Technische Universität München

Max-Planck-Institut für Plasmaphysik

PICLS: a gyrokinetic full-f particle-in-cell code for the scrape-off layer

Mathias Helmut Bösl

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

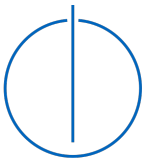
Vorsitzender:	Prof. Dr. Hans-Joachim Bungartz
Prüfer der Dissertation:	1. Prof. Dr. Frank Jenko
	2. Prof. Laurent Villard

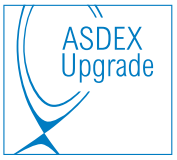
Die Dissertation wurde am 02.12.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 19.04.2021 angenommen.

DEPARTMENT OF INFORMATICS
TECHNISCHE UNIVERSITÄT MÜNCHEN

PICLS: a gyrokinetic full-f particle-in-cell code for the scrape-off layer

Author:	Mathias Helmut Bösl
Supervisor:	Prof. Dr. Frank Jenko
Advisors:	Dr. Andreas Bergmann
	Dr. Alberto Bottino
	Dr. David Coster
Submission Date:	30.11.2020





HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

EUROfusion

Abstract

On the way towards fusion energy devices numerical plasma simulations play a key role. For both, fusion research as well as risk mitigation in the planning process of fusion devices, numerical models are an important building block. Especially, five-dimensional gyrokinetic turbulence simulations are believed to deliver a good qualitative understanding of the plasma behavior and precise quantitative data. In the plasma core these models are well established for turbulence and transport studies. However, in the plasma edge, close to the device boundary, numerical models are still only at the beginning of covering the rich physics. Since this region is of significant importance for achievable fusion power and lifetime of plasma-facing components, covering the present additional challenges is vital. In particular, steep density and temperature gradients and large fluctuation amplitudes are dominant in this region and call for a "full-f" treatment. Since the current gyrokinetic core codes are not yet capable to tackle these complications, the development of new gyrokinetic models for the plasma edge is necessary. Some development efforts already started, but several research topics are still untouched.

Therefore, in this thesis we present the newly developed code PICLS, which is aimed to specifically study turbulence in open magnetic field line systems, as present in the Tokamak scrape-off layer (SOL) region. PICLS is a gyrokinetic particle-in-cell code and is based on an electrostatic full-f model with linearized field equation and kinetic electrons. For the discretization, B-spline Finite elements are applied. The plasma sheath that builds up directly in front of the device wall is modeled via logical sheath algorithms. In an open field line plasma, where the field lines are hitting the device wall the sheath region is critical. To model particle collisions, a Lenard-Bernstein collision operator was implemented.

With this setup we performed the first gyrokinetic particle-in-cell simulations of a well-studied one-dimensional parallel transport problem during an edge-localized mode in the SOL. This case was studied in the collisionless and collisional case. After thorough investigation and successful comparisons with previous continuum gyrokinetic simulations, we were confident to extend the model to three-spatial dimensions to study 3D open field line systems.

For the extension of the code to 3D, we implemented a so called "2D1D solver" with a normal Poisson solver in the 2D plane and a Fast Fourier Transformation (FFT) in the third spatial dimension. As a first 3D application we studied ion temperature gradient (ITG) modes in a periodic screw pinch setup without sheath boundary conditions. We

chose this simple setup to test the main 3D features of PICLS, such as the particle pusher and the 2D1D field solver against analytic solutions. This was an important validation and verification step since for open field line systems no analytical solutions are available. In a first attempt, as a final test case for this thesis we selected a 3D helical slab setup to study turbulence in helical open field lines. This test case was previously studied by a continuum gyrokinetic code and the results serve as comparison for our simulations. Applying this setup, most of the open field line features implemented in PICLS can be applied and first curvature-driven turbulence results can be obtained with PICLS. As this is only a first attempt, further studies and verification tests will have to be done in the future.

With developments and concepts introduced in this thesis a major step towards a gyrokinetic particle-in-cell code for transport and turbulence studies in the Tokamak scrape-off layer region was taken. The knowledge gained from this work can be used to independently study open field line configurations, or to further extend existing plasma core models towards the plasma edge/SOL region.

Zusammenfassung

Numerische Plasma-Simulationen spielen eine Schlüsselrolle bei der Entwicklung hin zu Fusionsenergie-Reaktoren. Sowohl in Bezug auf Fusionsforschung, also auch zur Risikominimierung während des Planungsprozesses von Fusionsreaktoren, sind numerische Modelle ein wichtiger Baustein. Vor allem von fünf-dimensionalen gyrokinetischen Turbulenz-Simulationen glaubt man, dass diese ein gutes qualitatives Verständnis und präzise quantitative Daten liefern können. Im Plasmainneren sind diese Modelle bereits für Transport- und Turbulenz-Simulationen etabliert. Im Plasma-Randbereich, nahe der Gefäßwand, stehen numerische Modelle hingegen immer noch erst am Anfang die dort vorherrschende vielfältige Physik zu erfassen. Da diese Region von grundlegender Bedeutung für die erreichbare Fusionsenergie und die Lebensdauer der dem Plasma ausgesetzten Komponenten ist, müssen die dort vorherrschenden zusätzlichen Herausforderungen verstanden werden. Speziell steile Dichte- und Temperatur-Gradienten und große Fluktuations-Amplituden sind dominant in dieser Region und verlangen die Verwendung der gesamten Verteilungsfunktion ("full-f"). Aktuell verfügbare gyrokinetische Codes für das Plasmainnere können diese Effekte noch nicht behandeln und deshalb ist die Entwicklung neuer gyrokinetischer Modelle für die Plasma-Peripherie notwendig. Einige Entwicklungen in diese Richtung sind bereits gestartet, doch es gibt noch zahlreiche unbeantwortete Forschungsfragen.
Aus diesem Grund stellen wir in dieser Arbeit den neu entwickelten Code PICLS vor, der speziell auf die Erforschung von Turbulenz in Plasmen mit offenen magnetischen Feldlinien abzielt, wie sie in der äußersten Schicht eines Tokamak-Reaktors (der sogenannten "Scrape-off Layer (SOL)") vorliegen. PICLS ist ein gyrokinetischer Particle-in-Cell Code und basiert auf einem elektrostatischen full-f Modell mit linearisierter Feldgleichung und kinetischen Elektronen. Für die Diskretisierung werden B-Spline Finite Elemente benutzt. Die Plasmaschicht, die direkt vor der Reaktorwand entsteht ("Sheath"), wird durch "logical sheath - Algorithmen" modelliert. In einem Plasma mit offenen Feldlinien, wo die Feldlinien auf die Reaktorwand auftreffen, ist die Sheath-Region entscheidend. Für die Teilchenstöße wurde ein Lenard-Bernstein Stoßoperator implementiert.
Mit dieser Konfiguration konnten wir die ersten gyrokinetischen Particle-in-Cell Simulationen eines bereits gut erforschten ein-dimensionalen parallelen Transport-Problems während einer am Rand lokalisierten Mode ("ELM") im SOL durchführen. Dieser Fall wurde mit und ohne Stöße analysiert. Nach gründlichen Untersuchungen und erfolgreichem Vergleich mit vorherigen gyrokinetischen Kontinuum-Simulationen, haben wir unser Modell auf

drei räumliche Dimensionen erweitert, um damit 3D Systeme mit offenen Feldlinien zu untersuchen.
Für die 3D-Erweiterung des Codes wurde ein so genannter "2D1D" Löser mit einem einfachen 2D Poisson Löser innerhalb einer Ebene und einer Fast Fourier Transformation (FFT) in der dritten räumlichen Dimension eingeführt. Als erste 3D Anwendung wurden Ionen-Temperatur-Gradient (ITG) Moden in einem periodischen "Screw pinch" ohne Sheath Randbedingungen studiert. Diesen einfachen Aufbau haben wir gewählt, um die wichtigsten 3D Features in PICLS, wie z. B. den Teilchen Pusher und den 2D1D Feldlöser, an Hand analytischer Lösungen zu testen. Dies war ein wichtiger Validierungs- und Verifikationsschritt, da für Systeme mit offenen Feldlinien keine analytischen Lösungen zur Verfügung stehen.
In einem ersten Versuch haben wir als finalen Testfall für diese Arbeit eine 3D spiralförmige Slab Konfiguration gewählt, um Turbulenz in spiralförmigen offenen Feldlinien zu erforschen. Dieser Fall wurde zuvor bereits mit Hilfe eines gyrokinetischen Kontinuum Codes untersucht und dessen Resultate dienen uns als Vergleich für unsere Simulationen. Durch die Verwendung dieses Aufbaus können der Großteil der in PICLS implementierten Features im Gesamtverbund getestet und erste Krümmungs-getriebene Turbulenz-Resultat mit PICLS erzielt werden. Da es sich bei den präsentierten Ergebnissen um einen ersten ambitionierten Versuch handelt, werden zukünftig noch weitere Untersuchungen und Verifikations-Tests gemacht werden müssen.
Mit den Entwicklungen und Konzepten, die in dieser Arbeit eingeführt wurden, konnte ein großer Schritt hin zu einem gyrokinetischen Particle-in-Cell Code für Transport- und Turbulenz-Studien in der Tokamak Scrape-off Layer Region gemacht werden. Das in dieser Arbeit gewonnene Wissen kann dazu genutzt werden unabhängig offene Feldlinien-Konfigurationen zu studieren, oder existierende numerische Modelle für das Plasmainnere in Richtung Plasmarand und SOL Region zu erweitern.

Contents

Abstract **i**

Zusammenfassung **iii**

1 Introduction: No virtual tokamak without the edge **1**
1.1 Tokamaks . . . 2
1.2 Scrape-off layer (SOL) . . . 3
1.2.1 SOL physics . . . 5
1.3 SOL modeling . . . 8
1.3.1 Fluid models . . . 9
1.3.2 Gyrokinetic models . . . 10
1.4 Motivation and outline . . . 13

2 The model behind PICLS **17**
2.1 Gyrokinetic model . . . 17
2.1.1 Derivation of gyrokinetic Lagrangian . . . 18
2.1.2 Lagrangian approximations . . . 21
2.1.3 Euler-Lagrange equations . . . 23
2.1.4 Polarization equation . . . 25
2.1.5 Summary . . . 26
2.2 Equations in slab geometry . . . 28
2.2.1 Reduction of B-field equations . . . 28
2.2.2 Equations of motion . . . 28
2.3 System of units and normalization . . . 29
2.3.1 System of units . . . 29
2.3.2 Normalized variables . . . 29
2.3.3 Normalized set of equations . . . 30
2.4 Total energy for the electrostatic system . . . 31

3 Sheath model **33**
3.1 Physical Debye sheath . . . 33
3.2 Logical (or insulating) sheath . . . 35
3.2.1 Algorithm . . . 36

3.3 Conducting sheath . . . 38
3.4 Non-perpendicular B-field on sheath . . . 40
3.4.1 Rotation of the gyro-ring . . . 41
3.4.2 Velocity perpendicular to boundary . . . 43
3.4.3 Reflection along B-field line . . . 44
3.5 Additional considerations for sheath models . . . 45
3.6 Source term . . . 46

4 Collision operator **47**
4.1 Collisions in the SOL . . . 47
4.2 Coulomb collisions and Landau operator . . . 47
4.3 Lenard-Bernstein (LB) collision operator . . . 49
4.4 PIC discretization of LB collision operator . . . 51
4.4.1 Conservation of moments . . . 53
4.5 Test of collision operator . . . 56
4.5.1 Basic tests with constant $\mathbf{u}_{\parallel}$ and v_T . . . 56
4.5.2 Test of conservation features and relaxation . . . 58
4.6 Summary . . . 62

5 PICLS: Numerical methods **63**
5.1 Background and general algorithmic structure . . . 63
5.2 Main properties and finite element methods . . . 64
5.2.1 Distribution function discretization . . . 66
5.2.2 Finite Element method: B-splines . . . 66
5.2.3 1D electric potential solver . . . 70
5.2.4 3D electric potential solver . . . 73
5.3 Data structures . . . 91
5.4 Parallelization . . . 92
5.4.1 Parallelization: particles . . . 92
5.4.2 Parallelization: field solver . . . 93
5.5 Scaling tests . . . 94
5.5.1 OpenMP scaling / intra-node . . . 95
5.5.2 MPI scaling / inter-node . . . 95

6 1D SOL simulations **99**
6.1 Introduction . . . 99
6.2 Physical model implemented . . . 101
6.2.1 Gyrokinetic electrostatic model in 1D1V with kinetic electrons . 101
6.2.2 Modification of the Polarization equation . . . 102

6.2.3 Geometry of the ELM heat-pulse problem . . . 104
6.3 Simulation setup . . . 104
6.3.1 Initial conditions . . . 105
6.3.2 ELM and inter-ELM phase . . . 106
6.4 Simulation results: collisionless 1D1V model . . . 108
6.4.1 ELM phase spatial profiles . . . 108
6.4.2 Divertor heat flux and sheath potential . . . 110
6.5 Simulation results: collisional 1D2V model . . . 115
6.6 Conclusions . . . 120

7 ITG simulations in screw pinch 123
7.1 Model details . . . 123
7.1.1 B-field and Euler Lagrange equations . . . 124
7.1.2 ITG instabilities in slab geometry . . . 127
7.2 Simulation setup . . . 131
7.3 Simulation results . . . 133
7.3.1 Particle trajectory tests . . . 133
7.3.2 Potential solver test . . . 136
7.3.3 Slab-ITG simulations . . . 137
7.4 Conclusions . . . 142

8 3D helical slab SOL simulations 149
8.1 Introduction . . . 149
8.2 Model . . . 150
8.3 Simulation setup . . . 152
8.3.1 Relevant parameters . . . 152
8.3.2 Initial conditions . . . 153
8.3.3 Simulation phase . . . 154
8.4 Simulation results . . . 155
8.4.1 Simulations with “large” time step $dt \approx 3.7$ns . . . 156
8.4.2 Simulations with “small” time step $dt \approx 0.1$ns . . . 159
8.5 Conclusions . . . 161

9 Summary and outlook 163
9.1 Summary . . . 163
9.2 Outlook . . . 165

Bibliography 169

Acknowledgments 181

1 Introduction: No virtual tokamak without the edge

The global population is already approaching 8 billion people and according to calculations of the United Nations Department of Economic and Social Affairs / Population Division this number will exceed 11 billion by 2100 [1]. Several reports state that despite some energy efficiency optimization effects, the world energy consumption will further increase, amongst others due to increasing economic wealth in developing countries (e.g., [2], [3]).
Unfortunately, energy production from fossil fuels is one of the key human contributions to global warming. This statement is still not globally accepted, but several international institutions support it (e.g., see a report from the Intergovernmental Panel on Climate Change [4]). Independent whether one supports this theory or not, due to the massive social pressure major governments are shifting away from traditional energy sources and thus alternative technologies for energy production become more and more relevant. An example for this is Germany, where the exit from coal energy until 2038 was decided by the government in 2018, or the European Union that released an ambitious "Green Deal" in 2019 to reduce greenhouse gas emissions. Also in the light of the upcoming electric mobility, a pre-condition to create a really significant, rather than only an idealistic impact on carbon dioxide reduction, is to produce electricity from alternative, low emission (especially carbon dioxide) energy sources.
Renewable energies have shown to be a key contributor for future greenhouse gas neutral energy production, but recent studies expect that they will not be able to cover all global requirements in mid-term future (e.g. [3]). However, if this is true or not can not be said, yet, as this also heavily depends on the social and political acceptance of these technologies. From a technical point of view, especially their volatility to environmental conditions is problematic for base load production. Nuclear fission was also seen as a significant source for the increasing energy consumption, but drastic events like Tschernobyl and Fukushima provoked severe skepticism within the public opinion towards this technology. In the course of these events, society is becoming more and more reluctant towards fission and a shift in mindset is not foreseeable.
A promising alternative that is under discussion, especially for base load energy production, is nuclear fusion. As an upcoming technology it still has to prove its viability

as a reliable and essential energy source. The large-scale experiment ITER, which is currently under construction and expected to have its first Deuterium-Tritium (D-T) plasma burning in 2035, is a key step towards checking this viability. However, alone for the construction of an experiment of that size investments of EUR 18-22 billion are planned [5]. To set this number in relation, the construction cost of USD 3.9 billion of the tallest skyscraper of New York, the One World Trade Center, shall be noted for comparison. In projects of that size small deviations or false assumption of machine specifications can lead to significant delays and increase in cost. From a cost and project management view mitigation measures and a thoughtful preparation are key for success. An important and extremely valuable building block for risk mitigation, but also for further developments in fusion research are numerical plasma simulations (see e.g., [6]). Since one of the most promising concepts for future fusion energy production and also the concept of ITER, is the tokamak, simulations of this concept are of particular interest.
Plasma theory and simulation has advanced significantly during the past decades, but to model a whole virtual tokamak, still certain areas must be further investigated. One key topic to study is the plasma periphery, including the so-called edge and scrape-off layer, where the plasma approaches the machine wall. Exactly in studying models for the turbulent plasma fluxes towards the device wall lies the focus of this thesis. But before we start to further detail our motivation some relevant terms need to be introduced in the following sections.

1.1 Tokamaks

A plasma generally consists of charged particles (electrons and ions) that balance out towards neutrality on larger scales, which is called quasi-neutrality. Plasmas that are relevant for fusion reactors must be heated up to several million Kelvin so that the positively charged nuclei can overcome the repulsive electric force. Due to the extreme heat required for the fusion reaction, the plasma needs to be confined and insulated against thermal losses. In tokamaks this is done via magnetic fields. Magnetic confinement basically uses the Lorentz force which obliges charged particles on helical orbits around the field lines. Thus, the particle is free to move along the magnetic field line, but perpendicular to it its motion is forced on a helical trajectory ([7], [8], [9]).
Up to today, the most developed and successful reactor concept is the tokamak, which is also the concept of choice for ITER. Another alternative concept with constantly improving and promising results is the stellarator, such as Wendelstein 7-X ([10], [11]). However, the development status of these machines is still considered less mature than in the tokamak case.

In tokamaks, by placing a set of planar electric coils a strong magnetic field is created that confines the charged particles on helical orbits along the field lines. One might expect that by assembling the coils in a ring and thus closing the magnetic field lines in a ring, the particles could already be confined within this ring. However, various drift effects due to the inhomogeneity of the magnetic field lead to very poor confinement properties of such a configuration. Such a machine has a magnetic field proportional to its radial distance to the symmetry axis R of the machine, $B_{tor} \propto R^{-1}$. Thus, it is maximal close to the symmetry axis. Together with the curvature drift it creates a vertical charge dependent particle drift. This implies that the electrons and ions are vertically separated and the evolving electric field subsequently causes an additional drift in radial direction for both kinds. Hence, the plasma confinement is lost.
To solve this problem and obtain stable plasma confinement, field lines need to be twisted in helical shape. Therefore, a current is driven in the plasma, which induces an additional poloidal magnetic field, that adds up to the purely toroidal external field introduced by the circularly arranged field coils. The transformer principle is applied to drive the current, by placing the primary coil in the center of the plasma ring. As a secondary coil, the plasma itself is used. Due to the resistivity of the plasma, an increasing current needs to flow in the central transformer coil. Because of this, tokamaks are nowadays operated in pulsed mode, but for future devices also other operational modes are planned. The resulting helically twisted magnetic field lines compensate the particle drifts. The toroidal field component is in general much stronger than the poloidal contribution and thus toroidally nested closed flux surfaces are created ([12], [7], [8], [9]).
For a schematic representation of the concept see figure 1.1.

