lanti, J. Dominski, S. Brunner, B. F. McMillan, and L. Villard, "Padé approximation of the adiabatic electron contribution to the gyrokinetic quasi-neutrality equation in the ORB5 code," *J. Phys. Conf. Ser.*, vol. 775, no. 1, p. 012006, nov 2016. <http://stacks.iop.org/1742-6596/775/i=1/a=012006?key=crossref.5970237ead440732086f3dc78b4903bc>
- [56] Y. Idomura, "A new hybrid kinetic electron model for full-f gyrokinetic simulations," *J. Comput. Phys.*, vol. 313, pp. 511–531, 2016. <http://dx.doi.org/10.1016/j.jcp.2016.02.057>
- [57] A. Y. Aydemir, "A unified Monte Carlo interpretation of particle simulations and applications to non-neutral plasmas," *Phys. Plasmas*, vol. 1, no. 4, p. 822, 1994. <http://scitation.aip.org/content/aip/journal/pop/1/4/10.1063/1.870740>
- [58] R. Hatzky, R. Kleiber, A. Könies, A. Mishchenko, M. Borchardt, A. Bottino, and E. Sonnendrücker, "Reduction of the statistical error in electromagnetic gyrokinetic particle-in-cell simulations," *J. Plasma Phys.*, vol. 85, no. 1, p. 905850112, feb 2019. [https://www.cambridge.org/core/product/identifier/S0022377819000096/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S0022377819000096/type/journal_article)
- [59] Z. Lin, T. S. Hahm, W. W. Lee, W. M. Tang, and R. B. White, "Turbulent transport reduction by zonal flows: Massively parallel simulations," *Science (80-. )*, vol. 281, no. 5384, pp. 1835–1837, sep 1998. <http://www.ncbi.nlm.nih.gov/pubmed/9743492>
- [60] P. Angelino, A. Bottino, R. Hatzky, S. Jolliet, O. Sauter, T. M. Tran, and L. Villard, "On the definition of a kinetic equilibrium in global gyrokinetic simulations," *Phys. Plasmas*, vol. 13, no. 5, p. 052304, may 2006. <http://aip.scitation.org/doi/10.1063/1.2193947><http://scitation.aip.org/content/aip/journal/pop/13/5/10.1063/1.2193947>
- [61] T. S. Hahm, "Nonlinear gyrokinetic equations for turbulence in core transport barriers," *Phys. Plasmas*, vol. 3, no. 12, pp. 4658–4664, jun 1996. <https://aip.scitation.org/doi/10.1063/1.872034><http://aip.scitation.org/doi/10.1063/1.872034>
- [62] J. D. Collier, B. F. McMillan, and J. R. Robinson, "Global gyrokinetic simulations with strong flows," *Phys. Plasmas*, vol. 23, no. 11, p. 112503, nov 2016. <http://aip.scitation.org/doi/10.1063/1.4965900>
- [63] L. Guazzotto, R. Betti, J. Manickam, and S. Kaye, "Numerical study of tokamak equilibria with arbitrary flow," *Phys. Plasmas*, vol. 11, no. 2, pp. 604–614, feb 2004. <http://aip.scitation.org/doi/10.1063/1.1637918>