1.2 Scrape-off layer (SOL)

Despite the fact, that the constructed magnetic field in a tokamak confines the charged particles on nested flux surfaces, the confinement is still not perfect. Turbulent behavior of the plasma due to micro instabilities and collisions with other particles cause particle and heat transport across these flux surfaces [7, 13]. And even disruptions - macroscopic instabilities - can appear and cause an extinction of the plasma and even more severe, large power and force loads on the materials that surround the plasma [14]. But already in case of smaller instabilities, when the plasma has contact with the reactor wall, it provokes a sputtering of impurities into the plasma region. This has to be minimized, since the impurities are a key contributor to radiation loss and can severely reduce the fusion yield [12].
To avoid these negative effects, a controlled plasma-wall contact needs to be established,

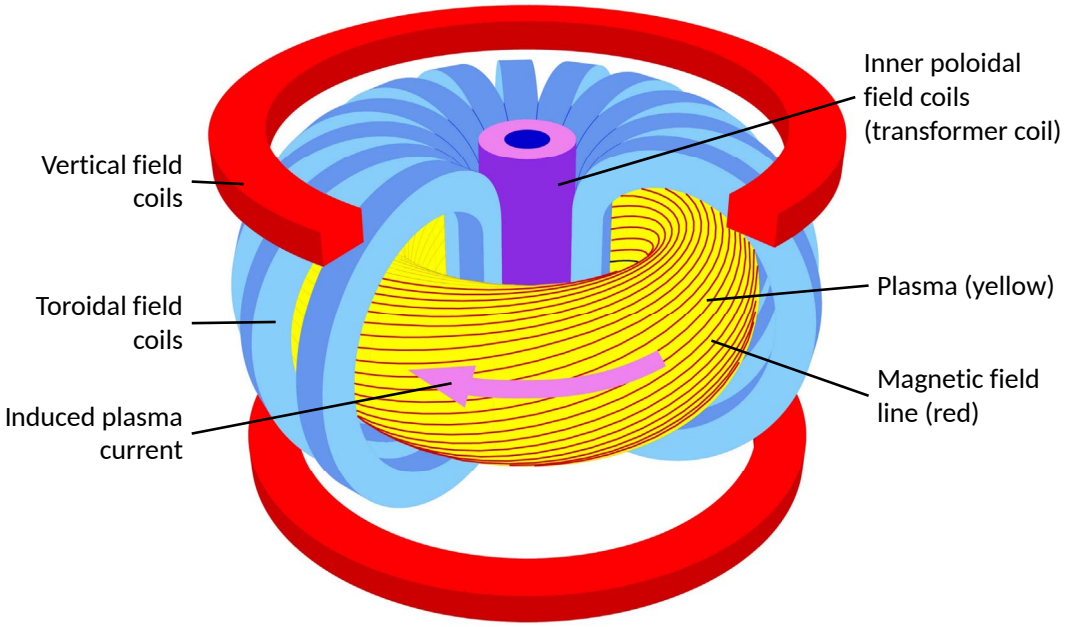


Figure 1.1: Schematic illustration of the tokamak fusion reactor concept. (Source: Max-Planck Institute for Plasmaphysics (IPP) Garching, adapted)

to guide impurities towards defined positions at the wall material and therewith prevent contact with other regions. In modern machines, a divertor concept has proven most successful and will also be used for ITER. Here, additional coils cause a poloidal field null point at one (or sometimes more regions), the so-called X-point. Additionally, the plasma-wall contact is shifted away from the hot plasma region, to guarantee a pure plasma center ([15], [12]). An alternative to the divertor is the limiter concept, which has shown to be inferior, e.g. in terms of helium ash and impurity influx into the confined plasma region. Thus, using a divertor also was a key success factor to discover the high confinement mode, H-mode, in ASDEX ([16], [17]).
Looking at the poloidal cross-section of the ASDEX Upgrade tokamak with a divertor geometry, shown in figure 1.2, three main plasma regions can be distinguished: Core, edge and scrape-off layer (SOL) region.

The core is the region, where fusion power is produced. It is the innermost and hottest region of the plasma. Here temperatures have to exceed at least $\sim$ 10keV (equal to $\sim$ 100 million °C) to reach the ignition condition for D-T fusion ([18]). The magnetic flux surfaces in the core region are nested and closed. The edge region is the subsequent region further towards the plasma wall, which also is characterized by closed flux surfaces. It is a rather thin layer and compared to the core often exhibits

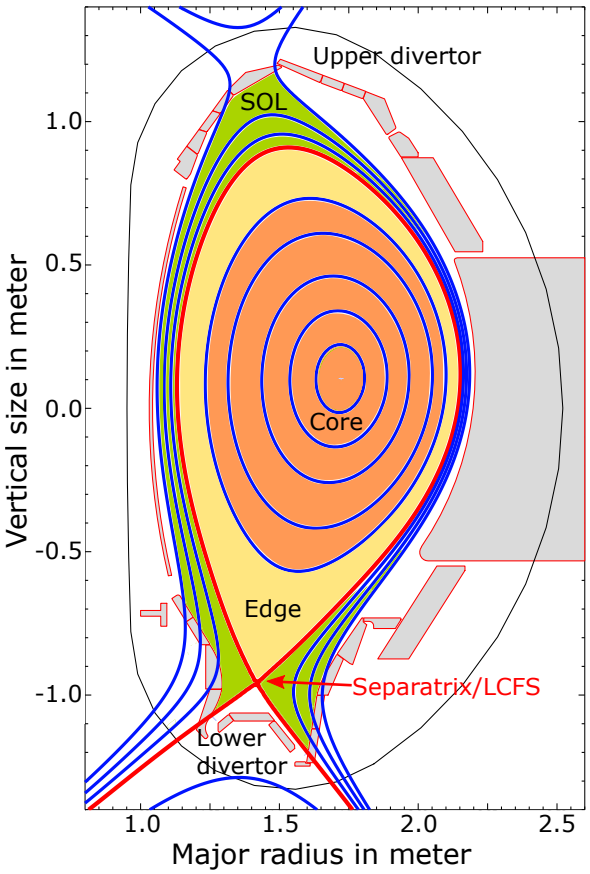


Figure 1.2: Illustration of the core, edge and SOL region in a poloidal cross-section of the ASDEX Upgrade tokamak with divertor geometry (profile: #17151, 4s). The last-closed flux surface (LCFS) is indicated in red and separates clearly edge (yellow) and SOL (green) plasma. The blue-lines resemble the magnetic flux surfaces and are reconstructed from the real shot. In the core (orange) and edge region, the flux surfaces are closed, whereas in the SOL the flux surfaces are open.

steep pressure gradients [19].
Once crossing the last closed flux surface (LCFS or Separatrix) an often even thinner layer of open field lines begins, the scrape-off layer (SOL). Here, the open field lines wind around toroidally several times before they hit the divertor target. Thus, in the SOL energetic particles are lost to the machine surface. Further details on the SOL are noted in section 1.2.1, as for our work we are mainly interested in this region. In general, the greatest part of the plasma lies within the core and edge region. This is required, to achieve high enough confinement for the fusion reaction. Particles that would be confined by the closed flux surfaces, however, can be transported across the surfaces by turbulence and collisions [20].

1.2.1 SOL physics

In the SOL, plasma that is transported out of the confined region, in general is lead to the divertor targets before it can reach the wall. This region is highly important for regulating the impurity level and confining the plasma. Plasma dynamics in the SOL

are defined by a balance of plasma outflow from the core, turbulent transport across magnetic field lines, parallel flow and losses towards the divertor targets [20–22]. Typical characteristics for the SOL are low plasma density and temperature ($T_e \sim 10 - 100$eV, due to the plasma-wall interaction [19]) high collisionality and large-scale turbulences of amplitudes of order of unity compared to the background (in density, electron temperature T_e and electrical potential). Experimentally measured profiles confirmed these large amplitude fluctuations and the exponential decay of radial profiles [7].

Parallel & cross-field transport

In parallel direction along the field lines, electrons and ions rapidly flow towards the divertor targets, where they re-combine and thus are lost for the plasma [21]. Initially electrons hit the divertor faster than ions, due to their higher mobility. The plasma establishes a thin layer of net positive charge of a few Debye lengths (due to Debye shielding), directly at the plasma-wall interface, the so-called Debye sheath ([8, 21]). This layer builds up a potential barrier that accelerates incident ions into the wall and repels incident electrons (only the fastest electrons can overcome the potential barrier). The particle fluxes of both species are kept approximately equal, thus the plasma is maintained quasi-neutral. The SOL plasma properties and the particle and energy flux towards the surface are significantly influenced by this sheath.
As mentioned earlier, in addition to the predominant parallel transport along the field lines also turbulence driven cross-field transport appears. In the SOL, however, radial transport is not only driven by turbulence or collisions, but also due to convection of coherent structures called filaments or blobs [23–26]. Typical scales for blobs are $\sim 1 - 10$m parallel to the field lines, due to the fast parallel charged particle motion and much smaller scales of $\sim 1 - 10$cm perpendicular to the field, which leads to an elongation along the field line. This fast transport towards the plasma chamber walls - not as desired along the field lines to the divertor - can damage the wall material and lead to impurity contamination of the core plasma [24, 27]). Radiative cooling due to the introduced impurities eventually lowers the core plasma temperature and can reduce the fusion yield. The practical implication of blobs on fusion reactors is still under research [28, 29]. In this thesis we will focus on the parallel transport towards the wall, since the majority of the lost power (also in a controlled fusion reactors) is lost along this path.

Power exhaust

A certain part of the fusion power from the plasma core is exhausted into the SOL as the outermost plasma region, where it has to be absorbed somehow. Since the exhaust

power increases with the machine size and power, controlling the exhaust is essential for the construction of large-scale devices, such as ITER. For ITER in the SOL the unmitigated steady-state parallel heat flux is expected to be ∼1 GW m^{-2} [30] and for DEMO even ∼20 GW m^{-2} [31]. However, the maximum tolerable heat flux normal to the divertor target for current materials is only ∼10 MW m^{-2} for steady-state and ∼20 MW m^{-2} for transient states [30] (e.g. from disruptions and edge-localized modes). To avoid damage or strong stress on the divertor, strategies are required to reduce the actual heat load hitting the target plates. These strategies are still under investigation and comprise radiative divertor detachment via impurities seeding (e.g., N, Ne) [32, 33], advanced divertor geometries [34] and applied resonant magnetic perturbations [35].
Although the power loads in ITER, which is planned to work with partial detachment, are already extreme, they are still expected to be manageable with the described strategies. For DEMO, however, the heat flux towards the divertor is significantly higher and remains a major problem that needs to be solved.
One of the main challenges is the extremely narrow heat-flux width at the outer midplane. Studies on current experimental machines showed that this width is not sensitive to the machine size, but at least for conventional aspect ratio tokamaks, rather to an inverse scaling with plasma current. By extrapolating the scaling to an ITER H-mode plasma, an outer-midplane heat-flux width of only ~ 1 mm was determined [36, 37]. This is much smaller than the 5 mm heat-flux width specified for the ITER design. However, the validity of the empirical extrapolation to ITER is still unclear.
The small heat-flux width predicted by Eich et al. (2013) was also confirmed by a heuristic drift-based model of Goldston (2011). But in this model, no turbulence effects are included. Thus, the extrapolation to ITER SOL parameter regimes can be questioned, since for example no turbulent heat transport along the divertor legs can be modelled. Therefore in 2017, C. S. Chang et al. used the electrostatic gyrokinetic XGC1 code to study the heat-flux-width with turbulent simulations based on first-principles. For ITER, the simulations predicted an outer-midplane heat-flux width of ~ 5.9 mm, which would be in the ITER design specifications. The issue is that at the moment no other gyrokinetic code is capable to benchmark these findings.
This example shows the relevance of understanding the SOL physics and developing further expertise in first-principle based simulations in the SOL. The plasma confinement significantly depends on the power exhaust in the SOL [20] and thus is also of extreme importance for the realization of future power plants.

1.3 SOL modeling

The complex physics in the SOL region also increase the complexity for plasma modeling and simulation [40]. In addition to the effects known from the plasma core, also radiation losses, sheath and atomic physics have to be regarded. Apart from that, the size of plasma structures ranges from the gyro-radius to machine size with strong gradients perpendicular and along the field lines. Also time scales vary in a wide range between the gyro-motion and turbulence time scales. Fluctuations are of the order of unity compared to the background, which means that in simulations a linearization of quantities (i.e., splitting the distribution function in a constant background and a perturbation $f = \delta f + f_0$) as often applied in the plasma core is not valid anymore [7]. And finally, also the divertor geometry adds up to the increase in complexity.
Although there exist fully six-dimensional kinetic code developments [41, 42], large time and length scale tokamak simulations that solve the Vlasov–Maxwell or Vlasov–Poisson equations require processing and memory capabilities that are far beyond what nowadays supercomputers can provide. Even now, when we approach exascale computing, the simulations are still way too costly. To cope with these computational restrictions, physical approximations are required to reduce the numerical calculation efforts.
Therefore, simulations within the SOL have been and are still primarily performed with the help of fluid- & gyrofluid-based codes. There are different types of codes, differing in the physical and numerical models they use (for a brief overview see section 1.3.1). One of the most widely used codes and the main tool used for ITER edge/SOL-plasma and divertor modeling is certainly the SOLPS package [43–45]. The key advantage of fluid codes is their computationally relatively low effort, compared to kinetic approaches. These models have been of use to analyse the qualitative physics of edge & SOL plasma, but for reliable quantitative prediction of plasma properties kinetic simulations are believed to be required [46, 47]. Also direct or indirect coupling of fluid transport codes to kinetic turbulence codes is seen to be an option [48].
Due to these quantitative drawbacks, there are major efforts to develop gyrokinetic first-principle codes for the SOL and edge plasma. The underlying physical model is a 5D description of low-frequency plasma dynamics reduced from the initial 6D kinetic equations [49–51]. Currently the most advanced gyrokinetic code is probably XGC1, which gained popularity with its calculation of the ITER scrape-off layer width [39], but also other developments are currently on-going (see section 1.3.2). The key advantage of this method is the reduced computational cost, compared to kinetic models and the intrinsic capability to study turbulent behavior with kinetic effects.
Due to their wide applications and expected up-coming relevance for edge and SOL simulations, selected fluid and gyrokinetic code developments are presented in the following section. Additionally, we want to note that all the simulations in this thesis

are designed to test the theoretical models and are based on generic modeling, where hypothetical situations are modeled with simplified assumptions.

1.3.1 Fluid models

Since the purpose of this thesis is to develop a code for gyrokinetic SOL simulations, in this section we only want to mention the main physical models used for fluid simulations and highlight specific developments. For a broader overview we refer to [52] and [53].
A large group of codes is calculating simplified transport models based on Braginskii fluid equations in 2D with assumed axisymmetry and are therefore also referred to as boundary-plasma transport codes. Unfortunately, plasma turbulence is not intrinsically included in these models. Thus, to simulate turbulent transport across magnetic flux surfaces, a coupling to turbulence codes, or the use of anomalous diffusion terms (based on experimental studies) is needed. The earlier mentioned SOLPS package is part of this group and contains a two-dimensional edge/SOL-plasma transport component, which for neutrals transport is coupled to a kinetic Monte Carlo code [54]. However, SOLPS still is the main tool for ITER SOL simulations since several important physical characteristics, such as impurities, ionization and radiation are implemented.
Another larger group of codes are boundary plasma turbulence models that solve drift-ordered fluid equations in 2D or 3D. Usually, they are derived by using a low-frequency approximation to the Braginskii equations. The frequencies of the gyro-motion are therefore ordered fast compared to the relevant frequencies [55]. These codes are computationally more expensive, but compared to the transport codes turbulence is directly included in the model. An example for this code class is the plasma turbulence code GRILLIX [56–60] that uses a global electromagnetic drift-reduced Braginskii model [53]. The key feature of this code is the application of a flux coordinate independent approach (see e.g., [58, 61]), which has the ability to cross the separatrix and perform X-point simulations.
A last big class of fluid codes are so-called gyrofluid codes, which solve 3D electromagnetic gyrofluid equations. These models can simulate finite-Larmor-radius and Landau-damping effects (see e.g. [62]). They can treat ion-gyroradius scale dynamics and even smaller. In this regime, drift-ordered fluid models normally break down [63]. But not all potentially relevant kinetic effects are captured, such as trapped particles, due to the few moments they keep in the models [64]. Two examples for these types of code are the BOUT++ [65] code and the GEM3 model [66].
But as already mentioned in section 1.3, fluid and gyrofluid approaches were quite successful in producing qualitatively viable results, but for quantitative studies, kinetic simulations are believed to be required at least to some extent [46].

1.3.2 Gyrokinetic models

As described earlier, in a gyrokinetic model, the original 6D problem is reduced to a 5D problem by applying small parameter assumptions and therewith the model focuses on low-frequency plasma dynamics. The fast gyromotion thus can be systematically removed from the description of the dynamics, by assuming strong magnetic background fields as present in tokamaks and stellarators [50, 51]. This results in a considerable simplification and improved computational cost.
Especially, in the plasma core these gyrokinetic simulations are nowadays widely used and produce comparable turbulence results across various codes (see e.g. [67]). But due to the extremely rich physics in the plasma edge/SOL (see section 1.3), these models are much less developed in this region. Recent efforts already suggested consistent models for the edge/SOL regions ([68]), but their applicability in the plasma periphery still has to be proven. To only name a few challenges that are especially relevant for gyrokinetic models in this region: large amplitude fluctuations, sheath boundaries, wide range of space and time scales, atomic physics, etc. As a result, code developments in the SOL need to specifically address these complications.
Most commonly, there are two basic methods applied to numerically solve the gyrokinetic system, particle-in-cell (PIC) methods [69, 70] and continuum methods [71]. Also a third hybrid method of the two, called semi-Lagrangian exists, but it is less widely used for gyrokinetic modeling and thus, we only want to refer to the development done in the GYSELA code [72].
In a PIC code Monte Carlo methods are applied that approximate integrals over phase space (e.g., the particle distribution function) by a finite number of markers [73]. In a more tangible picture, the markers can be described as "superparticles" that comprise many physical particles. The number of markers used in practice is much lower than the physical particle number [74]. These markers are initially distributed in the phase space and according to their positions, the 3D field equations are solved from this charge distribution on a fixed grid. Then the resulting fields are interpolated back to the marker positions and using these fields the markers are advanced to their new position according to the gyrokinetic Euler-Lagrange equations. This procedure is then repeated again and again for the new distributions.
In the continuum case (also called "Eulerian") on the other hand, the 5D gyrokinetic equations are discretized on a fixed phase-space mesh and solved on this mesh. The resulting partial differential equations are then solved with numerical methods like finite-volumes, finite-elements, etc. Equal to the PIC approach, also for continuum codes the 3D fields are calculated by using grid-based algorithms [71].
Both methods have their advantages and disadvantages, compared to the other. One of the main challenges of PIC algorithms is their inherent statistical noise in the distribution

function, due to their Monte Carlo sampling approach. For N markers, this error scales with $1/\sqrt{N}$. This effect is extremely relevant for the validity of results and thus several techniques for noise reduction have been developed [75–78]. Another often discussed problem resulting from statistical noise is the so-called Ampère's law cancellation problem in electromagnetic simulations, where two large terms in the Ampère's law cancel out numerically at average to high plasma β and small perpendicular wave numbers. But since its discovery, various mitigation techniques were introduced with convincing results [79–82] and nowadays it is not seen as a showstopper anymore.
For continuum models however, Monte Carlo related problems do not play a role. A key challenge for this method is their restriction of the time-step size for explicit time stepping, according to the Courant-Friedrichs-Levy (CFL) condition [75]. Thus, even when the fastest dynamics do not influence the simulation results, they still have to be resolved. Using semi-implicit and implicit time-steps as mitigation can be computationally very costly. Another issue that needs to be addressed in continuum codes is that the positivity of the distribution function is not automatically maintained [75].
Apparently, numerical models and challenges behind PIC and continuum codes are quite different. Thus, to be able to perform reliable quantitative studies, it is important to develop both further and use cross-checks between both methods for validation.