## Bibliography

---

- [64] G. Dif-Pradalier, V. Grandgirard, Y. Sarazin, X. Garbet, and P. Ghendrih, “Interplay between gyrokinetic turbulence, flows, and collisions: Perspectives on transport and poloidal rotation,” *Phys. Rev. Lett.*, vol. 103, no. 6, p. 065002, aug 2009. <https://link.aps.org/doi/10.1103/PhysRevLett.103.065002>
- [65] T. Vernay, S. Brunner, L. Villard, B. F. McMillan, S. Jolliet, T. M. Tran, and A. Bottino, “Synergy between ion temperature gradient turbulence and neoclassical processes in global gyrokinetic particle-in-cell simulations,” *Phys. Plasmas*, vol. 19, no. 4, p. 042301, 2012. <http://scitation.aip.org/content/aip/journal/pop/19/4/10.1063/1.3699189>
- [66] T. Vernay, S. Brunner, L. Villard, B. F. McMillan, S. Jolliet, T. M. Tran, A. Bottino, and J. P. Graves, “Neoclassical equilibria as starting point for global gyrokinetic microturbulence simulations,” *Phys. Plasmas*, vol. 17, no. 12, p. 122301, dec 2010. <http://pop.aip.org/resource/1/phpaen/v17/i12/p122301>  
<http://aip.scitation.org/doi/10.1063/1.3519513>
- [67] Z. Lin, W. M. Tang, and W. W. Lee, “Gyrokinetic particle simulation of neoclassical transport,” *Phys. Plasmas*, vol. 2, no. 8, pp. 2975–2988, aug 1995. <http://aip.scitation.org/doi/10.1063/1.871196>
- [68] S. Brunner, E. Valeo, and J. A. Krommes, “Collisional delta-f scheme with evolving background for transport time scale simulations,” *Phys. Plasmas*, vol. 6, no. 12, pp. 4504–4521, 1999. <http://pop.aip.org/resource/1/phpaen/v6/i12/p4504>  
<http://scitation.aip.org/getpdf/servlet/GetPDFServlet?filetype=pdf&id=PHPAEN000006000012004504000001&idtype=cvips&doi=10.1063/1.873738&prog=normal>
- [69] S. Parker and W. W. Lee, “A fully nonlinear characteristic method for gyrokinetic simulation,” *Phys. Fluids B*, vol. 5, no. 1, pp. 77–86, jan 1993. <http://aip.scitation.org/doi/10.1063/1.860870>
- [70] G. Hu and J. A. Krommes, “Generalized weighting scheme for  $\delta f$  particle simulation method,” *Phys. Plasmas*, vol. 1, no. 4, pp. 863–874, apr 1994. <http://aip.scitation.org/doi/10.1063/1.870745>
- [71] S. J. Allfrey and R. Hatzky, “A revised  $\delta f$  algorithm for nonlinear PIC simulation,” *Comput. Phys. Commun.*, vol. 154, no. 2, pp. 98–104, aug 2003. <https://www.sciencedirect.com/science/article/pii/S0010465503002881>  
<https://www.sciencedirect.com/science/article/pii/S0010465503002881?via=ihub>
- [72] J. H. Halton, “On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals,” *Numer. Math.*, vol. 2, no. 1, pp. 84–90, dec 1960. <https://link.springer.com/10.1007/BF01386213>

- [73] J. M. Hammersley, "Monte carlo methods for solving multivariable problems," *Ann. N. Y. Acad. Sci.*, vol. 86, no. 3, pp. 844–874, dec 1960. <http://doi.wiley.com/10.1111/j.1749-6632.1960.tb42846.x>
- [74] J. G. van der Corput, "Verteilungsfunktionen. I. Mitt." *Proc. Akad. Wet. Amsterdam*, vol. 38, pp. 813–821, 1935.
- [75] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," in *Proc. IEEE*, vol. 93, no. 2, feb 2005, pp. 216–231. <http://ieeexplore.ieee.org/document/1386650/>
- [76] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, jan 1999. <http://epubs.siam.org/doi/book/10.1137/1.9780898719604>
- [77] T. M. Tran, K. Appert, M. Fivaz, G. Jost, J. Vaclavik, and L. Villard, "Global gyrokinetic simulation of ion-temperature-gradient-driven instabilities using particles," in *Theory of fusion plasmas*. Societa Italiana di Fisica, 1999, p. 45.
- [78] R. Hatzky, T. M. Tran, A. Könies, R. Kleiber, and S. J. Allfrey, "Energy conservation in a nonlinear gyrokinetic particle-in-cell code for ion-temperature-gradient-driven modes in  $\theta$ -pinch geometry," *Phys. Plasmas*, vol. 9, no. 3, p. 898, mar 2002. <http://aip.scitation.org/doi/10.1063/1.1449889>
- [79] S. Jolliet, A. Bottino, P. Angelino, R. Hatzky, T. M. Tran, B. F. McMillan, O. Sauter, K. Appert, Y. Idomura, and L. Villard, "A global collisionless PIC code in magnetic coordinates," *Comput. Phys. Commun.*, vol. 177, no. 5, pp. 409–425, sep 2007. <http://linkinghub.elsevier.com/retrieve/pii/S0010465507002251>  
<http://www.sciencedirect.com/science/article/pii/S0010465507002251>
- [80] B. F. McMillan, S. Jolliet, T. M. Tran, L. Villard, A. Bottino, and P. Angelino, "Long global gyrokinetic simulations: Source terms and particle noise control," *Phys. Plasmas*, vol. 15, no. 5, pp. 1–10, may 2008. <http://aip.scitation.org/doi/10.1063/1.2921792>
- [81] Y. Chen and S. Parker, "Coarse-graining phase space in  $\delta f$  particle-in-cell simulations," *Phys. Plasmas*, vol. 14, no. 8, p. 082301, aug 2007. <http://aip.scitation.org/doi/10.1063/1.2751603>
- [82] E. Sonnendrücker, A. Wachter, R. Hatzky, and R. Kleiber, "A split control variate scheme for PIC simulations with collisions," *J. Comput. Phys.*, vol. 295, pp. 402–419, aug 2015. <https://www.sciencedirect.com/science/article/pii/S0021999115002442?via=ihub3Dihub>