Current code developments

The development of gyrokinetic continuum codes in the edge/SOL plasma was lagging behind drastically until only several years ago. As investigated by R. Cohen et al. 2008 three main codes - G5D [83], TEMPEST [84] and FEFI [66] - made efforts to study SOL physics, but eventually did not further follow this path, due to difficulties and not reported causes. Hence, in more recent papers of the codes, no further efforts in this direction could be found. However, several rather new developments are currently pushing the code capabilities further towards the plasma boundary.
In the range of continuum codes, the COGENT code development, was originally based on a fourth-order finite-volume discretization scheme for 4D (axisymmetric) simulations and spanned from the plasma core, across the separatrix into the SOL. To deal with the strong plasma transport anisotropy and the complex X-point divertor geometry COGENT applies a mapped multiblock grid technology. COGENT is constantly extended and in a recent effort, an extension to a 5D phase space (3D2V) to study fully kinetic turbulences in a tokamak edge region is addressed [85–88]. Another very promising development is the Gkeyll code developed at the Princeton Plasma Physics Laboratory. Gkeyll is a gyrokinetic code based on discontinuous Galerkin algorithms and can handle open field line plasmas and plasma-wall interactions, such as present

in the SOL [89–91]. Recently, gyrokinetic plasma turbulence simulations in the Texas Helimak device, a simple magnetized torus with features similar to the tokamak SOL, were modeled with the help of Gkeyll [92] and also electromagnetics were introduced into the model recently [93]. The GENE code, which is based on a finite-difference discretization and widely used for plasma core turbulence simulations, has chosen a similar approach and started to expand its capabilities towards the plasma edge and SOL by implementing sheath boundary conditions [94, 95]. In a very recent work within the GENE code family, the GENE-X code (for more details on GENE-X see [96]) was developed, which uses a locally field-aligned coordinate system following the flux-coordinate independent approach of the previously mentioned fluid code GRILLIX. This development is still on-going, but also very important for the field, since GENE with its wide range of capabilities is a suitable model for quantitative benchmarks with other boundary plasma codes, such as the previously mentioned XGC1 code.
On the PIC side, XGC1 is at the moment surely the most developed gyrokinetic code that can model the plasma edge and SOL. With XGC1 simulations across the separatrix are possible, where in a recent study turbulent fluxes across the separatrix were analyzed [97]. In another very recent publication, neoclassical and electrostatic turbulent transport under resonant magnetic perturbations (RMPs) was studied in a DIII-D H-mode edge plasma [98]. The current model includes a realistic divertor geometry, charge-exchange and ionization interactions, neutral particles and radiation cooling. With this advanced setup, XGC1 was able to produce the often discussed and previously mentioned (see section 1.2.1) $\sim$ 5.9mm outer-midplane heat flux width result [39]. To improve confidence in the heat-flux width calculation for ITER, XGC1 also could reproduce the heat-flux values for three major current experiments. Since the dominant effect in the ITER case was turbulent electron-heat-flux spreading, the relevance of gyrokinetic simulations for SOL simulations becomes obvious. However, the simulations performed for this study were extremely costly and thus other cheaper codes and varying physical models should also be studied. Another interesting but especially in terms of geometry less advanced 5D development is the ELMFIRE code [99]. ELMFIRE is capable of simulating a circular concentric background and includes a limiter scrape-off layer. For a recent study, logical boundary conditions were implemented to perform electrostatic simulations of a limited plasma [100].
In summary, compared to current PIC code developments, continuum approaches caught up significantly but are still not at the same level. Further improvements in both approaches are essential to achieve reliable and comparable quantitative results.

1.4 Motivation and outline

To achieve the goal of a complete virtual tokamak it is still a far way to go. One of the key challenges is to be able to qualitatively and quantitatively model the edge and SOL region of a fusion plasma. For quantitative studies especially gyrokinetic models have proven to be reliable and still computationally affordable in the core region. In the edge region first promising results were achieved, but further research has to be done. Essential topics to be investigated in this region are fluxes and plasma-wall interaction. In this thesis we address exactly these points, by developing a gyrokinetic full-f particle-in-cell code for open field lines in the SOL, called PICLS. Of specific interest in our case are sheath models for the Debye sheath, which build up right before the wall material or divertor and thus are influencing the particle and heat flux towards the wall. Another important aspect is the application and behavior of the finite-element discretization scheme we implement in PICLS.
Since PICLS is a new code developed from scratch, a step by step approach towards relevant physical models is required. Hence, we start with a rather simple simulation setup and then gradually progress towards more complex and realistic systems. In practice this means that we start with a one-dimensional slab geometry and then extend our simulations towards more challenging three-dimensional cases. An intermediate step for our code development is to study helical field lines, such as TORPEX [101] or simple tokamaks. Since the occurrent sheath physics are very important, this reduction of complexity also helps us to focus on the relevant sheath effects. For more advanced geometries like tokamak experiments, major development efforts will be required.
As mentioned before, a comparison of different algorithmic models, especially for quantitative gyrokinetic studies, is essential. Thus, the results produced with PICLS, are also relevant to benchmark and compare with other SOL code developments like GENE and Gkeyll. On the path towards full virtual tokamak simulations, the algorithms developed in PICLS could also be implemented in existing PIC codes for the plasma core, such as ORB5, to extend its capabilities towards the plasma edge and SOL. Or coupling the SOL code PICLS to a dedicated core plasma code could also be a potential solution.
In chapter 2 we therefore introduce the main concepts and equations of the applied physical model. We start with deriving our gyrokinetic Lagrangian with an electrostatic particle Lagrangian. From this we derive the required Euler-Lagrange equations and the polarization equation for the calculation of the electric field. The special case of the equations in slab geometry and the applied normalization schemes are also presented. We conclude this section with deriving the total conserved energy in the electrostatic system.
Chapter 3 is dedicated to the introduction of the sheath models we implemented for the

plasma wall boundary. Therefore, we first shortly explain the basic principle of a Debye sheath that appears directly in front of the device wall. Since in gyrokinetic simulations we are not able to resolve this very small region, logical sheath algorithms need to be used. The sheath models and algorithms implemented in PICLS are presented in the following and additional considerations for further development of these models are shortly discussed. Apart from this, we also introduce our method on how to treat particle reflections at the plasma wall for non-perpendicular incident B-field angles.
Another relevant ingredient for realistic plasma simulation in the SOL are collisions. Hence, in chapter 4 we specify the collision operator employed in PICLS. We shortly cover Coulomb collisions and the Landau operator to then focus on the implemented Lenard-Bernstein collision operator. The discretization of the operator via the Langevin approach and its conservation properties are discussed and several conservation tests are performed.
Since PICLS is a new code development, in chapter 5 the main numerical methods we used are explained. We start with its algorithmic structure to quickly dive into the applied Finite Element methods. Here, we cover its discretization via B-splines Finite Elements and discuss the employed 1D and 3D solver. The 3D solver uses a normal 2D solver routine in a 2D plane and an FFT in the third dimension. We conclude this part with describing PICLS' main data structures, its parallelization scheme and qualitative scaling tests.
With chapter 6 we start to apply PICLS to plasma turbulence simulations. We initially focus on a simplified 1D ELM heat-pulse problem without collisions, which was already previously studied by gyrokinetic continuum codes. After the introduction of a second velocity component we repeat the same test case with collisions in a 1D2V setup and again compare with previous results.
With the gained confidence from the 1D test case, we then apply PICLS to a 3D test case in chapter 7. Here, we study ion temperature gradient (ITG) modes in a periodic screw pinch setup without sheath boundary conditions. As part of this chapter, we use this simple 3D configuration to test the main 3D code features, such as the particle pusher and the 2D1D field solver. Since for open field line systems no analytical solutions are available this is an important validation and verification step.
For our simulations in chapter 8 we select a setup, where most of the previously introduced SOL features implemented in PICLS can be applied. This implies a 3D helical slab configuration to study turbulence in helical open field lines with sheath boundary conditions. Therewith plasma blob formation and convective radial transport of plasma blobs can be investigated. Since this test case was previously studied by a continuum gyrokinetic code, the results serve as comparison for our simulations. The results we show are only a first attempt and hence, further studies and verification tests will have to be done in the future.

The main results of this work are summarized in chapter 9. Here we also give an outlook on potential future research priorities and a near-term development plan.

2 The model behind PICLS

2.1 Gyrokinetic model

In general, the physical model behind PICLS is based on gyrokinetic (GK) theory. Here, one tries to reduce the original 6D Vlasov-Maxwell problem (3 velocity and 3 spatial coordinates) to a 5D problem (2 velocity and 3 spatial coordinates), by describing the plasma particle motion in terms of drifts of the gyrocenter, rather than combining gyromotion and drifts of the particles. As a result, computational time can be significantly reduced. Since many plasma phenomena in tokamaks appear on time scales much longer than the short gyromotion, in GK theory this fast gyromotion is decoupled from the motion of the gyrocenter (see figure 2.1). Even more important than the reduced dimensionality for particle-in-cell algorithm is the fact that the restrictions on the timestep can be relaxed from the plasma period to turbulent time scales. Also the restrictions on the grid spacing can be relaxed from the Debye length to the gyroradius [75]. An example therefore are low-frequency electromagnetic fluctuations responsible for the turbulent transport in tokamaks.
The small parameter assumptions, also called GK orderings, that are generally applied in GK theory are the following ([51]):

1. $\epsilon_\omega = \omega/\Omega_0 \ll 1$ (scale separation of fast particle gyrofrequency $\Omega_0 = eB_0/mc$)

2. $k_\parallel/|k_\perp| = \mathcal{O}(\epsilon_\omega) \ll 1$ (anisotropy of turbulence in $\parallel$ and $\perp$ direction)

3. $|k_\perp|\rho = \mathcal{O}(1)$ ($\perp$ spatial scales comparable to gyroradius)

In addition to this, an additional ordering on the background magnetic field is often assumed: $\epsilon_B = \rho/L_B \ll 1$ (small spatial variations of external magnetic field). With ρ the gyroradius, L_B the background magnetic field length scale, $k_\parallel$ and $|k_\perp|$ the wave vectors in parallel and perpendicular direction to the unperturbed field B_0.
For the derivation of the gyrokinetic Lagrangian the small parameter ϵ_B is required in the phase-space transformation to remove the fast gyromotion time scale (see 2.1.1).

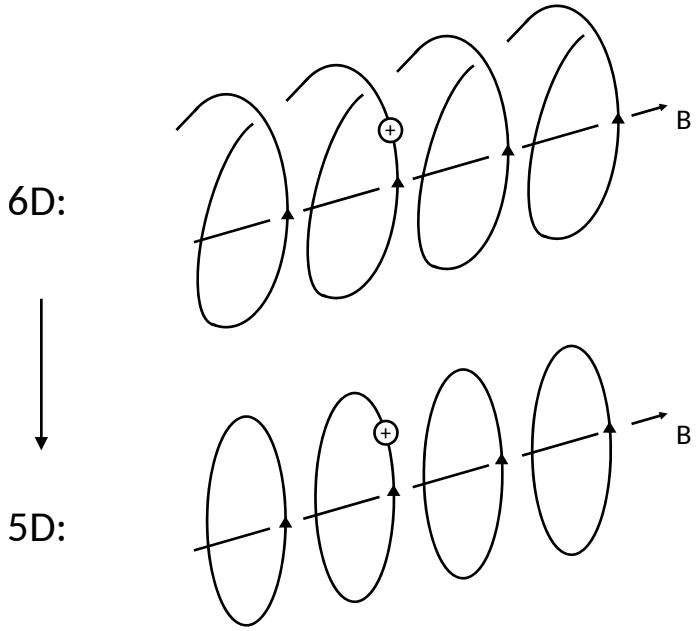


Figure 2.1: By decoupling the fast gyromotion from the gyrocenter motion, the problem can be decreased from 6D to 5D. In the picture of GK theory one can regard plasma particles as "charged rings" around a gyrocenter.

2.1.1 Derivation of gyrokinetic Lagrangian

Several ways exist to deduce the gyrokinetic equations ([69]). While in initial papers the GK equations were constructed by gyroaveraging the Vlasov equations with recursive methods (e.g., [70]), in more modern gyrokinetic theory the non-linear gyrokinetic equations are derived from a systematic Hamiltonian theory. In the following, this modern derivation of the gyrokinetic Lagrangian shall only be sketched and not performed in detail, due to the rather long equations appearing during the execution. We already applied this method for the derivation of the GK models in our previous publication (see [102]). Details on the derivation can be found in [49], [50], [103], [104] and [105]. As explained in [49], the small parameters ϵ_B and $\epsilon_\delta = (k_\perp \rho_{th}) e\phi_1 / T_\mathrm{i})$, representing small amplitudes of the fluctuating fields are used for the derivation. Where ϕ_1 is the amplitude of the fluctuating electrostatic potential, ρ_{th} the thermal Larmor radius and T_i the ion temperature. This choice comes from the assumption that the temporal variations of the fluctuating fields are small compared to the background. The idea behind gyrokinetic theory is to completely decouple the gyroangle dependency from the other variables and achieve trivial dynamics for the magnetic moment μ, i. e. $\dot{\mu} = 0$. The reduced five-dimensional phase-space dynamics of the particle are then described

by $(\mathbf{R}, v_\parallel, \mu)$. Where $\mathbf{R}$ describes the gyrocenter position and $v_\parallel$ velocity of the particle parallel to the magnetic field.
The phase-space Lagrangian of a particle in a magnetic field is the starting point for this derivation:

$$L_0 = \left(\frac{e}{c}\mathbf{A}(\mathbf{x}) + m\mathbf{v}\right) \cdot \dot{\mathbf{x}} - H(\mathbf{x}, \mathbf{v}, t) \tag{2.1}$$

with $\mathbf{x}$ and $\mathbf{v}$ the position and velocity of the charged particle. The first term (proportional to $\dot{\mathbf{x}}$) represents the symplectic part, the second one the Hamiltonian part. Via the phase-space Lagrangian perturbation, electromagnetic perturbations are introduced:

$$L_1 = \epsilon_\delta \left(\frac{e}{c}\mathbf{A_1}(\mathbf{x}, \epsilon_\delta t) \cdot \dot{\mathbf{x}} - e\phi_1(\mathbf{x}, \epsilon_\delta t\right) \tag{2.2}$$

Starting from the corresponding perturbed one-form, two major steps are executed:

1. The θ dependency of the symplectic part is moved to the Hamiltonian part, by applying a suitable velocity shift and performing a guiding center reduction. By translating the particle velocity $\mathbf{v}$ to $\bar{\mathbf{v}} = \mathbf{v} + \epsilon_\delta \frac{e}{mc}\mathbf{A_1}(\mathbf{x}, \epsilon_\delta t)$, Littlejohn's guiding center theory can directly be applied [105]. Additional to the velocity shift, the particle position needs to be adjusted by a modified Larmor radius.

2. The remaining θ dependency within the Hamiltonian part is then removed by applying a canonical Lie transform on the Hamiltonian to average with respect to the fast gyroangle. Canonical Lie transforms are near-identity canonical changes of coordinates and only act on the Hamiltonian part and thus do not modify the symplectic part (for further information see [106]).

We want to mention that in a novel approach, the near-identity coordinate transformations could also be constructed as polynomial transforms [107]. Constructing the Lagrangian in the described way, shifts the dependence on dynamical variables to the Hamiltonian or in the free field terms. Another important characteristic of the resulting Lagrangian is that self-consistent nonlinear Vlasov-Maxwell equations can be derived without any additional ordering. Thus, exact energy conservation of the resulting GK equations is satisfied ([69]). As a result, the Lie-transformed low-frequency particle Lagrangian can be achieved:

$$L_p \equiv \left(\frac{e}{c}\mathbf{A} + m_p v_\parallel \mathbf{b}\right) \cdot \dot{\mathbf{R}} + \frac{m_p c}{e_p}\mu\dot{\theta} - H_p. \tag{2.3}$$

With the velocity variables $v_\parallel$ (parallel velocity), $\mu = m_p v_\perp / (2B)$ (magnetic moment) and θ (gyroangle), the magnetic field strength B, the mass m_p and the background

vector potential **A**. Variables with a perpendicular subscript lie within the plane perpendicular to the magnetic background field **B**, whereas variables with a parallel subscript lie along the magnetic background field. For the subsequent derivations, we use the following second order electrostatic Hamiltonian, but also other choices would be possible:

$$H_p = m_p \frac{{v_\parallel}^2}{2} + \mu B + e_p J_{p,0}\phi - \frac{m_p c^2}{2B^2}|\nabla_\perp \phi|^2, \tag{2.4}$$

which can be expressed as:

$$H_p \equiv H_{p,0} + H_{p,1} + H_{p,2}, \tag{2.5}$$

$$H_{p,0} = \frac{m_p {v_\parallel}^2}{2} + \mu B, \tag{2.6}$$

$$H_{p,1} = e_p J_{p,0}\phi, \tag{2.7}$$

$$H_{p,2} = -\frac{m_p c^2}{2B^2}|\nabla_\perp \phi|^2. \tag{2.8}$$

The gyroaveraging operator $J_{p,0}$ is applied to the potential ϕ in $H_{p,1}$. $J_{p,0}$ applied to an arbitrary function ψ in configuration space is defined by:

$$(J_{p,0}\psi)(\mathbf{R}, \mu) = \frac{1}{2\pi}\int_0^{2\pi} \psi(\mathbf{R} + \boldsymbol{\rho}(\theta))\, d\theta, \tag{2.9}$$

where $\boldsymbol{\rho}$ is the vector from the guiding center position to the particle position.
The total gyrokinetic Lagrangian is then given by ([50]):

$$\begin{aligned} L = & \sum_p \int \mathrm{d}W_0 \mathrm{d}V_0 f_p(\mathbf{Z}_0, t_0) L_p(\mathbf{Z}(\mathbf{Z}_0, t_0; t), \dot{\mathbf{Z}}(\mathbf{Z}_0, t_0; t), t) \\ & + \int \mathrm{d}V \frac{E^2 - B_\perp^2}{8\pi} \end{aligned} \tag{2.10}$$

where $\mathbf{Z} \equiv (\mathbf{R}, v_\parallel, \mu, \theta)$, $\mathrm{d}W$ and $\mathrm{d}V$ stand for the volume elements in velocity and physical space. $f(\mathbf{Z}_0) = f_p$ is the distribution function for the species p at initial time

t_0. With the following definitions:

$$\begin{aligned} \mathrm{d}W &= \frac{2\pi}{m_p} B_\parallel^* \mathrm{d}v_\parallel \mathrm{d}\mu, && (2.11)\\ \mathrm{d}V &= J(X,Y,Z)\mathrm{d}X\mathrm{d}Y\mathrm{d}Z, && (2.12)\\ \mathrm{d}\Lambda &= \mathrm{d}V\mathrm{d}W, && (2.13)\\ B_\parallel^* &\equiv \mathbf{B}^* \cdot \mathbf{b} = B + \frac{cm_p}{e_p} v_\parallel \nabla \times \mathbf{b} \cdot \mathbf{b}, && (2.14)\\ \mathbf{B}^* &= \mathbf{B} + \frac{m_p c}{e_p} v_\parallel \nabla \times \mathbf{b}, && (2.15)\\ \mathbf{b} &= \mathbf{B}/B && (2.16) \end{aligned}$$

Here, m_p is the mass and e_p the charge of species p. We can further condense the total Lagrangian to:

$$L = \sum_p \int \mathrm{d}\Lambda L_p f_p + \int \mathrm{d}V \frac{E^2 - B_\perp^2}{8\pi}. \tag{2.17}$$

Now by entering eq. (2.3) into eq. (2.17), we achieve the required total gyrokinetic particle Lagrangian used in the following derivations:

$$L = \sum_p \int \mathrm{d}\Lambda \left(\left(\frac{e_p}{c} \mathbf{A} + m_p v_\parallel \mathbf{b} \right) \cdot \dot{\mathbf{R}} + \frac{m_p c}{e_p} \mu \dot{\theta} - H_p \right) f_p + \int \mathrm{d}V \frac{E^2 - B_\perp^2}{8\pi}. \tag{2.18}$$

For consistency, the GK Vlasov equation shall also be derived by the fact that the distribution function is conserved along the particle trajectories $f(\mathbf{Z}(\mathbf{Z}_0, t_0; t), t) = f(\mathbf{Z}_0, t_0)$. Taking the time derivative of this equation, we end up with the GK Vlasov equation:

$$\frac{d}{dt} f(\mathbf{Z}(\mathbf{Z}_0, t_0; t), t) = \frac{\partial}{\partial t} f(\mathbf{Z}, t) + \frac{\mathrm{d}\mathbf{Z}}{\mathrm{d}t} \cdot \frac{\partial}{\partial \mathbf{Z}} f(\mathbf{Z}, t) = 0. \tag{2.19}$$

2.1.2 Lagrangian approximations

The resulting Lagrangian (2.18) can be further approximated, according to GK theory without losing self-consistency of the final equations and energetic consistency ([50]). Two approximations are performed in this work, the so-called quasi-neutrality approximation and the linearized polarization approximation, which are widely used in GK simulations. Here, the approximation is performed according to the steps shown in [69], as already presented in our previous work [102].