- [83] Y. Chen and S. Parker, "Gyrokinetic turbulence simulations with kinetic electrons," *Phys. Plasmas*, vol. 8, no. 5, pp. 2095–2100, may 2001. <http://aip.scitation.org/doi/10.1063/1.1351828>
- [84] Y. Chen and S. Parker, "A  $\delta f$  particle method for gyrokinetic simulations with kinetic electrons and electromagnetic perturbations," *J. Comput. Phys.*, vol. 189, no. 2, pp. 463–475, aug 2003. <https://www.sciencedirect.com/science/article/pii/S0021999103002286>
- [85] A. Mishchenko, R. Hatzky, and A. Könies, "Conventional  $\delta f$ -particle simulations of electromagnetic perturbations with finite elements," *Phys. Plasmas*, vol. 11, no. 12, pp. 5480–5486, dec 2004. <http://aip.scitation.org/doi/10.1063/1.1812275>
- [86] A. Mishchenko, A. Könies, and R. Hatzky, "Gyrokinetic simulations with a particle discretization of the field equations," in *Proc. Jt. Varenna-Lausanne Int. Work.* Bologna: Societa Italiana di Fisica, 2004, p. 315.
- [87] R. Hatzky, A. Könies, and A. Mishchenko, "Electromagnetic gyrokinetic PIC simulation with an adjustable control variates method," *J. Comput. Phys.*, vol. 225, no. 1, pp. 568–590, jul 2007. <https://www.sciencedirect.com/science/article/pii/S0021999106006085>
- [88] A. Mishchenko, A. Bottino, R. Hatzky, E. Sonnendrücker, R. Kleiber, and A. Könies, "Mitigation of the cancellation problem in the gyrokinetic particle-in-cell simulations of global electromagnetic modes," *Phys. Plasmas*, vol. 24, no. 8, p. 081206, aug 2017. <http://aip.scitation.org/doi/10.1063/1.4997540>
- [89] A. Mishchenko, A. Könies, R. Kleiber, and M. D. J. Cole, "Pullback transformation in gyrokinetic electromagnetic simulations," *Phys. Plasmas*, vol. 21, no. 9, p. 092110, sep 2014. <http://aip.scitation.org/doi/10.1063/1.4895501>
- [90] R. Kleiber, R. Hatzky, A. Könies, A. Mishchenko, and E. Sonnendrücker, "An explicit large time step particle-in-cell scheme for nonlinear gyrokinetic simulations in the electromagnetic regime," *Phys. Plasmas*, vol. 23, no. 3, p. 032501, mar 2016. <http://aip.scitation.org/doi/10.1063/1.4942788>
- [91] A. Mishchenko, M. D. J. Cole, R. Kleiber, and A. Könies, "New variables for gyrokinetic electromagnetic simulations," *Phys. Plasmas*, vol. 21, no. 5, p. 052113, may 2014. <http://aip.scitation.org/doi/10.1063/1.4880560>
- [92] A. Mishchenko, A. Bottino, A. Biancalani, R. Hatzky, T. Hayward-Schneider, N. Ohana, E. Lanti, S. Brunner, L. Villard, M. Borchardt, R. Kleiber, A. Könies, and A. Koenies, "Pullback scheme implementation in ORB5," *Comput. Phys. Commun.*, dec 2018. [https://www.sciencedirect.com/science/article/pii/S0010465518304181?dgcid=rss\\_sd\\_all](https://www.sciencedirect.com/science/article/pii/S0010465518304181?dgcid=rss_sd_all)<http://arxiv.org/abs/1811.05346><https://linkinghub.elsevier.com/retrieve/pii/S0010465518304181>

- 
- [93] A. Biancalani, A. Bottino, S. Briguglio, A. Könies, P. Lauber, A. Mishchenko, E. Poli, B. D. Scott, and F. Zonca, “Linear gyrokinetic particle-in-cell simulations of Alfvén instabilities in tokamaks,” *Phys. Plasmas*, vol. 23, no. 1, p. 012108, jan 2016. <http://aip.scitation.org/doi/10.1063/1.4939803>
- [94] A. Biancalani, A. Bottino, M. D. J. Cole, C. Di Troia, P. Lauber, A. Mishchenko, B. Scott, and F. Zonca, “Nonlinear interplay of Alfvén instabilities and energetic particles in tokamaks,” *Plasma Phys. Control. Fusion*, vol. 59, no. 5, p. 054004, may 2017. <http://stacks.iop.org/0741-3335/59/i=5/a=054004?key=crossref.5535f60889456ea6eafc60d759965eb5>
- [95] P. J. Catto, W. M. Tang, and D. E. Baldwin, “Generalized gyrokinetics,” *Plasma Phys.*, vol. 23, no. 7, pp. 639–650, jul 1981. <http://stacks.iop.org/0032-1028/23/i=7/a=005?key=crossref.ec4e34417bc78b7c7d606d8fa458b827>
- [96] A. J. Brizard and T. S. Hahm, “Foundations of nonlinear gyrokinetic theory,” *Rev. Mod. Phys.*, vol. 79, no. 2, pp. 421–468, apr 2007. <http://link.aps.org/doi/10.1103/RevModPhys.79.421>  
<http://journals.aps.org/rmp/abstract/10.1103/RevModPhys.79.421>
- [97] B. F. McMillan and L. Villard, “Accuracy of momentum and gyrodensity transport in global gyrokinetic particle-in-cell simulations,” *Phys. Plasmas*, vol. 21, no. 5, p. 052501, may 2014. <http://aip.scitation.org/doi/10.1063/1.4873387>
- [98] Y. Sarazin, V. Grandgirard, J. Abiteboul, S. J. Allfrey, X. Garbet, P. Ghendrih, G. Latu, A. Strugarek, G. Dif-Pradalier, P. H. Diamond, S. Ku, C. S. Chang, B. F. McMillan, T. M. Tran, L. Villard, S. Joliet, A. Bottino, and P. Angelino, “Predictions on heat transport and plasma rotation from global gyrokinetic simulations,” *Nucl. Fusion*, vol. 51, no. 10, p. 103023, oct 2011. <http://stacks.iop.org/0029-5515/51/i=10/a=103023?key=crossref.fa51644f0c67bf3302c4798ff98f6af2>
- [99] C. Kim and S. Parker, “Massively Parallel Three-Dimensional Toroidal Gyrokinetic Flux-Tube Turbulence Simulation,” *J. Comput. Phys.*, vol. 161, no. 2, pp. 589–604, jul 2000. <https://www.sciencedirect.com/science/article/pii/S0021999100965185>
- [100] R. Hatzky, “Domain cloning for a particle-in-cell (PIC) code on a cluster of symmetric-multiprocessor (SMP) computers,” *Parallel Comput.*, vol. 32, no. 4, pp. 325–330, apr 2006. <https://www.sciencedirect.com/science/article/pii/S0167819106000111>
- [101] N. Ohana, C. Gheller, S. Brunner, E. Lanti, and L. Villard, “Accelerated Gyrokinetic Simulations with ORB5,” no. Submitted to CPC, 2019.
- [102] G. Brendan, *Systems Performance: Enterprise and the Cloud*. Prentice Hall, 2013.
- [103] F. Hariri, T. M. Tran, A. Jocksch, E. Lanti, J. Progsch, P. Messmer, S. Brunner, C. Gheller, and L. Villard, “A portable platform for accelerated PIC codes and

- its application to GPUs using OpenACC,” *Comput. Phys. Commun.*, may 2016. <http://linkinghub.elsevier.com/retrieve/pii/S0010465516301242>
- [104] V. K. Decyk and T. V. Singh, “Adaptable Particle-in-Cell algorithms for graphical processing units,” *Comput. Phys. Commun.*, vol. 182, no. 3, pp. 641–648, mar 2011. <http://www.sciencedirect.com/science/article/pii/S0010465510004558>
  - [105] X. Kong, M. C. Huang, C. Ren, and V. K. Decyk, “Particle-in-cell simulations with charge-conserving current deposition on graphic processing units,” *J. Comput. Phys.*, vol. 230, no. 4, pp. 1676–1685, feb 2011. <http://www.sciencedirect.com/science/article/pii/S0021999110006479>
  - [106] Intel, “Intel Xeon Phi Product Family,” 2016. <http://www.intel.eu/content/www/eu/en/processors/xeon/xeon-phi-detail.html>
  - [107] R. Teyssier, “Cosmological hydrodynamics with adaptive mesh refinement,” *Astron. Astrophys.*, vol. 385, no. 1, pp. 337–364, apr 2002. <http://dx.doi.org/10.1051/0004-6361:20011817>
  - [108] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer simulations*. CRC press, 2004.
  - [109] N. Ohana, A. Jocksch, E. Lanti, T. M. Tran, S. Brunner, C. Gheller, F. Hariri, and L. Villard, “Towards the optimization of a gyrokinetic Particle-In-Cell (PIC) code on large-scale hybrid architectures,” *J. Phys. Conf. Ser.*, vol. 775, no. 1, p. 012010, nov 2016. <http://stacks.iop.org/1742-6596/775/i=1/a=012010?key=crossref.80edc1eb5f240b55c155184ef1ee63f0>
  - [110] N. Ohana, A. Jocksch, E. Lanti, A. Scheinberg, S. Brunner, C. Gheller, and L. Villard, “The Particle-In-Fourier (PIF) Approach Applied to Gyrokinetic Simulations,” in *PASC17*, 2017. <https://infoscience.epfl.ch/record/231824?ln=en>
  - [111] N. Ohana, A. Jocksch, E. Lanti, A. Scheinberg, S. Brunner, C. Gheller, and L. Villard, “Porting a Legacy Global Lagrangian PIC Code on Many-Core and GPU-Accelerated Architectures,” in *PASC18*, jul 2018. <https://infoscience.epfl.ch/record/255881?ln=en>
  - [112] M. Boesl, A. Bergmann, A. Bottino, D. Coster, F. Jenko, E. Lanti, and N. Ohana, “Gyrokinetic full-f PIC simulations of an ELM heat pulse in the scrape-off layer,” *J. Plasma Phys.*, no. Submitted, 2019.
  - [113] A. Jocksch, F. Hariri, T. M. Tran, S. Brunner, C. Gheller, and L. Villard, “A Bucket Sort Algorithm for the Particle-In-Cell Method on Manycore Architectures,” in *Parallel Process. Appl. Math. 11th Int. Conf. PPAM 2015, Krakow, Poland, Sept. 6-9, 2015. Revis. Sel. Pap. Part I*, R. Wyrzykowski, E. Deelman, J. Dongarra, K. Karczewski,