Quasi-neutrality approximation

The E^2 term in the free field part of the Lagrangian is ordered small compared to the so-called $E \times B$ term, which corresponds to the second order ϕ term in the Hamiltonian. To show this relation, we introduce the Debye length squared $\lambda_{\mathrm{De}}^2 \equiv \frac{k_{\mathrm{B}} T_{\mathrm{e}}}{4\pi n_p e^2}$ and the squared ion Larmor radius $\rho_{\mathrm{s}}^2 \equiv \frac{k_{\mathrm{B}} T_{\mathrm{e}} m_{\mathrm{i}} c^2}{e^2 B^2}$. The ion Larmor radius in fusion plasmas is normally much larger than the Debye length, which delivers:

$$\frac{\rho_{\mathrm{s}}^2}{\lambda_{\mathrm{De}}^2} = \frac{4\pi n_p m_p c^2}{B^2} = \frac{c^2}{v_{\mathrm{a}}^2} \gg 1, \tag{2.20}$$

with v_{a} the Alfvén velocity. In the sum of the E^2 term in the free fields and the second order ϕ term in the Hamiltonian this relationship can now be applied:

$$\int \mathrm{d}V \frac{E^2}{8\pi} + \int \mathrm{d}\Omega f \frac{m}{2} \frac{c^2}{B^2} |\nabla_\perp \phi|^2 = \frac{1}{8\pi} \int \mathrm{d}V \left[E_\parallel^2 + \left(1 + \frac{\rho_{\mathrm{s}}^2}{\lambda_{\mathrm{De}}^2} \right) |\nabla_\perp \phi|^2 \right]. \tag{2.21}$$

In the end, the whole E^2 term can be neglected, since the $E_\parallel^2$ term is even smaller than the perpendicular part.

Linearized polarization approximation

For the second Lagrangian approximation, we assume that only the $(H_{p,0} + H_{p,1})$ part of the Hamiltonian multiplies with the full distribution function f and that $H_{p,2}$ is linearized. Therefore, we multiply $H_{p,2}$ with a time-independent equilibrium distribution function $f_{\mathrm{M},p}$:

$$\begin{aligned} L = & \sum_p \int \mathrm{d}\Lambda \left(\left(\frac{e_p}{c} \mathbf{A} + m_p v_\parallel \mathbf{b} \right) \cdot \dot{\mathbf{R}} + \frac{m_p c}{e_p} \mu \dot{\theta} - H_{p,0} - H_{p,1} \right) f_p \\ & + \sum_p \int \mathrm{d}\Lambda \frac{m_p c^2}{2B^2} |\nabla_\perp \phi|^2 f_{\mathrm{M},p} - \int \mathrm{d}V \frac{B_\perp^2}{8\pi}. \end{aligned} \tag{2.22}$$

This linearized field part is called the linearized polarization approximation. The $H_{p,2}$ term thus only affects the field equations and does not contribute to the drift motion. This linearized field part, is the only part of the model that does not use the full distribution function f. In a strict sense the model thus is also not a complete full-f model, but is also not comparable with δ-f models, where a background Maxwellian is assumed for the whole distribution and only perturbations from this Maxwellian background are modeled.

Neglecting electromagnetic perturbations

Finally, also electromagnetic perturbations are neglected, by setting $A_{\parallel} = 0$, which implies $B_{\perp}^2 = 0$ and therewith the $\int \mathrm{d}V \frac{B_{\perp}^2}{8\pi}$ term in the field part vanishes. Our final electrostatic total Lagrangian therewith becomes:

$$\begin{aligned} L &= \sum_p \int \mathrm{d}\Lambda \left(\left(\frac{e_p}{c}\mathbf{A} + m_p v_{\parallel}\mathbf{b} \right) \cdot \dot{\mathbf{R}} + \frac{m_p c}{e_p}\mu\dot{\theta} - H_{p,0} - H_{p,1} \right) f_p \\ &\quad + \sum_p \int \mathrm{d}\Lambda \frac{m_p c^2}{2B^2} |\nabla_{\perp}\phi|^2 f_{\mathrm{M},p}. \end{aligned} \tag{2.23}$$

2.1.3 Euler-Lagrange equations

For the derivation of the Euler-Lagrange equations, the variational principle is applied on the action integral, as described in [69] and [50]. Denoting:

$$I_p = \int_{t1}^{t2} L_p \mathrm{d}t \tag{2.24}$$

as the action integral (or functional). Due to Hamilton's least action principle, the motion of a mechanical system is extremal with respect to the paths of the action integral. Applying the functional derivative delivers:

$$\delta I[\mathbf{Z}, \phi] = \int_{t1}^{t2} \delta L_p[\mathbf{Z}, \phi] \mathrm{d}t = \int_{t1}^{t2} \left(\sum_{\alpha=1}^{6} \frac{\delta L_p}{\delta Z_\alpha} \cdot \delta Z_\alpha + \frac{\delta L_p}{\delta \phi} \cdot \delta\phi \right) \mathrm{d}t. \tag{2.25}$$

By taking the functional derivative of I_p with respect to $\mathbf{Z} = (\mathbf{R}, v_{\parallel}, \mu)$, the extremal behavior of the paths of the action integral results in $\frac{\delta I}{\delta \mathbf{Z}} = 0$. And since t_1 and t_2 are arbitrary, also $\frac{\delta L_p}{\delta \mathbf{Z}} = 0$ is valid.
The functional derivative of L_p with respect to a function ϕ is defined for an arbitrary variation $\delta\phi$ in the same function space as ϕ by:

$$\frac{\delta L_p}{\delta \phi} \cdot \delta\phi = \frac{d}{d\epsilon}_{|\epsilon=0} L_p(\phi + \epsilon\delta\phi) = \lim_{\epsilon \to 0} \frac{L_p(\phi + \epsilon\delta\phi) - L_p(\phi)}{\epsilon}. \tag{2.26}$$

By taking the functional derivative of L_p with respect to $\mathbf{Z}$, we can now obtain the particle equations of motion and from this the corresponding Euler-Lagrange equations for L_p can be derived. The drift motion of the gyrocenters can be described by them as:

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\delta L_p}{\delta \dot{\mathbf{Z}}} = \frac{\delta L_p}{\delta \mathbf{Z}}. \tag{2.27}$$

All required derivatives of L_p can be obtained by following [69]. For the θ derivatives one can calculate $\frac{\partial L_p}{\partial \dot{\theta}} = \mu$ and $\frac{\partial L_p}{\partial \theta} = 0$. The Euler-Lagrange equation for θ thus results in $\frac{\mathrm{d}\mu}{\mathrm{d}t} = 0$, which shows that μ is an exact invariant. Since the dependence on θ has been removed from the Lagrangian no evolution equation for θ is required.
The Euler-Lagrange equations for $\mathbf{R}$ and $v_\parallel$ yield:

$$\begin{aligned} \dot{\mathbf{R}} &= \frac{\partial H_{p,0\&1}}{\partial v_\parallel}\frac{\mathbf{B}^*}{B_\parallel^*} + \frac{c}{e_p B B_\parallel^*}\mathbf{B}\times\nabla H_{p,0\&1}, \\ \dot{v_\parallel} &= -\frac{\mathbf{B}^*}{B_\parallel^*}\cdot\nabla H_{p,0\&1}, \end{aligned} \tag{2.28}$$

with the remaining parts of the Hamiltonian in the full-f model with the linearized polarization term:

$$H_{p,0\&1} = \frac{m_p {v_\parallel}^2}{2} + \mu B + e_p J_{p,0}\phi. \tag{2.29}$$

The gradient of the remaining Hamiltonian can be calculated as:

$$\nabla H_{p,0\&1} = \mu\nabla B + e_p \nabla J_{p,0}\phi. \tag{2.30}$$

By entering (2.30) into equation (2.28) eventually gives the required Euler Lagrange equations:

$$\begin{aligned} \dot{\mathbf{R}} &= v_\parallel\frac{\mathbf{B}^*}{B_\parallel^*} + \frac{c}{e_p B B_\parallel^*}\mathbf{B}\times\left[\mu\nabla B + e_p\nabla J_{p,0}\phi\right], \\ \dot{v_\parallel} &= -\frac{\mathbf{B}^*}{B_\parallel^*}\frac{1}{m_p}\cdot\left[\mu\nabla B + e_p\nabla J_{p,0}\phi\right]. \end{aligned} \tag{2.31}$$

Without linearizing the $H_{p,2}$ part, one can also derive the complete full-f system, by using:

$$H_{p,0\&1\&2} = \frac{m_p {v_\parallel}^2}{2} + \mu B + e_p J_{p,0}\phi - \frac{m_p c^2}{2B^2}|\nabla_\perp\phi|^2 \tag{2.32}$$

and the gradient of the full Hamiltonian $H_{p,0\&1\&2}$:

$$\nabla H_{p,0\&1\&2} = \mu\nabla B + e_p\nabla J_{p,0}\phi - \frac{m_p c^2}{2B^2}\nabla|\nabla_\perp\phi|^2 + \frac{m_p c^2}{B^3}\nabla B|\nabla_\perp\phi|^2. \tag{2.33}$$

By entering the full Hamiltonian $H_{p,0\&1\&2}$ from (2.33) instead of the remaining $H_{p,0\&1}$ into equation (2.28) the full-f Euler Lagrange equations are derived:

$$\begin{aligned}\dot{\mathbf{R}} &= v_\parallel \frac{\mathbf{B}^*}{B_\parallel^*} + \frac{c}{e_p B B_\parallel^*}\mathbf{B}\times\left[\mu\nabla B + e_p\nabla J_{p,0}\phi - \frac{m_p c^2}{2B^2}\nabla|\nabla_\perp\phi|^2 + \frac{m_p c^2}{B^3}\nabla B|\nabla_\perp\phi|^2\right],\\ \dot{v_\parallel} &= -\frac{\mathbf{B}^*}{B_\parallel^*}\frac{1}{m_p}\cdot\left[\mu\nabla B + e_p\nabla J_{p,0}\phi - \frac{m_p c^2}{2B^2}\nabla|\nabla_\perp\phi|^2 + \frac{m_p c^2}{B^3}\nabla B|\nabla_\perp\phi|^2\right].\end{aligned} \tag{2.34}$$

2.1.4 Polarization equation

For the derivation of the polarization equation (or GK Poisson equation), we need to calculate the functional derivative of L with respect to the electrostatic potential ϕ and set it to 0:

$$\frac{\delta L}{\delta\phi} = 0. \tag{2.35}$$

In the first term of equation (2.23) the only ϕ-dependence is in the $H_{p,1}$ part. By using that the gyro-average operator $J_{p,0}$ is a linear operator of ϕ, we receive:

$$\frac{\delta L}{\delta\phi}\delta\phi = -\sum_p\left(\int \mathrm{d}\Lambda e_p J_{p,0}(\delta\phi)f + \int \mathrm{d}V\frac{m_p c^2}{B^2}\nabla_\perp\phi\cdot\nabla_\perp\delta\phi f_{\mathrm{M},p}\right) = 0. \tag{2.36}$$

This equation is solved in the code and is also known as the weak form of the polarization equation. The strong form can be derived by using the Hermiticity of $J_{p,0}$:

$$\int \phi J_{p,0}(f)\mathrm{d}W = \int f J_{p,0}(\phi)\mathrm{d}W, \tag{2.37}$$

and from applying the Green's formula on the second integral of (2.36). Another assumption is that ϕ vanishes at the boundary and $B_\parallel^*$ is taken out of $\mathrm{d}W = \frac{2\pi}{m_p}B_\parallel^*\mathrm{d}v_\parallel\mathrm{d}\mu$. Therewith eq. (2.36) can be rewritten as:

$$-\sum_p\int \mathrm{d}V\delta\phi\int \mathrm{d}W\left(e_p J_{p,0}f + \frac{1}{B_\parallel^*}\nabla_\perp\left(\frac{m_p c^2}{B^2}B_\parallel^* f_{\mathrm{M},p}\nabla_\perp\phi\right)\right) = 0. \tag{2.38}$$

Since the choice of $\delta\phi$ is arbitrary, also

$$\sum_p\int \mathrm{d}W\left(e_p J_{p,0}f + \frac{1}{B_\parallel^*}\nabla_\perp\left(\frac{mc^2}{B^2}B_\parallel^* f_{\mathrm{M},p}\nabla_\perp\phi\right)\right) = 0 \tag{2.39}$$

has to be fulfilled.
The velocity integral over the Maxwellian distribution can now be performed, by noting that in the second term the spatial gradient and the integral with respect to dW commute, to achieve:

$$-\sum_p \nabla_\perp \frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \phi = \sum_p \int \mathrm{d}W e_p J_{p,0} f. \tag{2.40}$$

Here, $n_{p,0}$ stands for the density of the equilibrium Maxwellian f_M:

$$n_{p,0} = \int \mathrm{d}W f_{\mathrm{M},p}. \tag{2.41}$$

Note that for electrons $J_{p,0} = 1$, which means that due to their small Larmor radius we neglected electron FLR effects and also the polarization is neglected for the electrons. The resulting linear equation has the form of a quasi-neutrality condition, $\sum_p e n_p = 0$, with the particle charge density $e n_p$.
In the full-f version without linearization, the polarization equation looks very similar:

$$-\sum_p \nabla_\perp \frac{n_{\mathrm{i},0} m_p c^2}{B^2} \nabla_\perp \phi = \sum_p \int \mathrm{d}W e_p J_{p,0} f, \tag{2.42}$$

but with a different definition for the density $n_{\mathrm{i},0}$:

$$n_{\mathrm{i},0} = \int \mathrm{d}W f_{\mathrm{i},0}. \tag{2.43}$$

2.1.5 Summary

In the following, the relevant equations for the GK model implemented in PICLS shall be summarized again for a better visibility.

Summary for linearized full-f

Starting with the Euler-Lagrange equations of the full-f model with a linearized polarization term:

$$\begin{aligned}
\dot{\mathbf{R}} &= v_\parallel \frac{\mathbf{B}^*}{B_\parallel^*} + \frac{c}{e_p B B_\parallel^*} \mathbf{B} \times \left[\mu \nabla B + e_p \nabla J_{p,0} \phi \right], \\
\dot{v_\parallel} &= -\frac{\mathbf{B}^*}{B_\parallel^*} \frac{1}{m_p} \cdot \left[\mu \nabla B + e_p \nabla J_{p,0} \phi \right], \\
\dot{\mu} &= 0.
\end{aligned} \tag{2.44}$$

The polarization equation shall be displayed again:

$$-\sum_p \nabla_\perp \frac{n_{\mathrm{i},0} m_p c^2}{B^2} \nabla_\perp \phi = \sum_p \int \mathrm{d}W e_p J_{p,0} f. \tag{2.45}$$

With $n_{\mathrm{i},0} = \sum_i n_\mathrm{i}$ being the sum over the background density of all ion species. For completeness of the set of equations, the Vlasov equation is shown, as well:

$$\frac{d}{dt} f = \frac{\partial f}{\partial t} + \dot{\mathbf{R}} \nabla f + \dot{\mathbf{v}}_\parallel \frac{\partial f}{\partial v_\parallel}. \tag{2.46}$$

Despite the approximations we assumed during the derivation, this set of equations is still able to correctly describe a huge class of micro-instabilities that are excited by temperature and density gradients. Examples for these instabilities would be ion temperature gradient (ITG) modes and trapped electron modes (TEM) (see [69]).

Summary for full-f without linearization

Also in the full-f case without linearized polarization approximation, the relevant equations shall be shown again. First the Euler-Lagrange equations:

$$\begin{aligned}
\dot{\mathbf{R}} &= v_\parallel \frac{\mathbf{B}^*}{B_\parallel^*} + \frac{c}{e_p B B_\parallel^*} \mathbf{B} \times \left[\mu \nabla B + e_p \nabla J_{p,0} \phi - \frac{m_p c^2}{2B^2} \nabla |\nabla_\perp \phi|^2 + \frac{m_p c^2}{B^3} \nabla B |\nabla_\perp \phi|^2 \right], \\
\dot{v_\parallel} &= -\frac{\mathbf{B}^*}{B_\parallel^*} \frac{1}{m_p} \cdot \left[\mu \nabla B + e_p \nabla J_{p,0} \phi - \frac{m_p c^2}{2B^2} \nabla |\nabla_\perp \phi|^2 + \frac{m_p c^2}{B^3} \nabla B |\nabla_\perp \phi|^2 \right], \\
\dot{\mu} &= 0.
\end{aligned} \tag{2.47}$$

And the non-linearized polarization equation:

$$-\sum_p \nabla_\perp \frac{n_\mathrm{i} m_p c^2}{B^2} \nabla_\perp \phi = \sum_p \int \mathrm{d}W e_p J_{p,0} f. \tag{2.48}$$

With $n_{\mathrm{i},0} = \sum_i n_\mathrm{i}$ being the sum over the real density of all ion species. The Vlasov equation is the same as in the delta-f case. A non-linear Polarization equation is not yet included in PICLS, but an implementation in the near future is planned. For this, Alexey Mishchenko's iterative method (see [108]) could help as a guideline.

2.2 Equations in slab geometry

Within a large part of this thesis we are solely using slab geometry for our simulation domains. In a later stage, we also apply simple cylindrical, or circular tokamak geometries. Other more complex coordinate systems, like field-aligned coordinates (see [109]), that would be required for more advanced geometries (e.g., a realistic Tokamak geometry) were not applied, but are discussed for future development. In slab geometry, however, in 1D (one spatial dimension) the respective domain is represented by a straight line and in 3D by a square or flux tube.

2.2.1 Reduction of B-field equations

Resulting from the slab geometry, many B-field related terms can be further simplified and used in our set of equations. Here, a general overview of relevant terms is listed:

$$
\begin{aligned}
J(X,Y,Z) &= 1, && (2.49)\\
\mathbf{B} &= B(x,y,z)\mathbf{e}_z, && (2.50)\\
\nabla\times\mathbf{B} &= \frac{\partial B}{\partial y}\mathbf{e}_x - \frac{\partial B}{\partial x}\mathbf{e}_y, && (2.51)\\
\nabla B &= \frac{\partial B}{\partial x}\mathbf{e}_x + \frac{\partial B}{\partial y}\mathbf{e}_y + \frac{\partial B}{\partial z}\mathbf{e}_z, && (2.52)\\
\mathbf{b} &= \mathbf{e}_z, && (2.53)\\
\mathbf{b}\times\nabla B &= -\frac{\partial B}{\partial y}\mathbf{e}_x - \frac{\partial B}{\partial x}\mathbf{e}_y, && (2.54)\\
\mathbf{b}\cdot\nabla B &= \frac{\partial B}{\partial z}, && (2.55)\\
\nabla\times\mathbf{b} &= \frac{1}{B}\left(\nabla\times\mathbf{B} + \mathbf{b}\times\nabla B\right) && (2.56)\\
&= \frac{1}{B}\left(\frac{\partial B}{\partial y} - \frac{\partial B}{\partial y}\right)\mathbf{e}_x - \frac{1}{B}\left(\frac{\partial B}{\partial x} - \frac{\partial B}{\partial x}\right)\mathbf{e}_y = 0, && (2.57)\\
\mathbf{B}^* &= \mathbf{B}, && (2.58)\\
B_\parallel^* &\equiv \mathbf{B}^*\cdot\mathbf{b} = B. && (2.59)
\end{aligned}
$$

2.2.2 Equations of motion

Using the simplified B-field equations of the previous section 2.2.1 we can rewrite the equations of motion (2.31) in the slab case, that are implemented in PICLS for slab

geometry simulations:

$$\begin{aligned}\dot{\mathbf{R}} &= v_{\parallel}\mathbf{b} + \frac{c}{e_p}\mu\frac{\mathbf{B}\times\nabla B}{B^2} + c\frac{\mathbf{B}\times\nabla J_{p,0}\phi}{B^2},\\ \dot{v_{\parallel}} &= -\frac{\mu}{m_p}\mathbf{b}\cdot\nabla B - \frac{e_p}{m_p}\mathbf{b}\cdot\nabla J_{p,0}\phi. \end{aligned} \tag{2.60}$$

2.3 System of units and normalization

2.3.1 System of units

The system of units we chose is the so-called centimetre–gram–second system (or CGS) and is based on the idea to develop a system of units based on three basic units length mass and time. This system was first introduced by the British Association for the Advancement of Science in 1873 ([110]). For electromagnetic studies this system is particularly interesting, since both, the E- and the B-field have the same units in CGS. In the following, some of the relevant units used throughout this thesis are shown:

$$e = [\mathrm{g}]^{1/2}[\mathrm{cm}]^{3/2}[\mathrm{s}]^{-1}, \tag{2.61}$$
$$\frac{e}{c} = [\mathrm{g}]^{1/2}[\mathrm{cm}]^{1/2}, \tag{2.62}$$
$$B = [\mathrm{g}]^{1/2}[\mathrm{cm}]^{-1/2}[\mathrm{s}]^{-1}, \tag{2.63}$$
$$E = [\mathrm{g}]^{1/2}[\mathrm{cm}]^{-1/2}[\mathrm{s}]^{-1}, \tag{2.64}$$
$$\mu = [\mathrm{g}]^{1/2}[\mathrm{cm}]^{5/2}[\mathrm{s}]^{-1}, \tag{2.65}$$
$$k_B T = [\mathrm{g}]^{1}[\mathrm{cm}]^{2}[\mathrm{s}]^{-2}, \tag{2.66}$$
$$\nabla_{\perp} = [\mathrm{cm}]^{-1}. \tag{2.67}$$

2.3.2 Normalized variables

In PICLS we implemented two different sets of normalization schemes for verification of the code normalization. In the first set, the code uses the normal CGS units of the physical quantities. In the second one, the equations are normalized before numerical solutions are calculated. In this section we want to discuss this normalized system of

equations implemented in PICLS. The full set of normalization equations is:

$$\begin{gathered} t = \tau\bar{t}, \quad R = L\bar{x}, \quad v_\parallel = V\bar{v}_\parallel, k_\perp = \bar{k_\perp}/\rho_{ce}, \\ e_p = e\bar{Z}_p, \quad m_p = M\bar{m}_p, \quad T_p = k_B T\bar{T}_p, \\ E \equiv \nabla J_{p,0}\phi = E_{\text{norm}}\bar{E} = \frac{k_\text{B}T}{eL}\bar{E}, \\ B = B_{\text{norm}}\bar{B} = \frac{1}{V}\frac{k_\text{B}T}{eL}\bar{B}, \quad \mu = \mu_{\text{norm}}\bar{\mu} = \frac{eLV^3M}{ck_\text{B}T}\bar{\mu}, \end{gathered} \tag{2.68}$$

with the electron charge $-e$, the electron mass $M = m_\text{e}$ and the initial thermal electron velocity $V = \sqrt{k_\text{B}T/m_\text{e}}$. B and T are initialized by its input values B_0 and T_0 in the code. The spatial scale L is set to the electron gyroradius:

$$\rho_{ce} = \sqrt{m_\text{e}k_\text{B}T_\text{e}}\frac{c}{eB_0}. \tag{2.69}$$

Subsequently, τ becomes the inverse electron cyclotron frequency $w_{ce}^{-1} = \frac{m_\text{e}c}{eB_0}$ and $k_\perp$ is normalized to the inverse of the electron cyclotron radius $1/\rho_{ce}$.
Our primary reason for the choice of the normalization scheme is to easily perform a-dimensional scans in the future (e.g., on ρ^* or ν^*). Nevertheless, other normalization schemes (or no normalization, such as SI or CGS) are also justifiable. For our numerical calculations this choice is not decisive, since double precision numbers can easily handle variations of magnitudes.