- J. Kitowski, and K. Wiatr, Eds. Cham: Springer International Publishing, 2016, pp. 43–52. [http://dx.doi.org/10.1007/978-3-319-32149-3\\_{ }5](http://dx.doi.org/10.1007/978-3-319-32149-3_{ }5)
- [114] D. Hilbert, “Über die stetige Abbildung einer Linie auf ein Flächenstück,” *Math. Ann.*, vol. 38, pp. 459–460, 1891.
- [115] I. Surmin, S. Bastrakov, E. Efimenko, A. Gonoskov, A. Korzhimanov, and I. Meyerov, “Particle-in-Cell laser-plasma simulation on Xeon Phi coprocessors,” *Comput. Phys. Commun.*, vol. 202, pp. 204–210, may 2016. <http://www.sciencedirect.com/science/article/pii/S0010465516300194>
- [116] W. W. Lee, “Gyrokinetic particle simulation model,” *J. Comput. Phys.*, vol. 72, no. 1, pp. 243–269, sep 1987. <http://www.sciencedirect.com/science/article/pii/0021999187900805><https://aip.scitation.org/doi/10.1063/1.864140>
- [117] S. Parker, W. W. Lee, and R. A. Santoro, “Gyrokinetic simulation of ion temperature gradient driven turbulence in 3D toroidal geometry,” *Phys. Rev. Lett.*, vol. 71, no. 13, pp. 2042–2045, sep 1993. <https://link.aps.org/doi/10.1103/PhysRevLett.71.2042>
- [118] G. Rewoldt, W. M. Tang, and R. J. Hastie, “Collisional effects on kinetic electromagnetic modes and associated quasilinear transport,” *Phys. Fluids*, vol. 30, no. 3, p. 807, sep 1987. <https://aip.scitation.org/doi/10.1063/1.866332>
- [119] H. Doerk, F. Jenko, M. J. Pueschel, and D. R. Hatch, “Gyrokinetic microtearing turbulence,” *Phys. Rev. Lett.*, vol. 106, no. 15, p. 155003, apr 2011. <https://link.aps.org/doi/10.1103/PhysRevLett.106.155003>
- [120] J. Y. Kim, W. Horton, and J. Q. Dong, “Electromagnetic effect on the toroidal ion temperature gradient mode,” *Phys. Fluids B Plasma Phys.*, vol. 5, no. 11, pp. 4030–4039, nov 1993. <http://aip.scitation.org/doi/10.1063/1.860623>
- [121] H. Naitou, K. Tsuda, W. W. Lee, and R. D. Sydora, “Gyrokinetic simulation of internal kink modes,” *Phys. Plasmas*, vol. 2, no. 11, pp. 4257–4268, nov 1995. <http://aip.scitation.org/doi/10.1063/1.871051>
- [122] W. W. Lee, J. L. V. Lewandowski, T. S. Hahm, and Z. Lin, “Shear-Alfvén waves in gyrokinetic plasmas,” *Phys. Plasmas*, vol. 8, no. 10, pp. 4435–4440, 2001. <http://scitation.aip.org/content/aip/journal/pop/8/10/10.1063/1.1400124>
- [123] Y. Idomura, S. Tokuda, and Y. Kishimoto, *Gyrokinetic simulations of tokamak micro-turbulence including kinetic electron effects*. Japan: Japan Society of Plasma Science and Nuclear Fusion Research, 2004. [http://inis.iaea.org/search/search.aspx?orig\\_{ }q=RN:36083997](http://inis.iaea.org/search/search.aspx?orig_{ }q=RN:36083997)

- [124] Z. Lin, Y. Nishimura, Y. Xiao, I. Holod, W. L. Zhang, and L. Chen, “Global gyrokinetic particle simulations with kinetic electrons,” in *Plasma Phys. Control. Fusion*, vol. 49, no. 12 B. IOP Publishing, dec 2007, pp. B163–B172. <http://stacks.iop.org/0741-3335/49/i=12B/a=S15?key=crossref.e65577f33cfb79b6b2364377c3094376>
- [125] Z. Huang, S. Coda, G. Merlo, S. Brunner, L. Villard, B. Labit, and C. Theiler, “Experimental observations of modes with geodesic acoustic character from the core to the edge in the TCV tokamak,” *Plasma Phys. Control. Fusion*, vol. 60, no. 3, p. 034007, mar 2018. <http://stacks.iop.org/0741-3335/60/i=3/a=034007?key=crossref.6367425e91b5d228e8b8733fe59998f4>
- [126] E. Fable, O. Sauter, S. Coda, T. P. Goodman, M. A. Henderson, H. Weisen, A. Zabolotsky, and C. Zucca, “Inward thermodiffusive particle pinch in electron internal transport barriers in TCV,” *Plasma Phys. Control. Fusion*, vol. 48, no. 9, pp. 1271–1283, 2006.
- [127] E. Fable, C. Angioni, and O. Sauter, “Gyrokinetic calculations of steady-state particle transport in electron internal transport barriers,” *Plasma Phys. Control. Fusion*, vol. 50, no. 11, 2008.
- [128] E. Fable, C. Angioni, and O. Sauter, “The role of ion and electron electrostatic turbulence in characterizing stationary particle transport in the core of tokamak plasmas,” *Plasma Phys. Control. Fusion*, vol. 52, no. 1, 2010.
- [129] A. Mariani, S. Brunner, J. Dominski, A. Merle, G. Merlo, O. Sauter, T. Görler, F. Jenko, and D. Told, “Identifying microturbulence regimes in a TCV discharge making use of physical constraints on particle and heat fluxes,” *Phys. Plasmas*, vol. 25, no. 1, 2018.
- [130] B. F. McMillan, S. Jolliet, T. M. Tran, L. Villard, A. Bottino, and P. Angelino, “Avalanchelike bursts in global gyrokinetic simulations,” *Phys. Plasmas*, vol. 16, no. 2, p. 022310, feb 2009. <http://aip.scitation.org/doi/10.1063/1.3079076>
- [131] L. Villard, B. F. McMillan, O. Sauter, F. Hariri, J. Dominski, G. Merlo, S. Brunner, and T. M. Tran, “Turbulence and zonal flow structures in the core and L-mode pedestal of tokamak plasmas,” in *J. Phys. Conf. Ser.*, vol. 561. IOP Publishing, nov 2014, p. 012022. <http://stacks.iop.org/1742-6596/561/i=1/a=012022?key=crossref.2297d24f1f3d65f251abc02591ea3552>
- [132] L. Villard, B. F. McMillan, E. Lanti, N. Ohana, A. Bottino, A. Biancalani, I. Novikau, S. Brunner, O. Sauter, N. Tronko, and A. Mishchenko, “Global turbulence features across marginality and non-local pedestal-core interactions,” *Plasma Phys. Control. Fusion*, vol. 61, no. 3, p. 034003, mar 2019. <http://stacks.iop.org/0741-3335/61/i=3/a=034003?key=crossref.c8ca4645c7d50553f6e81d07107cdd71>