2.3.3 Normalized set of equations

By taking the normalization of section 2.3.2 and entering it into the equations of motion in slab geometry from section 2.2.1, the normalized equations of motion implemented in the code can be derived:

$$\begin{aligned} \frac{\partial\bar{x}}{\partial\bar{t}} &= c(-\frac{1}{eZ_p}\bar{\mu}\mu_{\text{norm}}\frac{\frac{\partial\bar{B}}{\partial\bar{y}}\frac{B_{\text{norm}}}{L}}{\bar{B}B_{\text{norm}}} - \frac{\bar{E}_y E_{\text{norm}}}{\bar{B}B_{\text{norm}}})\frac{\tau}{L}, \\ \frac{\partial\bar{y}}{\partial\bar{t}} &= c(-\frac{1}{eZ_p}\bar{\mu}\mu_{\text{norm}}\frac{\frac{\partial\bar{B}}{\partial\bar{x}}\frac{B_{\text{norm}}}{L}}{\bar{B}B_{\text{norm}}} + \frac{\bar{E}_x E_{\text{norm}}}{\bar{B}B_{\text{norm}}})\frac{\tau}{L}, \\ \frac{\partial\bar{z}}{\partial\bar{t}} &= \bar{v}_\parallel, \\ \frac{\partial\bar{v}_\parallel}{\partial\bar{t}} &= (-\frac{\bar{\mu}\mu_{\text{norm}}}{M\bar{m}_p}\frac{\partial\bar{B}}{\partial\bar{z}}\frac{B_{\text{norm}}}{L} - \frac{eZ_p}{M\bar{m}_p}\bar{E}_z E_{\text{norm}})\frac{\tau}{V}. \end{aligned} \tag{2.70}$$

Entering the normalization factors into the polarization equation (2.40) yields:

$$-\sum_p \nabla_\perp \frac{n_0 M \bar{m}_p}{\bar{B}^2 B_{\mathrm{norm}}^2} \nabla_\perp \phi = \sum_p \int \mathrm{d}W e \bar{Z}_p J_{p,0} f. \tag{2.71}$$

2.4 Total energy for the electrostatic system

Simply by deriving the equations from the not directly time dependent Lagrangian density, the energy is already conserved, because adjusting the Lagrangian on the level of the Hamiltonian conserves energy. Contrary to this energy conservation can be violated, if terms are dropped at a later stage. The total conserved energy in our electrostatic model is [111]:

$$\mathcal{E}_{\mathrm{tot}} = \sum_p \int \mathrm{d}W \mathrm{d}V H_p f_p = \mathcal{E}_{\mathrm{k}} + \mathcal{E}_{\mathrm{f}}. \tag{2.72}$$

Here, the particle energy (or total kinetic energy) $\mathcal{E}_{\mathrm{k}}$ has the same definition for full-f with and without linearization in the polarization term:

$$\mathcal{E}_{\mathrm{k}} = \sum_p \int \mathrm{d}W \mathrm{d}V H_{p,0} f_p. \tag{2.73}$$

The definition for the field energy in the case with linearization is:

$$\mathcal{E}_{\mathrm{f}} = \sum_p \int \mathrm{d}W \mathrm{d}V H_{p,1} f_p + \sum_p \int \mathrm{d}W \mathrm{d}V H_{p,2} f_{\mathrm{M},p} \tag{2.74}$$

$$= \sum_p \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} \phi f_p - \sum_p \int \mathrm{d}W \mathrm{d}V \frac{m_p c^2}{2B^2} |\nabla_\perp \phi|^2 f_{\mathrm{M},p}. \tag{2.75}$$

In the next step, we consider the polarization equation (2.40), multiply it times ϕ and integrate over space:

$$-\sum_p \int \mathrm{d}V \mathrm{d}W \left(\nabla_\perp \frac{f_{p,0} m_p c^2}{B^2} \nabla_\perp \phi \right) \phi = \sum_p \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} f \phi. \tag{2.76}$$

Now we integrate by parts, and use the Hermiticity of $J_{p,0}$ and divide by 2, to obtain:

$$\sum_p \int \mathrm{d}V \mathrm{d}W \frac{f_{p,0} m_p c^2}{2B^2} |\nabla_\perp \phi|^2 = \sum_p \frac{1}{2} \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} \phi f_p. \tag{2.77}$$

Inserting this result in the second term of equation (2.75) gives:

$$\mathcal{E}_\mathrm{f} = \sum_p \frac{1}{2} \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} \phi f_p. \tag{2.78}$$

For the field energy in the case without linearization, the procedure is equal:

$$\mathcal{E}_\mathrm{f} = \sum_p \int \mathrm{d}W \mathrm{d}V H_{p,1} f_p + \sum_p \int \mathrm{d}W \mathrm{d}V H_{p,2} f_p \tag{2.79}$$

$$= \sum_p \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} \phi f_p - \sum_p \int \mathrm{d}W \mathrm{d}V \frac{m_p c^2}{2B^2} |\nabla_\perp \phi|^2 f_p. \tag{2.80}$$

This time, we use the polarization equation of the full-f case without linearization (2.42), multiply it again times ϕ and integrate over space:

$$-\sum_p \int \mathrm{d}V \mathrm{d}W \left(\nabla_\perp \frac{f_p m_p c^2}{B^2} \nabla_\perp \phi \right) \phi = \sum_p \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} f_p \phi, \tag{2.81}$$

Again, integrating this expression by parts, using the Hermiticity of $J_{p,0}$ and dividing by 2 gives:

$$\sum_p \int \mathrm{d}V \mathrm{d}W \frac{f_p m_p c^2}{2B^2} |\nabla_\perp \phi|^2 = \sum_p \frac{1}{2} \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} \phi f_p. \tag{2.82}$$

In the end, the same result as in the linearized case (see eq. (2.78)) is obtained:

$$\mathcal{E}_\mathrm{f} = \sum_p \frac{1}{2} \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} \phi f_p \tag{2.83}$$

and the total conserved energy of the system in both cases can be written as:

$$\mathcal{E}_\mathrm{tot} = \mathcal{E}_\mathrm{k} + \mathcal{E}_\mathrm{f} = \sum_p \int \mathrm{d}W \mathrm{d}V H_{p,0} f_p + \sum_p \frac{1}{2} \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} \phi f_p, \tag{2.84}$$

with $H_{p,0} = \frac{1}{2} m_p {v_\parallel}^2 + \mu B$ in the kinetic part.

3 Sheath model

3.1 Physical Debye sheath

Plasma within fusion devices, such as tokamaks or stellarators, is mostly confined within a volume of nested magnetic surfaces that are closed in themselves. Thus, plasma motion along the field lines in adequately constructed devices does not lead to plasma loss. However, collisions and drifts cause plasma to diffuse into a loss region and from there the particles are transported towards the device wall - e.g., a divertor or limiter - via open magnetic field lines. In their work Chodura [112] and Stangeby [21] describe that on this path towards the plasma wall, the plasma particles have to pass two regions, the presheath and the sheath, that both add up to the scrape-off layer:

1. *Pre-sheath*: The pre-sheath is closer to the plasma core and the by far larger region. Compared to the sheath its gradients are rather weak and generally the flow is subsonic.

2. *Sheath*: This is a very narrow region directly attached to the wall with large gradients (e.g., for temperature and density) and supersonic velocity of the flow.

The models and methods developed and tested within this thesis are dedicated to open field line simulations and thus specifically to sheath models. Subsequently, we want to focus on sheath models for the rest of this chapter and refer the reader to [112], [21] and [113] for further details on these regions.
Physically, the sheath that builds up in front of the mostly grounded device wall is a region of net positive charge and thus also called electrostatic Debye sheath. Due to a potential drop within the sheath (see figure 3.1), ions are accelerated towards the wall, whereas the faster electrons are to a large extent reflected back towards the plasma. Gradients of electric field and flow variables are comparably high in the sheath. The reason therefore is that the plasma flow in the sheath is accelerated, but in general at very small length scales, i.e. the Debye length and the ion gyro radius. The collisionality within the sheath can generally be ignored.
To show the discrepancy of the scales present in the Debye sheath and the scales, we want to list some values for the LAPD experiment (see [52]):

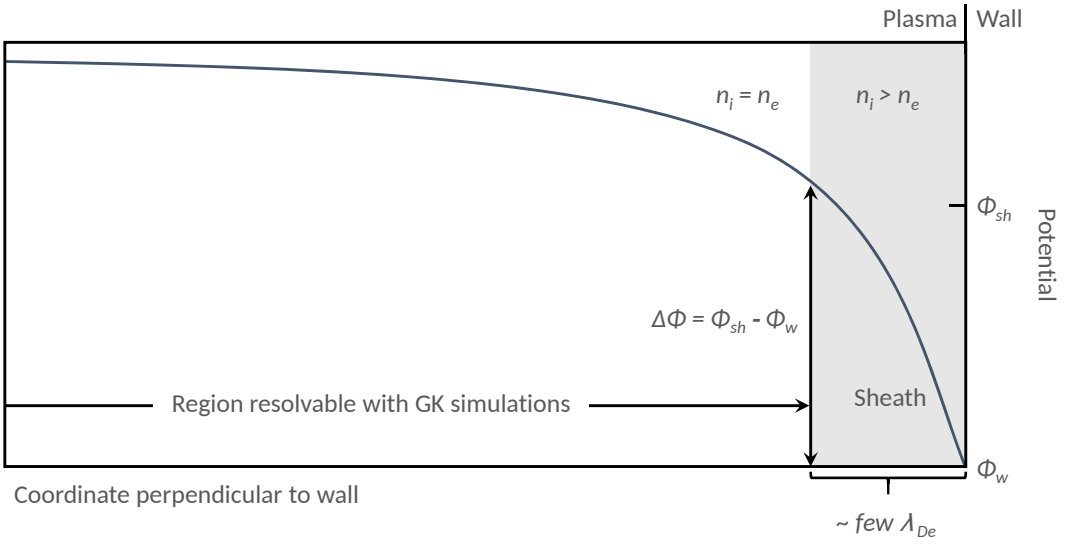


Figure 3.1: Schematic sketch of the potential close to the plasma-wall interface. The sheath is represented by the small grey region and is only of the size of some electron Debye lengths. Here, we have an excess of protons and thus a net positive charge. These small length scales and net positive charge cannot be simulated via gyrokinetic theory, which is constructed to comply with quasi-neutrality condition. Thus, a different set of boundary conditions is required to model its effects.

- Length: By comparing the Debye length ($\lambda_{\mathrm{De}} \approx 10^{-6}\mathrm{m}$) with the ion gyro radius ($\sim 10^{-2}\mathrm{m}$) and the much larger turbulence scale ($\sim 10\mathrm{m}$), shows already the huge scaling difference.

- Frequency: This picture is reassured, when comparing the very large electron plasma frequency ($\omega_{pe} \approx 10^{9}\mathrm{s}^{-1}$) to the ion gyro frequency ($\approx 10^{6}\mathrm{s}^{-1}$) and the even smaller turbulence frequencies of interest ($\omega_{*} \approx 10^{4}\mathrm{s}^{-1}$)

Apparently, the scales present within the Debye sheath are quite different from the ones that can be treated via gyrokinetic models. Additionally, increasing spatial and temporal resolution solely for being able to simulate the sheath is computationally not viable. But for the simulation of open field line plasmas, covering heat and particle fluxes towards the wall is significant. To resolve this dilemma, we need an appropriate sheath model, which is able to model the effects of a sheath, without actually resolving it. The so-called logical sheath boundary conditions are a suitable model for this task, and will be shown in sections 3.2 and 3.3.

3.2 Logical (or insulating) sheath

We generally base the setup of the logical sheath model used in this thesis on Parker's model, published in 1993 ([114]), and have already published our implementation previously (see [102]). He originally developed his model for fully kinetic 1D2V PIC simulations, where he tried to find a model that essentially resembles sheath effects for sheath simulations with coarse spatial grids. In recent works, also Shi [115] and Pan [94] implemented Parker's logical sheath boundary conditions for parallel heat flux studies in gyrokinetic 1D1V Vlasov-simulations. We will also take this boundary condition as a method of choice for our 1D1V simulations in chapter 6, which were already described in [102]. A variant of this model is also used in the XGC gyrokinetic PIC code [116]. In this model, the total parallel current to the wall is set to zero ($j_{\parallel} = 0$) at every point in time and thus the wall can be regarded as insulating. Subsequently, these boundary conditions are called insulating sheath boundary conditions, as well.
From a physical point of view, incident ions that flow towards the sheath boundary are accelerated by the dropping sheath potential. Whereas electrons are only absorbed if their velocity is high enough to overcome the sheath potential drop at the wall. Electrons below this critical velocity are reflected back into the domain. Within the logical sheath model $j_{\parallel} = 0$ is set, which means that only the fastest electrons can overcome the sheath potential drop to exactly balance the number of ions that hit the wall. In case of the ions, however, all of them can freely exit the system. The velocity of the slowest electron still exiting the domain is defined as the electron cut-off velocity v_{ce}. The sheath potential then can be calculated from v_{ce} according to:

$$\delta\phi = \phi_{\mathrm{sh}} - \phi_{\mathrm{w}} = \frac{m}{2e} v_{\mathrm{ce}}^2, \tag{3.1}$$

where ϕ_{sh} is the sheath potential and ϕ_{w} the wall potential. This can be seen as a negative step jump from the sheath potential to the wall as correctly approximated by $\lambda_{\mathrm{De}} \to 0$. For a grounded wall with $\phi_w = 0$ this resembles exactly the sheath potential. With z as spatial coordinate, we can write the $j_{\parallel} = 0$ condition at the sheath position z_{sh} as:

$$\Sigma_{\mathrm{i}} q_{\mathrm{i}} \int_0^{\infty} f_{\mathrm{i}}(z_{\mathrm{sh}}, v_{\parallel}, t) v_{\parallel} \mathrm{d}v_{\parallel} = e \int_{v_{\mathrm{ce}}}^{\infty} f_{\mathrm{e}}(z_{\mathrm{sh}}, v_{\parallel}, t) v_{\parallel} \mathrm{d}v_{\parallel}. \tag{3.2}$$

The total absorption of ions and the partial absorption of electrons at z_{sh}, leads to the following distribution functions for electrons and ions at the wall:

$$f_{\mathrm{i}}(z_{\mathrm{sh}}, -v_{\parallel}, t) = 0 \quad v_{\parallel} > 0, \tag{3.3}$$

$$f_{\mathrm{e}}(z_{\mathrm{sh}}, -v_{\parallel}, t) = \begin{cases} 0 & v_{\parallel} > v_{\mathrm{ce}} \\ f_{\mathrm{e}}(z_{\mathrm{sh}}, v_{\parallel}, t) & v_{\mathrm{ce}} > v_{\parallel} > 0. \end{cases} \tag{3.4}$$

Here, $-v_{\parallel}$ stands for velocities of particles moving away from the wall, back into the system. We want to emphasize, that the formulation of the boundary condition on the ions in equation (3.3) would not be sufficient to treat the cold ion case. In our case, no force accelerates the ions towards the wall. Eq. (3.3) also does not imply a force on the ions from the potential drop across the logical sheath and thus the Bohm criterion will not be satisfied. Implementing these effects in the equations could be considered for future work.
The $j_{\parallel} = 0$ condition in the described 1D1V model is valid on each field line. However, in a three-dimensional model a current flow out of the wall at one point and an inflow at a different point is permitted. Nevertheless, the total integrated in- and outflow of the wall still has to cancel and $j_{\parallel} = 0$ must hold for the total parallel current into the walls.

3.2.1 Algorithm

Numerically, the logical sheath model in our code is implemented according to the following algorithm (based on [114]), as described in our previous work [102]:

1) Advance particle trajectories

2) At each time, step count number of electrons n_{e} and ions n_{i} that hit the wall

3) Compare n_{e} and n_{i}
   a) If $n_{\mathrm{i}} \leq n_{\mathrm{e}}$ (probable condition)
      - Order all n_{e} electrons according to velocity
      - Let fastest n_{i} electrons and n_{i} ions leave the domain
      - Reflect remaining $n_{\mathrm{e}} - n_{\mathrm{i}}$ slow electrons
   b) If $n_{\mathrm{e}} \leq n_{\mathrm{i}}$ (very improbable condition)
      - Order all n_{i} ions according to velocity
      - Let fastest n_{e} ions and n_{e} electrons leave the domain

- Reflect slowest $n_\mathrm{i} - n_\mathrm{e}$ ions
- (Alternative: Absorb all ions and maintain net positive wall charge for next time step)

In figure 3.2 a graphical representation of the algorithm's functionality is displayed. Following this algorithm, the electron cut-off velocity v_ce can be computed and based on

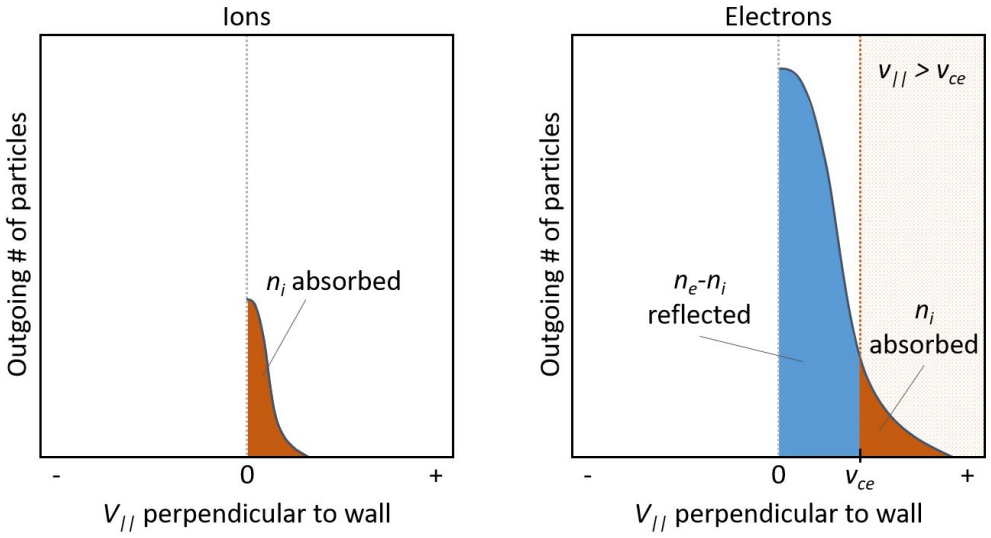


Figure 3.2: Schematic representation of number of outgoing ions and electrons at the sheath position as a function of $v_\parallel$ (with different scales). As described in the algorithm in section 3.2.1, all n_i ions are absorbed by the wall, whereas only the fastest n_i electrons are absorbed. The $n_\mathrm{e} - n_\mathrm{i}$ slower electrons are reflected back into the plasma.

v_ce, we can calculate the sheath potential according to eq. (3.1). The very improbable $n_\mathrm{e} \leq n_\mathrm{i}$ case described by the b) part is rather for completeness and appears only in very rare occasions. Nevertheless, by writing the algorithm in the current form, the $j_\parallel = 0$ condition is fully met.
The alternative approach to store the excess ions as net positive wall charge would contribute to the potential calculation of the next time step. But with assuming a grounded wall the charge hitting the wall for the next time step would be neglected.
In general, we implemented two different versions of this insulating sheath model. The first option is to apply the $j_\parallel = 0$ criterion to each cell in the x/y-plane at the sheath position. Thus, for each cell a cut-off velocity v_ce can be calculated. The second option is to see the entire x/y-plane as one cell and apply $j_\parallel = 0$ for the entire plane. In this case, only one single v_ce is calculated for all particles.
Alternative sheath models to the logical/insulating sheath that allow a net current

flowing into the wall $j_{\parallel} \neq 0$ are so-called conducting sheath models. These will be introduced in section 3.3.

3.3 Conducting sheath

The main difference compared to the insulating sheath model is that the total current towards the wall is not bound to 0, thus $j_{\parallel} \neq 0$ is allowed. In other words, the number of ions that are absorbed by the wall per time step does not have to equalize the number of electrons as in the insulating sheath case.
In this model, no boundary conditions are applied on the ions. This means that all ions that hit the boundary are absorbed and none of them are reflected back into the plasma. However, the absorption/reflection criterion for the electrons is again depending on the electron cut-off velocity from equation (3.1). But this time v_{ce} is calculated from the sheath potential ϕ_{sh}, according to:

$$v_{\mathrm{ce}} = \sqrt{\frac{2e}{m}\phi_{\mathrm{sh}}}, \tag{3.5}$$

with the wall taken to be just outside the simulation domain and potential set to zero ($\phi_{\mathrm{w}} = 0$). The sheath potential is calculated at the sheath position ($z = 0$ or $z = L_z$) from the previously derived polarization equation (or GK Poisson equation) (2.42). This procedure is valid in the case that more electrons than ions are hitting the wall, which is the vast majority of cases. In the more unusual case that more ions than electrons are hitting the wall, all electrons would be absorbed and the sheath condition would apply to the ions.
Sheath currents are able to self-consistently fluctuate in interaction with turbulence in and out of the wall and are only reflected by the sheath potential that builds up. Starting for example from an initial condition where electrons and ions are uniformly distributed in space, electrons will quickly leave the plasma. This causes a net positive charge in the plasma, which rapidly leads to an increase in the sheath potential and eventually reflects electrons back into the plasma. Apparently, fluid codes employ a similar procedure. These codes calculate the potential from the fluid vorticity equation everywhere in the domain (including ϕ_{sh}) and then set the boundary condition on the parallel electron velocity with the help of the calculated ϕ_{sh} [117, 118].
Within PICLS for the conducting sheath we use two different approaches to calculate the potential and thus also the cut-off velocity per hitting electron (see also figure 3.3):

1. **Calculate ϕ_{sh} per sheath cell:** For each simulation cell that is located at the sheath boundary the sheath potential is calculate at the center of the x/y-plane

directly at the sheath position ($z = 0$ or $z = L_z$). This is possible, due to the usage of bspline Finite Elements. Thus, for all particles that hit the sheath within this x/y-plane the same cut-off velocity is calculated. In this case, $n_x \times n_y$ potential values have to be calculated for each domain boundary.

2. **Calculate $\phi_{\mathbf{sh}}$ per electron position:** Depending on the (x,y,z) coordinates where the particle hits the sheath (with $z = 0$ or $z = L_z$), $\phi_{\text{sh}}(x, y, z)$ is calculated for each electron, individually. The velocity of each electron is then compared with its individual v_{ce}. This approach is more precise than the previously mentioned, but depending on the number of hitting electrons can require much more computation time.

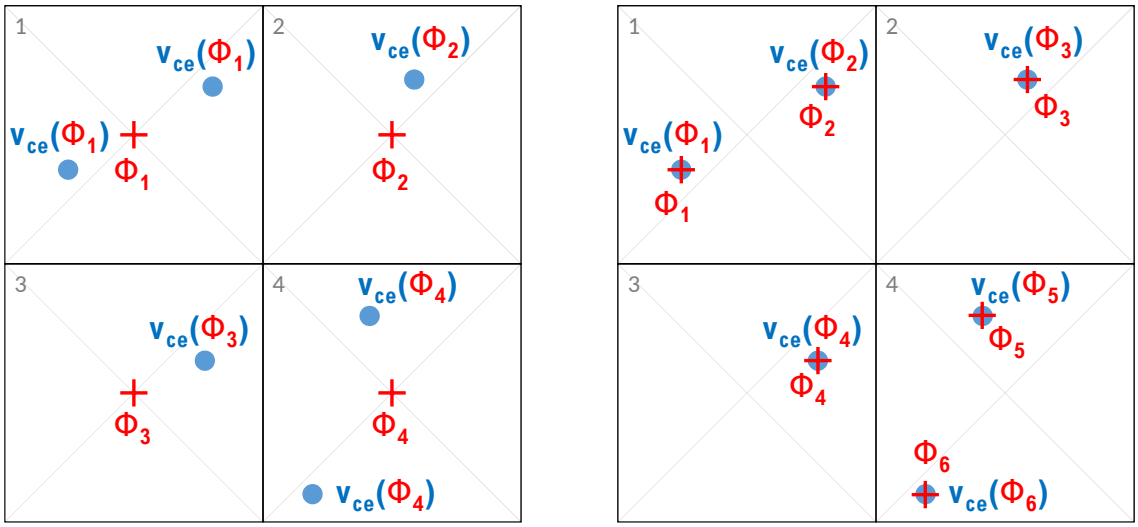


Figure 3.3: Illustration of the two different approaches for the ϕ_{sh} calculation for a conducting sheath (see section 3.3). The blue dots indicate the positions of hitting electrons in the (x, y) plane directly at the sheath (with $z = 0$ or $z = L_z$). The red crosses indicate the (x, y) position, where the potential for the cut-off velocities of the electrons is calculated. In a) ϕ_{sh} is calculated at the center of the x/y-plane and valid for each hitting electron in this cell. In b) however, ϕ_{sh} is calculated for each electron, individually, depending on the (x, y, z) coordinates where the particle hits the sheath (with $z = 0$ or $z = L_z$).

For both cases holds, that all electrons that exceed the calculated cut-off velocity are absorbed by the wall, the rest are reflected back into the domain. The calculation of the sheath potential on the specific sheath position is done via spline interpolation. No

further boundary conditions (e.g., on the current) are taken into account in the implemented conducting sheath models. It is important to mention that so far conducting sheath boundary conditions are only implemented for cases where the B-field strikes the domain wall perpendicularly. For arbitrary angles the calculation of the sheath potential then requires additional boundary conditions. The main complication here is the calculation of the potential at the sheath position. However, for the adaptation of the particle trajectories at the boundary, already a suitable approach was derived and will be presented in the following section.

3.4 Non-perpendicular B-field on sheath

In case a marker is hitting the sheath boundary of the domain, it can either be absorbed by the wall and thus exit the domain, or it can be reflected back into the simulation domain. In this section we want to focus on how different angles of B-fields onto the boundary surface affect the reflection / absorption behavior of markers by the wall. The simplest case one can imagine is a B-field line that strikes the sheath in a 90° angle (see fig. 3.4 a)). In this case, a marker that follows this field line and hits the sheath can be reflected back into the domain by simply reversing its parallel velocity $v_{\parallel} \rightarrow -v_{\parallel}$ and reflecting it back along the incident field line in z-direction. The x- and y-coordinate remain untouched. This approach can be chosen in the drift-kinetic, as well as the gyrokinetic model. Once the field line strikes the sheath surfaces with an angle $\alpha \neq 90°$,

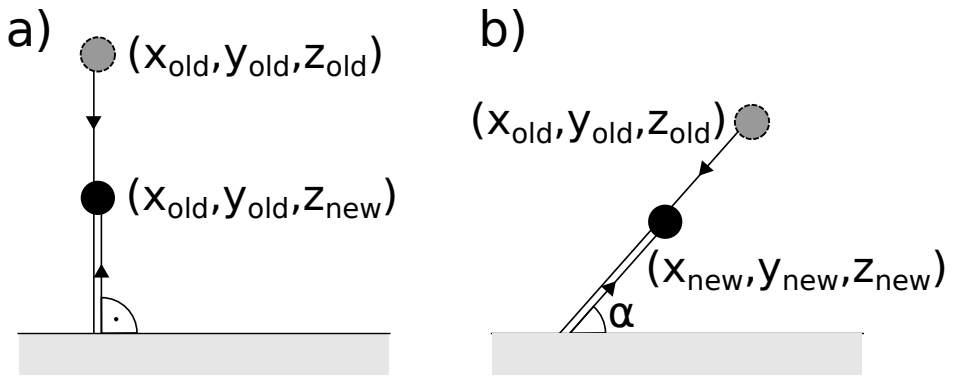


Figure 3.4: Illustration of particle trajectories along a B-field line (black line) that are reflected from the boundary back into the plasma domain. In a) the B-field line strikes the surface perpendicularly and thus only the z-coordinate needs to be updated. In b) the angle α between the surface and the direction of $\mathbf{B}$ is $\neq 90°$. Hence, all spatial coordinates need to be updated.

the reflection process becomes more complicated. In a drift-kinetic model, in addition

to reversing $v_{\parallel}$, now all three spatial coordinates need to be updated according to the reflection of the marker along the B-field line. In figure 3.4 b), this reflection of the marker along the B-field line is illustrated. However, in a gyrokinetic model the complexity of the reflection behavior increases even more, due to the gyro-ring, which is now not perpendicular to the boundary surface anymore. To understand how this can be treated for arbitrary B-field incidence angles, we will in the following introduce our approach for the rotation of the gyro-ring in the (x, y, z)-domain, the calculation of the marker velocity perpendicular to the boundary and the reflection along the incident B-field line.
It is important to mention that we did not yet treat the case of non-perpendicular B-field angles for the calculation of the electric potential. So far in PICLS the Poisson equation can only be consistently solved for perpendicular strike angles onto the domain boundary.

3.4.1 Rotation of the gyro-ring

This consideration is obviously only required for a gyrokinetic model and not for a drift-kinetic one. In the gyrokinetic case, the gyro-ring of a marker is always perpendicular to the B-field vector $\mathbf{B}$ at the gyro-center coordinate (x, y, z). By a given B, B_x, B_y and B_z the direction of $\mathbf{B}$ can be defined by two angles, the angle between $\mathbf{B}$ and the z-axis α_z and the angle between $\mathbf{B}$ projected on the x/y-plane (B_{xy}) and the x-axis α_x (this is a free choice). These angles are illustrated in figure 3.5. With the definitions

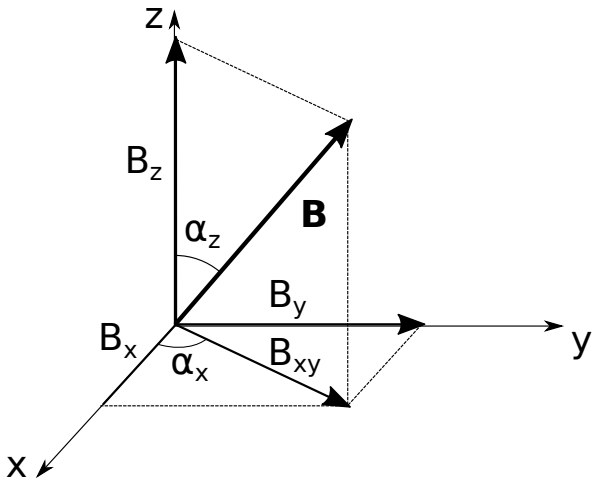


Figure 3.5: Illustration of an arbitrary B-field vector $\mathbf{B}$ in the (x, y, z) domain. The direction of $\mathbf{B}$ can be determined by the angle between $\mathbf{B}$ and the z-axis α_z and the angle between $\mathbf{B}$ projected on the x/y-plane (B_{xy}) and the x-axis α_x.

$B_{xy} = \sqrt{B_x^2 + B_y^2}$ and $B = \sqrt{B_x^2 + B_y^2 + B_z^2}$, we obtain:

$$\sin(\alpha_z) = \frac{B_{xy}}{B}, \quad \cos(\alpha_z) = \frac{B_z}{B}, \quad \tan(\alpha_z) = \frac{B_{xy}}{B_z}, \tag{3.6}$$

$$\sin(\alpha_x) = \frac{B_y}{B_{xy}}, \quad \cos(\alpha_x) = \frac{B_x}{B_{xy}}, \quad \tan(\alpha_x) = \frac{B_y}{B_x}. \tag{3.7}$$

These angles can be used to identify the plane in which the gyro-ring lies. The approach we use is to construct the gyro-ring in the (x, y)-plane and then rotate the plane perpendicularly to $\mathbf{B}$.

1) Construct gyro-ring

Using the definition of the Larmor radius (in CGS):

$$r_\mathrm{L} = \sqrt{2\mu m/B} \cdot \frac{c}{|q|}, \tag{3.8}$$

we can calculate the x/y-coordinates of the Larmor-points around the point of origin $(0, 0, 0)$ in the $x/y/z = 0$-plane:

$$\begin{aligned} x_{1,i} &= r_\mathrm{L} \cdot \cos(2\pi i/n_\mathrm{L}), \\ y_{1,i} &= r_\mathrm{L} \cdot \sin(2\pi i/n_\mathrm{L}). \end{aligned} \tag{3.9}$$

Where we define i as the index of the specific Larmor point and n_L as the number of Larmor points per gyro-ring. Here we use, that in our PIC scheme a finite number of gyro-markers needs to be distributed across the gyro-ring to take gyrokinetic effects into account. The number of gyro-markers depends on the size of r_L and is calculated within the simulation (typical values are 4-8). The z-coordinate remains unchanged:

$$z_{1,i} = 0. \tag{3.10}$$

2) Rotate around y- and z-axis

We now apply two consecutive rotations around the y- and z-axis to rotate the gyro-ring into the plane perpendicular to $\mathbf{B}$. First, we rotate the gyro-ring around the y-axis with the previously calculated angle α_z by using the matrix:

$$M_y(\alpha_z) = \begin{bmatrix} \cos(\alpha_z) & 0 & \sin(\alpha_z) \\ 0 & 1 & 0 \\ -\sin(\alpha_z) & 0 & \cos(\alpha_z) \end{bmatrix}. \tag{3.11}$$

In a second step we now rotate the already y-rotated gyro-ring around the z-axis by the angle α_x, to eventually position it in the plane perpendicular to $\mathbf{B}$. Therefore, we define a second rotation matrix:

$$M_z(\alpha_x) = \begin{bmatrix} \cos(\alpha_x) & -\sin(\alpha_x) & 0 \\ \sin(\alpha_x) & \cos(\alpha_x) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.12}$$

Both matrices can be multiplied to obtain a rotation matrix $M_{yz}(\alpha_z, \alpha_x)$ around the y- and z-axis with the angles α_z and α_x:

$$M_{yz}(\alpha_z, \alpha_x) = \begin{bmatrix} \cos(\alpha_x)\cos(\alpha_z) & -\sin(\alpha_x) & \cos(\alpha_x)\sin(\alpha_z) \\ \sin(\alpha_x)\cos(\alpha_z) & \cos(\alpha_x) & \sin(\alpha_x)\sin(\alpha_z) \\ -\sin(\alpha_z) & 0 & \cos(\alpha_z) \end{bmatrix}. \tag{3.13}$$

We can now multiply $M_{yz}(\alpha_z, \alpha_x)$ with the original gyro-ring coordinates $(x_{1,i}, y_{1,i}, z_{1,i})$ and add the coordinates of the gyro-ring center (x_0, y_0, z_0) to get the coordinates of the gyro-points (x_i, y_i, z_i) in the plane perpendicular to $\mathbf{B}$:

$$\begin{aligned} x_i &= x_0 + x_{1,i}\cos(\alpha_x)\cos(\alpha_z) - y_{1,i}\sin(\alpha_x) + z_{1,i}\cos(\alpha_x)\sin(\alpha_z), && (3.14)\\ y_i &= y_0 + x_{1,i}\sin(\alpha_x)\cos(\alpha_z) + y_{1,i}\cos(\alpha_x) + z_{1,i}\sin(\alpha_x)\sin(\alpha_z), && (3.15)\\ z_i &= z_0 - x_{1,i}\sin(\alpha_z) + z_{1,i}\cos(\alpha_z) && (3.16) \end{aligned}$$

Using that $z_{1,i} = 0$ and equations (3.9), we obtain:

$$\begin{aligned} x_i &= x_0 + r_\mathrm{L}\cdot\cos(2\pi i/n_\mathrm{L})\cos(\alpha_x)\cos(\alpha_z) - r_\mathrm{L}\cdot\sin(2\pi i/n_\mathrm{L})\sin(\alpha_x), && (3.17)\\ y_i &= y_0 + r_\mathrm{L}\cdot\cos(2\pi i/n_\mathrm{L})\sin(\alpha_x)\cos(\alpha_z) + r_\mathrm{L}\cdot\sin(2\pi i/n_\mathrm{L})\cos(\alpha_x), && (3.18)\\ z_i &= z_0 - r_\mathrm{L}\cdot\cos(2\pi i/n_\mathrm{L})\sin(\alpha_z)' && (3.19) \end{aligned}$$

3.4.2 Velocity perpendicular to boundary

In case of perpendicular B-field lines on the sheath boundary one of the key parameters to consider is $v_\parallel$, which is also perpendicular to the sheath boundary and determines whether the incoming particle has enough velocity to overcome the sheath potential or not. For non-perpendicular B-field lines, however, the direction of $v_\parallel$ is also not perpendicular anymore, but under a certain angle $\neq 0$. In this case, one could assume that the velocity we would have to calculate is the velocity perpendicular to the sheath boundary, which differs from $v_\parallel$. This perpendicular velocity could be calculated from the incident angle of the field line, the distance of the particle to the sheath surface and

the gyro-ring radius. However, in most cases the reflection or absorption of particles is sufficiently determined by $v_\parallel$. Thus, using $v_\parallel$ as velocity for the logical sheath criterion is appropriate. Only for very small incident angles of the B-field (approximately $< 2°$) this cannot longer be justified, due to the influence of collisions on the cross-field transport [119].