- 
- [133] R. Waltz, J. Candy, F. L. Hinton, C. Estrada-Mila, and J. E. Kinsey, "Advances in comprehensive gyrokinetic simulations of transport in tokamaks," *Nucl. Fusion*, vol. 45, no. 7, pp. 741–750, jul 2005. <http://stacks.iop.org/0029-5515/45/i=7/a=023?key=crossref.4e372590ad2532cf9c2148b9ce996d76>
  - [134] J. Candy, "Beta scaling of transport in microturbulence simulations," *Phys. Plasmas*, vol. 12, no. 7, p. 072307, jul 2005. <http://aip.scitation.org/doi/10.1063/1.1954123>
  - [135] R. E. Waltz, M. E. Austin, K. H. Burrell, and J. Candy, "Gyrokinetic simulations of off-axis minimum- q profile corrugations," *Phys. Plasmas*, vol. 13, no. 5, p. 052301, may 2006. <http://aip.scitation.org/doi/10.1063/1.2195418>
  - [136] J. Chowdhury, R. Ganesh, P. Angelino, J. Vaclavik, L. Villard, and S. Brunner, "Role of nonadiabatic untrapped electrons in global electrostatic ion temperature gradient driven modes in a tokamak," *Phys. Plasmas*, vol. 15, no. 7, p. 072117, jul 2008. <http://aip.scitation.org/doi/10.1063/1.2957917>
  - [137] J. Dominski, S. Brunner, T. Görler, F. Jenko, D. Told, and L. Villard, "How non-adiabatic passing electron layers of linear microinstabilities affect turbulent transport," *Phys. Plasmas*, vol. 22, no. 6, p. 062303, jun 2015. <http://aip.scitation.org/doi/10.1063/1.4922659>  
<http://scitation.aip.org/content/aip/journal/pop/22/6/10.1063/1.4922659>
  - [138] Y. Asahi, V. Grandgirard, Y. Idomura, X. Garbet, G. Latu, Y. Sarazin, G. Dif-Pradalier, P. Donnel, and C. Ehrlacher, "Benchmarking of flux-driven full-F gyrokinetic simulations," *Phys. Plasmas*, vol. 24, no. 10, p. 102515, oct 2017. <http://aip.scitation.org/doi/10.1063/1.4998015>
  - [139] T. Görler, X. Lapillonne, S. Brunner, T. Dannert, F. Jenko, F. Merz, and D. Told, "The global version of the gyrokinetic turbulence code GENE," *J. Comput. Phys.*, vol. 230, no. 18, pp. 7053–7071, aug 2011. <http://linkinghub.elsevier.com/retrieve/pii/S0021999111003457>
  - [140] A. Pochelon, P. Angelino, R. Behn, S. Brunner, S. Coda, N. Kirneva, S. Y. Medvedev, H. Reimerdes, J. Rossel, O. Sauter, L. Villard, D. WÄGNER, A. Bottino, Y. Camenen, G. P. CANAL, P. K. CHATTOPADHYAY, B. P. Duval, A. FASOLI, T. P. Goodman, S. Jolliet, A. Karpushov, B. Labit, A. Marinoni, J. M. Moret, A. PITZSCHKE, L. Porte, M. RANCIC, V. S. UDINTSEV, and the TCV Team, "Recent TCV Results - Innovative Plasma Shaping to Improve Plasma Properties and Insight," *Plasma Fusion Res.*, vol. 7, pp. 2 502 148–2 502 148, 2012. <http://jlc.jst.go.jp/DN/JST.JSTAGE/pfr/7.2502148?lang=en&from=CrossRef&type=abstract>
  - [141] M. Fontana, "Turbulence studies in TCV using the Correlation ECE diagnostic," 2018. <https://infoscience.epfl.ch/record/258091?ln=fr>

## Bibliography

---

- [142] B. F. McMillan, X. Lapillonne, S. Brunner, L. Villard, S. Jolliet, A. Bottino, T. Görler, and F. Jenko, “System size effects on gyrokinetic turbulence,” *Phys. Rev. Lett.*, vol. 105, no. 15, p. 155001, oct 2010. <http://link.aps.org/doi/10.1103/PhysRevLett.105.155001>
- [143] “OpenMP Specifications,” p. 368, 2015. <http://openmp.org/wp/openmp-specifications/>

# Emmanuel Lanti | Curriculum Vitae

✉ emmanuel.lanti@alumni.epfl.ch

## Education

### Swiss Plasma Center - EPFL

*Ph.D. at the Swiss Plasma Center*

"Global Flux-Driven Simulations of Ion Temperature-Gradient and Trapped-Electron Modes Driven Turbulence With an Improved Multithreaded Gyrokinetic PIC Code"

Lausanne

2015–2019

### École Polytechnique Fédérale de Lausanne (EPFL)

*Masters of Science in Physics with a minor in Computational Science and Engineering*

Title of the MSc thesis: "Shaping effects on tokamak scrape-off layer turbulence"

Lausanne

2012–2014

### École Polytechnique Fédérale de Lausanne (EPFL)

*Bachelor of Science in Physics*

Lausanne

2008–2012

### Gymnase de Nyon

*Maturité gymnasiale*

Orientation: "physique et application des mathématiques"

Nyon

2005–2008

## Publications and conference participation

### As first author

- E. Lanti, N. Ohana, N. Tronko, *et al.*, "Orb5: a global electromagnetic gyrokinetic code using the pic approach in toroidal geometry", arXiv, 1905.01906 (2019)
- E. Lanti, B. F. McMillan, S. Brunner, *et al.*, "Gradient-and flux-driven global gyrokinetic simulations of ITG and TEM turbulence with an improved hybrid kinetic electron model", Journal of Physics: Conference Series **1125**, 1 (2018)
- E. Lanti, J. Dominski, S. Brunner, *et al.*, "Padé approximation of the adiabatic electron contribution to the gyrokinetic quasi-neutrality equation in the ORB5 code", Journal of Physics: Conference Series **775**, 1 (2016)