3.4.3 Reflection along B-field line

Once evaluated that a marker is reflected back into the plasma domain, the reflection trajectory needs to be calculated. Our approach is to evaluate how far the marker would have travelled out of the domain along the z-direction, if no boundary would be present. Therefore, we define the z-position where the particle starts as z_{old} and the position where the particle would end as z_{new}. The distance between both is defined as:

$$\delta z = |z_{\text{new}} - z_{\text{old}}|. \tag{3.20}$$

In addition, we define the length of this path within and out of the domain:

$$\begin{aligned} z_{\text{in}} &= |z_{\text{old}} - z_{\text{sh}}|, \\ z_{\text{out}} &= |z_{\text{new}} - z_{\text{sh}}|, \end{aligned} \tag{3.21}$$

where z_{sh} is the z-position of the sheath (see figure 3.6). We assume that the marker follows the same trajectory before and after the reflection. Therefore, we calculate the fraction of time the particle travels away from its old position as:

$$\delta t = \frac{|z_{\text{in}} - z_{\text{out}}|}{\delta z} \cdot \Delta t, \tag{3.22}$$

where we used eqs. (3.21) and the finite time step Δt. The direction of the net particle movement away from its "old" position depends on whether $z_{\text{in}} > z_{\text{out}}$, or not, and can be taken into account by setting:

$$\begin{aligned} v_{\parallel,\text{old}} &\rightarrow v_{\parallel,\text{old}}, \quad \text{for} \quad z_{\text{in}} < z_{\text{out}}, \\ v_{\parallel,\text{old}} &\rightarrow -v_{\parallel,\text{old}}, \quad \text{for} \quad z_{\text{in}} > z_{\text{out}}. \end{aligned} \tag{3.23}$$

The new marker coordinates $(x, y, z, v_\parallel)_{\text{new}}$ can thus be calculated by applying the Euler-Lagrange equations on the "old" spatial marker coordinates $(x, y, z, v_\parallel)_{\text{old}}$, with the "old" E-field values, the updated velocity (see eq. (3.23)) and the new time step δt (see eq. (3.22)). $v_{\parallel,\text{new}}$ remains the same velocity as calculated within the particle pusher, only its sign changes to $v_{\parallel,\text{new}} = -v_{\parallel,\text{new}}$.

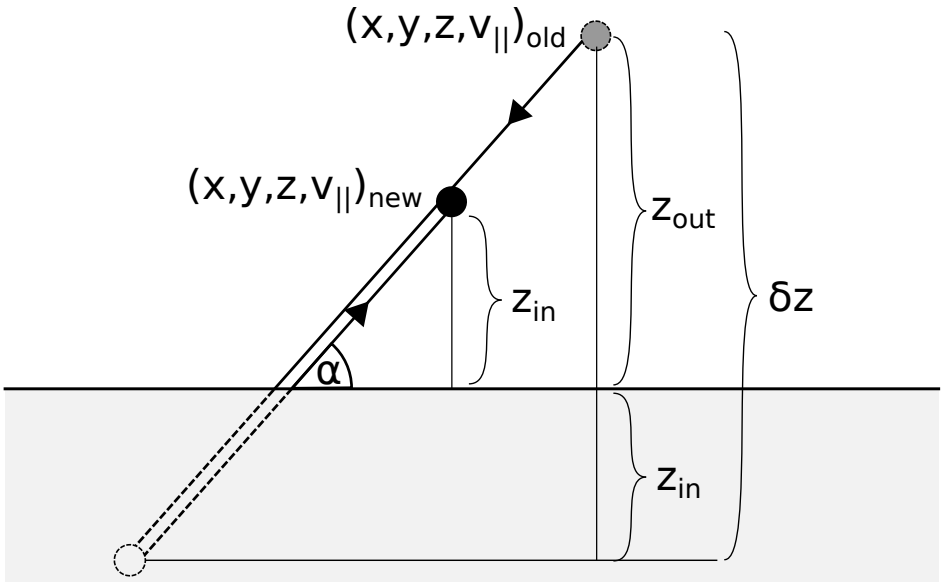


Figure 3.6: Illustration of a marker's trajectory along a B-field line which is reflected from the boundary back into the plasma domain. The B-field line strikes the boundary surface under an angle α. δz is the distance the particle would travel in z-direction without a boundary. z_{in} and z_{out} are the distances the marker travels inside and outside the wall. The marker in reality is reflected at the boundary and travels back on the incident field line with the same length z_{in} in the z-direction.

3.5 Additional considerations for sheath models

Due to their relevance in plasma physics, there are numerous studies and a vast literature on sheaths. Sheaths have been studied with different incident angles of the magnetic field and also kinetic effects have been taken into account [119]. For gyrokinetic turbulence studies, the Polarization equation (2.40) needs to be solved to couple the gyrokinetic upstream region with the sheath region. This coupling heavily depends on the specific model and how it calculates the electric field close to the boundary.
The previously described insulating and conducting sheath models are simplified models that mainly determine the particle flux into the wall by either determining the sheath potential, or the current into the boundary. However, several effects are not yet taken into account. For the conducting sheath for example, the currents can freely flow into the wall, which at short time scales can lead to a violation of the quasi-neutrality condition. However, due to the effect of the electric field on longer time scales, the plasma reacts to restore its quasi-neutrality. To also prevent this short time violation, a current flow from one region of the sheath to another region could be implemented to

ensure $j_{||} = 0$ over the whole sheath.
In addition, our implemented models do not guarantee the Bohm sheath criterion. Here, for a steady-state sheath in the sheath entrance, the ion outflow velocity needs to exceed the sound speed c_s. However, if the pressure and potential gradients from the source to the sheath region are large enough to accelerate ions to near-sonic velocities, this criterion does not have to be enforced additionally. In the simulations within this thesis, we generally start with a low density and increase it steadily, until a steady state is reached over a period of a few sound transit times. During this time, the potential and pressure drop from the central source towards the boundaries is sufficiently large, to accelerate the ions to near-sonic velocities.
Other physically relevant factors, such as a neutrals model, which is relevant to achieve detached plasma states, could also be added to a more realistic sheath model. Also charge exchange, recycling and recombination at the vessel wall could be of interest.
Different from gyrokinetic models in fluid models in recent years a lot of research was dedicated to improving sheath models, to ensure the development of consistent boundary conditions (e.g., [117], [120], [121]). Despite the achieved progress, sheath models still remain to be an open research topic and the results gained from fluid code studies could in future also become relevant to GK codes.

3.6 Source term

As described, we apply sheath boundary conditions on both sides of the domain in z-direction at positions $z = 0$ and $z = L_z$ to reflect open field lines starting and ending at a divertor. These boundary conditions can also be seen as particle sinks, depending on the criteria determined by the sheath to exit the domain. Hence, we also need to allow particles to enter the domain between these two boundaries. Within PICLS, this is done via a particle source, which generates particles from a pre-defined source function (e.g., Maxwellian distribution) via a Monte Carlo method. The type of source function strongly depends on the simulation setup and the specific problem to be studied. In general it is dependent on the plasma temperature, the plasma density and the position within the domain. In addition, a particle source intensity can be defined, which defines the number of particles inserted per time.

4 Collision operator

4.1 Collisions in the SOL

As mentioned in section 1.2 apart from micro instabilities, collisions are a key driver to transport particles across closed magnetic flux surfaces that would be confined otherwise [20]. With the resulting particle transport also heat transport across flux surfaces is achieved [7, 13]. These collisional transport phenomena are described by the classical and, in the specific case of a tokamak, neoclassical transport theory [13, 122].
In the SOL, collisions are of specific relevance, since they cause plasma to diffuse from the confined region into SOL from where they are transported towards the device wall [20]. Collisionality is higher in the edge/SOL region than in the core region, due to lower temperature there. It plays a dominant role at the plasma edge and then changes significantly across the SOL [15, 20].
This importance of collisional effects in the SOL was also suggested by fully kinetic PIC simulations ([123],[124]). Especially in highly collisional machines, such as LAPD, collisions are significant for the transport behavior in open field lines.

4.2 Coulomb collisions and Landau operator

In gyrokinetics, apart from the self-consistent electromagnetic mean fields, due to the collective particle behavior, plasma particles of same and different species are also coupled through binary interactions (collisions). For charged particles these collisions are expressed by Coulomb collisions [125].
Here, two charged particles with charge q_1 and q_2 and mass m_1 and m_2 are considered. Figure 4.1 sketches an exemplary electron-ion Coulomb collision, with the relative position $\mathbf{r}$ and $\mathbf{v} = \frac{d\mathbf{r}}{dt}$ initially along the x axis, the impact parameter b, the angle α and the deflection angle θ. Introducing the reduced mass $\mu = m_1 m_2/(m_1 + m_2)$ delivers the equation of motion:

$$\mu \frac{d\mathbf{v}}{dt} = \frac{q_1 q_2 \mathbf{r}}{4\pi\epsilon_0 r^3} \tag{4.1}$$

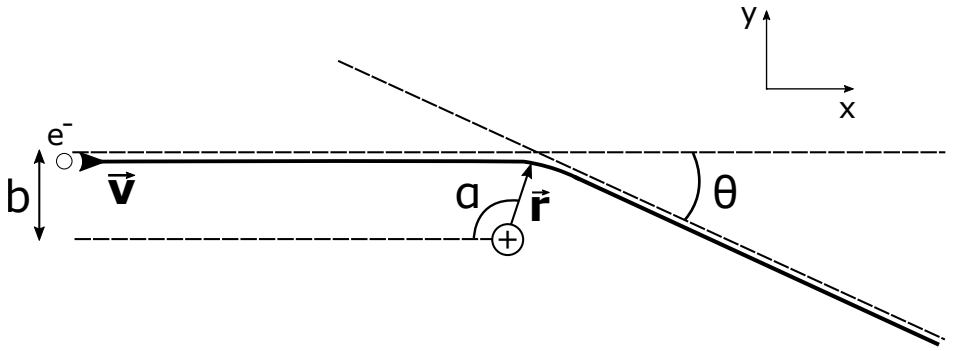


Figure 4.1: Exemplary representation of an electron-ion Coulomb collision.

and the angular momentum conservation:

$$vb = r^2\dot{\alpha} \rightarrow \frac{1}{r^2} = \frac{\dot{\alpha}}{vb}. \tag{4.2}$$

With equations 4.1 and 4.2 the relative velocity gain and thus a relation between the impact parameter and the deflection angle can be calculated:

$$\tan\frac{\theta}{2} = \frac{|q_1 q_2|}{4\pi\epsilon_0\mu v^2 b} = \frac{b_0}{b}, \tag{4.3}$$

where $b_0 = |q_1 q_2|/4\pi\epsilon_0\mu v^2$ is the impact parameter that would result in a 90° deflection. The deflection angle is related to the center of mass.
To account for these Coulomb collisions in a gyrokinetic model, a collision operator needs to be introduced. A general total collision operator for species p includes the sum of contributions from collisional interactions with all other species p' (where $p = p'$ is possible):

$$C_p[f_p] = \sum_{p'} C_{pp'}[f_{p'}, f_p], \tag{4.4}$$

where f_p is the particle distribution function of species p. The full Landau operator accounts for the binary particle interaction between charged particles by considering the electrostatic fields they generate [122, 126]. The Landau operator can be written in an explicit drag and diffusion form [125]:

$$C_{pp'}[f_{p'}, f_p] = \frac{\partial}{\partial \mathbf{v_p}} \cdot \left[\Gamma_{pp'} f_p - \frac{\partial}{\partial \mathbf{v_p}} \cdot (\bar{D}_{pp'} f_p)\right], \tag{4.5}$$

with the drag vector $\Gamma_{pp'}$ and the diffusion tensor $\bar{D}_{pp'}$ defined as:

$$\Gamma_{pp'} = \frac{q_p^2 q_{p'}^2 \ln\Lambda}{8\pi\epsilon_0^2 m_p^2 \mu_{pp'}} \int \bar{U}(\mathbf{u}) \cdot \frac{\partial f_{p'}}{\partial \mathbf{v}_{p'}} \mathrm{d}^3 v_{p'}, \tag{4.6}$$

$$\bar{D}_{pp'} = \frac{q_p^2 q_{p'}^2 \ln\Lambda}{8\pi\epsilon_0^2 m_p^2} \int \bar{U}(\mathbf{u}) f_{p'} \mathrm{d}^3 v_{p'}. \tag{4.7}$$

Here, the collision relative velocity $\mathbf{u} = \mathbf{v_p} - \mathbf{v_{p'}}$ as well as the tensor $\bar{U}(\mathbf{u}) = 1/u(\mathbb{1} - \hat{u}\hat{u})$ are used. $\ln\Lambda$ is the Coulomb logarithm, defined as the integral over the inverse collision impact parameter b, $\ln\Lambda = \int_{b_0}^{\lambda_\mathrm{D}} \frac{\mathrm{d}b}{b}$, and ϵ_0 is the permittivity of the free space. In our model however, we implement a simplified and less computationally heavy Lenard-Bernstein (or Dougherty) collision operator, which will be describe in the following chapter and which was already introduced in [127].

4.3 Lenard-Bernstein (LB) collision operator

The Lenard-Bernstein (LB) collision operator can be used in the presence of small-angle collisions and allows exact analytic solution. This model conserves certain features of the actual Landau collision terms. In particular, collision driven diffusion in velocity space, which causes a relaxation of the distribution function towards the Maxwell distribution and conserves the number of particles. In the limit of infrequent collisions, the results of a Landau operator can be retrieved [128]. However, the evaluation of so-called Rosenbluth potentials (see [126]) is avoided by using the simplified LB operator. These Rosenbluth potentials are expressions that appear in the drag and diffusion terms of the previously shown Landau operator. In addition to the relaxation towards a Maxwellian, the operator also conserves particle number, momentum, and energy analytically and implies pitch-angle scattering. It is long wavelength and thus ignores finite-Larmor-radius (FLR) corrections, which implies classical cross-field diffusion. Collisions with neutrals are neglected in the current version.
The Lenard-Bernstein collision operator acting on the full-f model can be written as:

$$\begin{aligned} C_p[f_p] &= \sum_{p'} C_{pp'}[f_p] \\ &= \sum_{p'} \nu_{pp'} \frac{\partial}{\partial \mathbf{v}} \cdot \left[(\mathbf{v} - \mathbf{u}_{\parallel,\mathbf{p'}}) f_p + v_{\mathrm{T},pp'}^2 \frac{\partial f_p}{\partial \mathbf{v}} \right] \\ &= \sum_{p'} \nu_{pp'} \left(\frac{\partial}{\partial v_\parallel} \left[(v_\parallel - u_{\parallel,p'}) f_p + v_{\mathrm{T},pp'}^2 \frac{\partial f_p}{\partial v_\parallel} \right] + \frac{\partial}{\partial \mu} \left[2\mu f_p + 2\frac{m_p v_{\mathrm{T},pp'}^2}{B} \mu \frac{\partial f_p}{\partial \mu} \right] \right), \end{aligned} \tag{4.8}$$

with the definitions:

$$u_\parallel = \frac{\int \mathrm{d}^3 v v_\parallel f_p}{n_p}, \tag{4.9}$$

$$m_p v_{\mathrm{T},pp'}^2 = \frac{\int \mathrm{d}^3 v m_p (\mathbf{v} - \mathbf{u}_{\mathbf{p}'})^2 f_p}{3 n_p}, \tag{4.10}$$

$$n_p = \int \mathrm{d}^3 v f_p. \tag{4.11}$$

Additionally, for the collision frequencies for self-species collisions standard expressions can be used [129]:

$$\nu_{\mathrm{ee}} = \frac{4\sqrt{2\pi} n_{\mathrm{e}} \lambda e^4}{3\sqrt{m_{\mathrm{e}}} T_{\mathrm{e}}^{3/2}} \quad \text{and} \quad \nu_{\mathrm{ii}} = \frac{4\sqrt{\pi} n_{\mathrm{i}} \lambda e^4}{3\sqrt{m_{\mathrm{i}}} T_{\mathrm{i}}^{3/2}}, \tag{4.12}$$

with the Coulomb logarithm $\lambda = 6.6 - 0.5 \ln(n_0/10^{20}) + 1.5 \ln T_{\mathrm{e}0}$, where the initial density n_0 is expressed in m^{-3} and the initial electron temperature $T_{\mathrm{e}0}$ in eV. The collision frequency is velocity independent and thus neglects the v^{-3} dependence expected for Coulomb collisions.
For electron-ion collisions for simplicity we also apply the Lenard-Bernstein collision operator with the collision frequency $\nu_{\mathrm{ei}} = \nu_{\mathrm{ee}}/1.96$. This value approximately accounts for the plasma's parallel conductivity coefficient. An option to solely apply pitch-angle-scattering is also implemented in the code [125, 130]. Ion-electron collisions are neglected, since its collision frequency ν_{ie} is much smaller than the ion-ion term ($\nu_{\mathrm{ie}}/\nu_{\mathrm{ii}} \sim \sqrt{m_{\mathrm{e}}/m_{\mathrm{i}}}$). Due to this small collision frequency, neglecting ion-electron collisions does not have a significant effect. This approximation is also used in previous gyrokinetic continuum code studies in Gkeyll [90] and GENE [95].
From equation (4.8), the drag coefficient Γ and diffusion coefficient D can be excerpted:

$$\Gamma = -\nu_{pp'}(\mathbf{v} - \mathbf{u}_{\parallel,\mathbf{p}'}), \tag{4.13}$$

$$D = \nu_{pp'} v_{\mathrm{T},pp'}^2. \tag{4.14}$$

Both coefficients are required for the discretization of the collision operator in a particle-in-cell model.
Instead of the selected Lenard-Bernstein collision model, also other options would have been possible, such as a simple Krook operator, or a more advanced Fokker-Planck operator, which also evaluates the Rosenbluth potentials (see [125]). However, the reason for our choice was to implement a collision operator that was similar to the one used for other gyrokinetic continuum simulations [90, 95], to ensure comparability.

Additionally, for our current studies the characteristics of the LB operator are sufficient and computationally relatively reasonable. In future, however, to achieve a more realistic collisional model, we plan to implement a non-linear Fokker-Planck collision operator, similar to the one introduced in [125].