### As co-author

- A. Mishchenko, A. Bottino, A. Biancalani, *et al.*, "Pullback scheme implementation in ORB5", Computer Physics Communications **238**, (2019)
- L. Villard, B. F. McMillan, E. Lanti, *et al.*, "Global turbulence features across marginality and non-local pedestal-core interactions", Plasma Physics and Controlled Fusion **61**, 3 (2019)
- I. Novikau, A. Biancalani, A. Bottino, *et al.*, "Implementation of energy transfer technique in ORB5 to study collisionless wave-particle interactions in phase-space", arXiv, 1903.05024 (2019)
- N. Tronko, A. Bottino, C. Chandre, *et al.*, "First principles gyrokinetic analysis of electromagnetic plasma instabilities", arXiv, 1903.10812 (2019)
- G. Merlo, J. Dominski, A. Bhattacharjee, *et al.*, "Cross-verification of the global gyrokinetic codes GENE and XGC", Physics of Plasmas **25**, 6 (2018)
- N. Ohana, L. Villard, E. Lanti, *et al.*, "Mode excitation by an antenna in global gyrokinetic simulations", Journal of Physics: Conference Series **1125**, 1 (2018)
- F. Riva, E. Lanti, S. Jolliet, *et al.*, "Plasma shaping effects on tokamak scrape-off layer turbulence", Plasma Physics and Controlled Fusion **59**, 3 (2017)
- A. Jocksch, N. Ohana, E. Lanti, *et al.*, "Prediction of the Inter-Node Communication Costs of a New Gyrokinetic Code with Toroidal Domain", Parallel Processing and Applied Mathematics (Springer International Publishing), 370–380 (2018)
- F. Hariri, T. M. Tran, A. Jocksch, *et al.*, "A portable platform for accelerated PIC codes and its application to GPUs using OpenACC", Computer Physics Communications **207**, (2016)
- N. Ohana, A. Jocksch, E. Lanti, *et al.*, "Towards the optimization of a gyrokinetic Particle-In-Cell (PIC) code on large-scale hybrid architectures", Journal of Physics: Conference Series **775**, 1 (2016)
- M. Boesl, A. Bergmann, A. Bottino, *et al.*, "Gyrokinetic full-f PIC simulations of an ELM heat pulse in the scrape-off layer", J. Plasma Phys., (In preparation)

## Conferences

---

- E. Lanti, B. F. McMillan, S. Brunner, *et al.*, “Gradient-and flux-driven global gyrokinetic simulations of ITG and TEM turbulence with an improved hybrid kinetic electron model”, Joint Varenna-Lausanne International Workshop, Varenna, Italy (2018)
- E. Lanti, S. Brunner, N. Ohana, *et al.*, “An improved hybrid electron model for global gyrokinetic simulations using the ORB5 PIC code”, EFTC, Athens, Greece (2017)
- E. Lanti, A. Scheinberg, A. Jocksch, *et al.*, “Hybrid OpenMP/MPI parallelization of the charge deposition step in the global gyrokinetic Particle-In-Cell code ORB5”, PSAC17, Lugano, Switzerland (2017)
- E. Lanti, J. Dominski, S. Brunner, *et al.*, “Padé approximation of the adiabatic electron contribution to the gyrokinetic quasi-neutrality equation in the ORB5 code”, Joint Varenna-Lausanne International Workshop, Varenna, Italy (2016)
- E. Lanti, T. M. Tran, A. Jocksch, *et al.*, “A Portable Platform for Accelerated PIC Codes and Its Application to Multi- and Many Integrated Core Architectures Using Hybrid MPI/OpenMP”, PASC16, Lausanne, Switzerland (2016)
- E. Lanti, T. M. Tran, A. Jocksch, *et al.*, “Towards porting a gyrokinetic PIC code on coprocessor-equipped supercomputers”, Joint Annual Meeting of the Austrian Physical Society and the Swiss Physical Society, Vienna, Austria (2015)

## Professional experience

---

### Teaching assistant

EPFL

2015–present

General Physics II with Prof. Ricci.

Numerical Physics I & II with Prof. Villard

General Physics I with Dr. Hogge

General Physics I with Dr. Blanc

### Teaching assistant

EPFL

during MSc

General Physics with Prof. Forrò. General physics with Prof. Ricci.

General physics with Prof. Dietler (corrections only)

### Volunteer firefighter

SDIS Nyon-Dôle

Nyon

2014–present

## Computer skills

---

**Programming:** Python, Fortran, Matlab, Bash, HDF5, OpenMP, MPI, OpenACC, C++

**Miscellaneous:** HPC, Linux,  $\LaTeX$

## Languages

---

◦ **French**, mothertongue

◦ **Italian**, intermediate

◦ **English**, intermediate

◦ **German**, basic