4.4 PIC discretization of LB collision operator

For the discretization of the described Lenard-Bernstein collision operator for our PIC model, the so-called Langevin approach is used, as explained in Vernay (2013) [125]. Here, a collection of particles subjected to deterministic and stochastic motion in a general 1D phase space (x,t) is considered. The probability that a particle moves from position x_1 to x_2 between times t_1 and t_2, $P(x_1, x_2, t_1, t_2) \cdot (x_2 - x_1)$ is introduced. The statistical particle distribution then can be defined as:

$$f(x,t) = \int f(x - \Delta x, t - \Delta t) P(x - \Delta x, x, t - \Delta t, t) \mathrm{d}\Delta x \tag{4.15}$$

Eq. (4.15) can be Taylor expanded with respect to Δx, by assuming that the deflections Δx are small:

$$\begin{aligned} f(x,t) \;\approx\; & f(x, t - \Delta t) \underbrace{\int P(x - \Delta x, x, t - \Delta t, t) \mathrm{d}\Delta x}_{=1} \\ & - \frac{\partial}{\partial x} \left[f(x, t - \Delta t) \underbrace{\int \Delta x P(x - \Delta x, x, t - \Delta t, t) \mathrm{d}\Delta x}_{=\langle \Delta x \rangle} \right] \\ & + \frac{\partial^2}{\partial x \partial x} \left[f(x, t - \Delta t) \underbrace{\int \Delta x \Delta x P(x - \Delta x, x, t - \Delta t, t) \mathrm{d}\Delta x}_{=\langle \Delta x \Delta x \rangle} \right]. \end{aligned} \tag{4.16}$$

Dividing (4.16) by Δt and taking the limit $\Delta t \to 0$ gives the Fokker-Planck equation:

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}(\Gamma f) - \frac{\partial^2}{\partial x \partial x}(D f) = 0, \tag{4.17}$$

with the drag and diffusion coefficients defined as:

$$\Gamma = \lim_{\Delta t \to 0} \frac{\langle \Delta x \rangle}{\Delta t}, \tag{4.18}$$

$$D = \lim_{\Delta t \to 0} \frac{\langle \Delta x \Delta x \rangle}{2\Delta t}. \tag{4.19}$$

By introducing particles to model the distribution function f, equation (4.17) can be solved with the Langevin approach [125, 131]. The time evolution of the whole distribution f is therefore represented by individual single particle motions. For particle n at time t, the position in phase space $x_n(t)$, is given by its previous position $x_n(t - \Delta t)$ at time step $t - \Delta t$. The temporal change of the variable Δx_i is thus given by:

$$\Delta x_n = x_n(t) - x_n(t - \Delta t) = \langle \Delta x \rangle + \mathcal{R}\sqrt{\langle \Delta x \Delta x \rangle} = \Gamma \Delta t + \mathcal{R}\sqrt{2D\Delta t}. \tag{4.20}$$

$\mathcal{R}$ is a random number which is sampled from a probability distribution function (PDF) of average 0 and variance 1. This random number is required for the diffusive motion in the second moment of the probability function P (last term of eq. 4.16). In terms of the PIC algorithm, the Langevin approach performs random kicks on the velocity coordinates of the particles, that maintain the previously mentioned features, such as energy conservation.
Treating collisions in the gyrokinetic 2D velocity space $(v_\parallel, \mu)$ with random numbers on a finite time step can however cause $\xi = v_\parallel / v$ to leave its correct interval of $[-1, 1]$ [125]. Thus, for the collision operation the velocity space needs to be expanded back to the real 3D space (v_x, v_y, v_z). Based on the form of the LB collision operator we choose a coordinate system where the velocity of the incoming particle $\mathbf{v}_{\text{in}}$ lies in the x/z-plane. In particular, $v_{\perp,\text{in}}$ lies on the x-axis and $v_{\parallel,\text{in}}$ on the z-axis:

$$\mathbf{v}_{\text{in}} = \begin{pmatrix} v_{\parallel,\text{in}} \\ v_{\perp,\text{in}} \end{pmatrix}_{\text{2D}} = \begin{pmatrix} v_{\perp,\text{in}} \\ 0 \\ v_{\parallel,\text{in}} \end{pmatrix}_{\text{3D}} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}_{\text{3D}}, \tag{4.21}$$

with $v_\perp = \sqrt{2B(\mathbf{R})\mu/m}$.
We can use the drag and diffusion coefficients (4.13, 4.14) in the Langevin approach

(4.20), to achieve the following form for the velocity change in the v_x, v_y, v_z space:

$$\begin{aligned}\Delta\mathbf{v} &= \Gamma\Delta t + \mathcal{R}\sqrt{2D\Delta t} \\ &= -\nu_{pp'}(\mathbf{v} - \mathbf{u}_{\parallel,\mathbf{p}'})\Delta t + \sqrt{2\nu_{pp'}v_{\mathrm{T},pp'}^2\Delta t}\mathcal{R} \\ &= -\nu_{pp'}\left(\begin{pmatrix} v_{\perp,\mathrm{in}} \\ 0 \\ v_{\parallel,\mathrm{in}} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ u_{\parallel,p'} \end{pmatrix}\right)\Delta t + v_{\mathrm{T},pp'}\sqrt{2\nu_{pp'}\Delta t}\mathcal{R}, \end{aligned} \tag{4.22}$$

where we used that $\mathbf{u}_{\parallel,\mathbf{p}'}$ is in the $v_\parallel$/z-direction. For the changes in $x/y/z$-direction we then get:

$$\begin{aligned}\Delta v_x &= -\nu_{pp'}v_{\perp,\mathrm{in}}\Delta t + v_{\mathrm{T},pp'}\sqrt{2\nu_{pp'}\Delta t}\mathcal{R}_1, \\ \Delta v_y &= v_{\mathrm{T},pp'}\sqrt{2\nu_{pp'}\Delta t}\mathcal{R}_2, \\ \Delta v_z &= -\nu_{pp'}(v_{\parallel,\mathrm{in}} - u_{\parallel,p'})\Delta t + v_{\mathrm{T},pp'}\sqrt{2\nu_{pp'}\Delta t}\mathcal{R}_3, \end{aligned} \tag{4.23}$$

with the independent random numbers $\mathcal{R}_1$, $\mathcal{R}_2$ and $\mathcal{R}_3$; each of them sampled from a PDF of average 0 and variance 1. One has to reverse transform the coordinates back to the 2D gyrokinetic velocity space, to obtain the velocity values for the particle after the collision operation:

$$v_{\parallel,\mathrm{out}} = v_{\parallel,\mathrm{in}} + \Delta v_z, \tag{4.24}$$

$$v_{\perp,\mathrm{out}} = \sqrt{(v_{\perp,\mathrm{in}} + \Delta v_x)^2 + \Delta v_y^2}. \tag{4.25}$$

This provides the outgoing gyrokinetic velocity variables ($v_{\parallel\mathrm{out}}, \mu_{\mathrm{out}} = \frac{1}{2}mv_{\perp,\mathrm{out}}^2/B$) and ensures that $\xi_{\mathrm{out}} \in [-1, 1]$.

4.4.1 Conservation of moments

Conservation of particle number, parallel momentum $\langle v_\parallel \rangle$ and kinetic energy $\langle v^2 \rangle$ is essential for the implementation of the collision operator. In the case of the Lenard-Bernstein operator analytically these quantities are conserved for an infinite number of particles and infinitely small time steps. Nevertheless, for finite values of particle number and time step length, we can introduce corrections to make sure that conservation relations hold up to round-off. As an initial step, we want to focus on a correction where we take a finite time step Δt into account, but not account for statistical errors related to finite number of particles. This is the equation implemented in PICLS so far and for our usage already shows good conservation of moments, as shown in section

4.5.2.
Particle number is intrinsically conserved in PICLS, because the code is based on a full-f model. However, for the parallel momentum and the kinetic energy, the idea is to regard $u_\parallel$ and v_T as free parameters, which are then determined in order to ensure conservation of moments. In case of the parallel momentum, the change of the parallel velocity, which is similar to Δv_z from equation (4.23) since $v_\parallel$ lies in the z-direction, is thus set to zero. Additionally, all marker weights within a configuration space bin are summed up to obtain:

$$0 = \sum_{n=1}^{N} w_n \Delta v_{\parallel,n} = \sum_{n=1}^{N} w_n \Delta v_{z,n} \tag{4.26}$$

$$= \sum_{n=1}^{N} w_n \left[-\nu (v_{\parallel,\mathrm{in},n} - u_\parallel) \Delta t + v_\mathrm{T} \sqrt{2\nu \Delta t} \mathcal{R}_{3,n} \right], \tag{4.27}$$

with the total number of markers within the bin N. Using the relation $\langle \mathcal{R}_{3,n} = 0 \rangle$, which comes from our choice of the random numbers, the second term drops and we can write a relation for $\langle u_\parallel \rangle$ to conserve parallel momentum:

$$\langle u_\parallel \rangle = \sum_{n=1}^{N} w_n v_{\parallel,\mathrm{in},n} / \sum_{n=1}^{N} w_n. \tag{4.28}$$

This is precisely the PIC discretization relation for (4.9). Hence, in the picture of infinite particle numbers, for $u_\parallel$ no correction is necessary to conserve parallel momentum. In order to conserve kinetic energy, we now have to derive a relation for v_T, for which, we can apply the already derived relation for $u_\parallel$ (4.28). For the total kinetic energy change due to Langevin kicks, the following relation must be valid on average over all particles:

$$0 = \mathbf{v}_\mathrm{out}^2 - \mathbf{v}_\mathrm{in}^2 = (\mathbf{v} + \Delta\mathbf{v})^2 - \mathbf{v}^2 = 2\mathbf{v}\Delta\mathbf{v} + (\Delta\mathbf{v})^2. \tag{4.29}$$

Writing the explicit expression of eq. (4.22) for $\Delta\mathbf{v}$ and expressing the total velocity values as sum over the markers yields:

$$\begin{aligned} 0 &= \sum_{n=1}^{N} w_n \Big\{ 2(\mathbf{v}_{\mathrm{in},n} - \mathbf{e}_z u_\parallel) \cdot \left[-\nu\Delta t(\mathbf{v}_{\mathrm{in},n} - \mathbf{e}_z u_\parallel) + v_\mathrm{T}\sqrt{2\nu\Delta t}\vec{\mathcal{R}}_n \right] \\ &\quad + \left[-\nu\Delta t(\mathbf{v}_{\mathrm{in},n} - \mathbf{e}_z u_\parallel) + v_\mathrm{T}\sqrt{2\nu\Delta t}\vec{\mathcal{R}}_n \right]^2 \Big\} \qquad (4.30) \\ &= \sum_{n=1}^{N} w_n \Big\{ -\nu\Delta t(2-\nu\Delta t)(\mathbf{v}_{\mathrm{in},n} - \mathbf{e}_z u_\parallel)^2 \\ &\quad + 2v_\mathrm{T}\sqrt{2\nu\Delta t}(1-\nu\Delta t)(\mathbf{v}_{\mathrm{in},n} - \mathbf{e}_z u_\parallel)\mathcal{R}_n + 2v_\mathrm{T}^2\nu\Delta t\vec{\mathcal{R}}_n^2 \Big\}, \qquad (4.31) \end{aligned}$$

where $\vec{\mathcal{R}} = (\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3)$. We can invoke the properties of the PDF for the random numbers $\langle\vec{\mathcal{R}}_n\rangle = 0$ and $\langle\vec{\mathcal{R}}_n^2\rangle = \langle\mathcal{R}_{1,n}^2\rangle + \langle\mathcal{R}_{2,n}^2\rangle + \langle\mathcal{R}_{3,n}^2\rangle = 3$ to obtain:

$$3(\sum_{n=1}^{N} w_n)2\nu\Delta t\langle v_\mathrm{T}^2\rangle = \nu\Delta t(2-\nu\Delta t)\sum_{n=1}^{N} w_n(\mathbf{v}_{\mathrm{in},n} - \mathbf{e}_z u_\parallel)^2. \qquad (4.32)$$

From eq. (4.32), we can directly derive a relation for $\langle v_\mathrm{T}^2\rangle$ to conserve kinetic energy:

$$\langle v_\mathrm{T}^2\rangle = (1-\nu\Delta/2)\frac{\sum_{n=1}^{N} w_n(\mathbf{v}_{in,n} - \mathbf{e}_z u_\parallel)^2}{3\sum_{n=1}^{N} w_n}. \qquad (4.33)$$

This relation is the required PIC discretization of (4.10). We want to remark that the correction factor $(1-\nu\Delta t/2)$ is required to achieve conservation of kinetic energy for finite time steps.
As a next step, relations for $u_\parallel$ and v_T can be derived to ensure conservation to round-off even in the presence of a finite number of particles, i.e. allowing for statistical fluctuations. We derived these relations, but did not implement them in the code, since the relations (4.28) and (4.33) already showed sufficient conservation properties for our requirements. The derivation for $u_\parallel$ and v_T relations again works very similar as in the previously described case, but this time no averages over markers are applied to account for fluctuations.
By enforcing conservation of parallel momentum, a first relation for v_T and $u_\parallel$ can be derived:

$$v_\mathrm{T} = \sqrt{\nu\Delta t/2}\frac{B - Au_\parallel}{C}, \qquad (4.34)$$

where the definitions $A = \sum_{n=1}^{N} w_n$, $B = \sum_{n=1}^{N} w_n v_{\parallel,\mathrm{in},n}$ and $C = \sum_{n=1}^{N} w_n \mathcal{R}_{3,n}$ were used. The second relation for v_T and $u_\parallel$ can be achieved from presuming kinetic energy conservation (see eq. 4.29) and using the definitions $D = \sum_{n=1}^{N} w_n \mathcal{R}_n^2$, $E = \sum_{n=1}^{N} w_n v_{\mathrm{in},n}^2$ and $F = \sum_{n=1}^{N} w_n \mathbf{v}_{\mathrm{in},n} \mathcal{R}_n$:

$$\begin{aligned} 2\nu\Delta t D v_\mathrm{T}^2 &= \nu\Delta t(2-\nu\Delta t)(E - 2Bu_\parallel + Au_\parallel^2) \\ &- 2(1-\nu\Delta t)\sqrt{2\nu\Delta t}(F - Cu_\parallel v_\mathrm{T}). \end{aligned} \tag{4.35}$$

By inserting (4.34) into (4.35), a quadratic second order equation can be obtained for $u_\parallel$:

$$\begin{aligned} 0 &= (\nu\Delta t)^2 \left[(\frac{A}{C})^2 D - A \right] u_\parallel^2 + 2\nu\Delta t \left[B - \nu\Delta t \frac{ABD}{C^2} \right] u_\parallel \\ &+ \left[(\nu\Delta t)^2 (\frac{B}{C})^2 D - \nu\Delta t(2-\nu\Delta t)E + 2\sqrt{2\nu\Delta t}(1-\nu\Delta t)F \right]. \end{aligned} \tag{4.36}$$

One of the resulting roots for $u_\parallel$ then has to be inserted in equation (4.34) to obtain a relation for v_T. But as mentioned previously, since the conservation properties of our first correction are more than sufficient for our requirements, we did not implemented this correction term and thus cannot comment on its stability.

4.5 Test of collision operator

To test the implementation of basic collision operator features, it is helpful to decrease the complexity of the system in a first step. After successful testing of basic features, the behavior of the full collision operator with its conservation properties can be studied.

4.5.1 Basic tests with constant $\mathbf{u}_\parallel$ and v_T

In order to test against analytical functions and decrease complexity, we choose a simplified setup with only a single species s with self-species collisions and set $\mathbf{u}_\parallel = 0$, $v_\mathrm{T} = \mathrm{const}$ and $\nu = \mathrm{const}$. Therewith, we can rewrite equation (4.8) as:

$$\frac{\partial}{\partial t} f = C_p[f] = \nu \frac{\partial}{\partial \mathbf{v}} \cdot \left[\mathbf{v} f + v_\mathrm{T}^2 \frac{\partial f}{\partial \mathbf{v}} \right]. \tag{4.37}$$

With the definitions:

$$n = \int \mathrm{d}^3 v f, \quad \langle \mathbf{v} \rangle = \frac{1}{n} \int \mathrm{d}^3 v \mathbf{v} f, \quad \langle v^2 \rangle = \frac{1}{n} \int \mathrm{d}^3 v v^2 f, \tag{4.38}$$

for density n, average velocity $\langle \mathbf{v} \rangle$ and kinetic energy of the system $\langle v^2 \rangle$, analytical expressions (exponential relaxations) can be derived and compared with the numerical simulations.
First, we take the time derivative of $\langle \mathbf{v} \rangle$, to achieve an analytical expression for the evolution of the average velocity over time:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\langle \mathbf{v} \rangle &= \frac{1}{n}\int \mathrm{d}^3 v \mathbf{v} \frac{\partial f}{\partial t} = \frac{\nu}{n}\int \mathrm{d}^3 v \mathbf{v} \frac{\partial}{\partial \mathbf{v}}\left[\mathbf{v} f + v_\mathrm{T}^2 \frac{\partial f}{\partial \mathbf{v}}\right] && (4.39) \\ &= -\frac{\nu}{n}\int \mathrm{d}^3 v \left[\mathbf{v} f + v_\mathrm{T}^2 \frac{\partial f}{\partial \mathbf{v}}\right] = -\nu \underbrace{\frac{1}{n}\int \mathrm{d}^3 v \mathbf{v} f}_{=\langle \mathbf{v} \rangle} && (4.40) \\ &= -\nu \langle \mathbf{v} \rangle. && (4.41) \end{aligned}$$

A solution of this equation can be found by the exponentially decaying function:

$$\langle \mathbf{v} \rangle (t) = \langle \mathbf{v} \rangle (t=0) e^{-\nu t}. \tag{4.42}$$

To achieve an analytical expression for the kinetic energy, we can again use the same approach, starting with the time derivative:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\langle v^2 \rangle &= \frac{1}{n}\int \mathrm{d}^3 v v^2 \frac{\partial f}{\partial t} = \frac{\nu}{n}\int \mathrm{d}^3 v v^2 \frac{\partial}{\partial \mathbf{v}}\left[\mathbf{v} f + v_\mathrm{T}^2 \frac{\partial f}{\partial \mathbf{v}}\right] && (4.43) \\ &= -2\frac{\nu}{n}\int \mathrm{d}^3 v \left[v^2 f + v_\mathrm{T}^2 \mathbf{v} \frac{\partial f}{\partial \mathbf{v}}\right] = -2\nu \langle v^2 \rangle + 6\frac{\nu}{n} v_\mathrm{T}^2 \underbrace{\int \mathrm{d}^3 v f}_{=n} && (4.44) \\ &= -2\nu(\langle v^2 \rangle - 3 v_\mathrm{T}^2). && (4.45) \end{aligned}$$

In this case, the analytical solution can be expressed as:

$$\langle v^2 \rangle (t) = 3 v_\mathrm{T}^2 + \left[\langle v^2 \rangle (t=0) - 3 v_\mathrm{T}^2\right] e^{-2\nu t}. \tag{4.46}$$

To now test the collision operator implementation in the code, an arbitrary initial velocity distribution can be chosen. This distribution has to relax according to the relations derived in (4.46) and (4.42). Additionally, the initial velocity distribution has to relax towards a Maxwellian distribution with the initially defined $\mathbf{u}_\parallel$ and v_T.
In figure 4.2, the time evolution of $\langle v_\parallel \rangle$ and $\langle v^2 \rangle$ is plotted in non-normalized units. Together with the simulation results, also the analytical solutions from equations (4.42) and (4.46) are plotted and show very good agreement. The implemented collision operator apparently is able to reproduce the analytical results in this simplified test case.

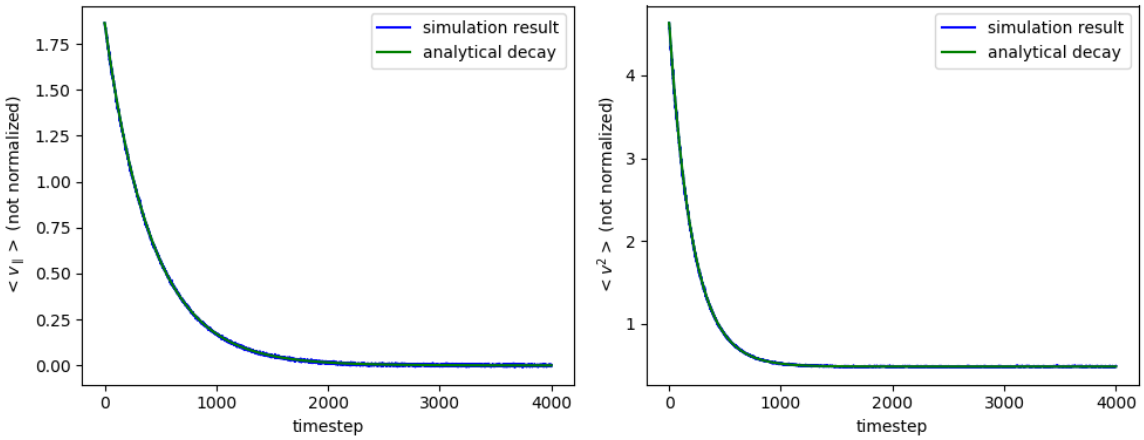


Figure 4.2: Time evolution of $\langle v_\parallel \rangle$ and $\langle v^2 \rangle$ in non-normalized units plotted per time step until a steady state is reached. The simulation results (blue) show very good agreement with the analytical solutions (green) from equations (4.42) and (4.46).

In figure 4.3 the marker distribution in $v_\parallel$ and $\xi = v_\parallel / v$ is shown for (a) an arbitrary positive initial distribution with only parallel velocity components and (b) an equilibrated state after 4000 time steps (arbitrarily chosen), to which the system relaxes. In the equilibrated state, the distribution in $v_\parallel$ has reached a Maxwellian with the defined values for $\mathbf{u}_\parallel$ and v_T. In ξ the distribution has reached an equilibrated state with equal distribution of parallel and perpendicular velocity components, as expected.
After successful testing of the simplified setup against analytical solutions, specific features of the actual collision operator shall be tested.

4.5.2 Test of conservation features and relaxation

After having performed tests on a simplified test case with a single species s and $\mathbf{u}_\parallel = 0$, $v_\mathrm{T} = \text{const}$, we now let the algorithm calculate $\mathbf{u}_\parallel$ and v_T self-consistently from the particle distribution at each time step acc. to (4.28) and (4.33). It is important to mention that here the correction factor $(1 - \nu\Delta/2)$ for v_T is implemented, as described in section 4.4.1. For the conservation tests, we still keep one single species and thus also only self-species collisions. Due to the construction of the collision operator, for any arbitrary initial state, the particle distribution should relax towards a Maxwellian in $v_\parallel$ and conserve $\langle v_\parallel \rangle$ and its kinetic energy $\langle v^2 \rangle$. The number of particles is automatically conserved, due to the chosen full-f model, which conserves the marker weights.
As already shown in [127], figure 4.4 displays the time evolutions of $\langle v_\parallel(t) \rangle / \langle v_\parallel(0) \rangle$ and $\langle v(0)^2 \rangle / \langle v(0)^2 \rangle$ for an exemplary simulation to highlight the changes of parallel

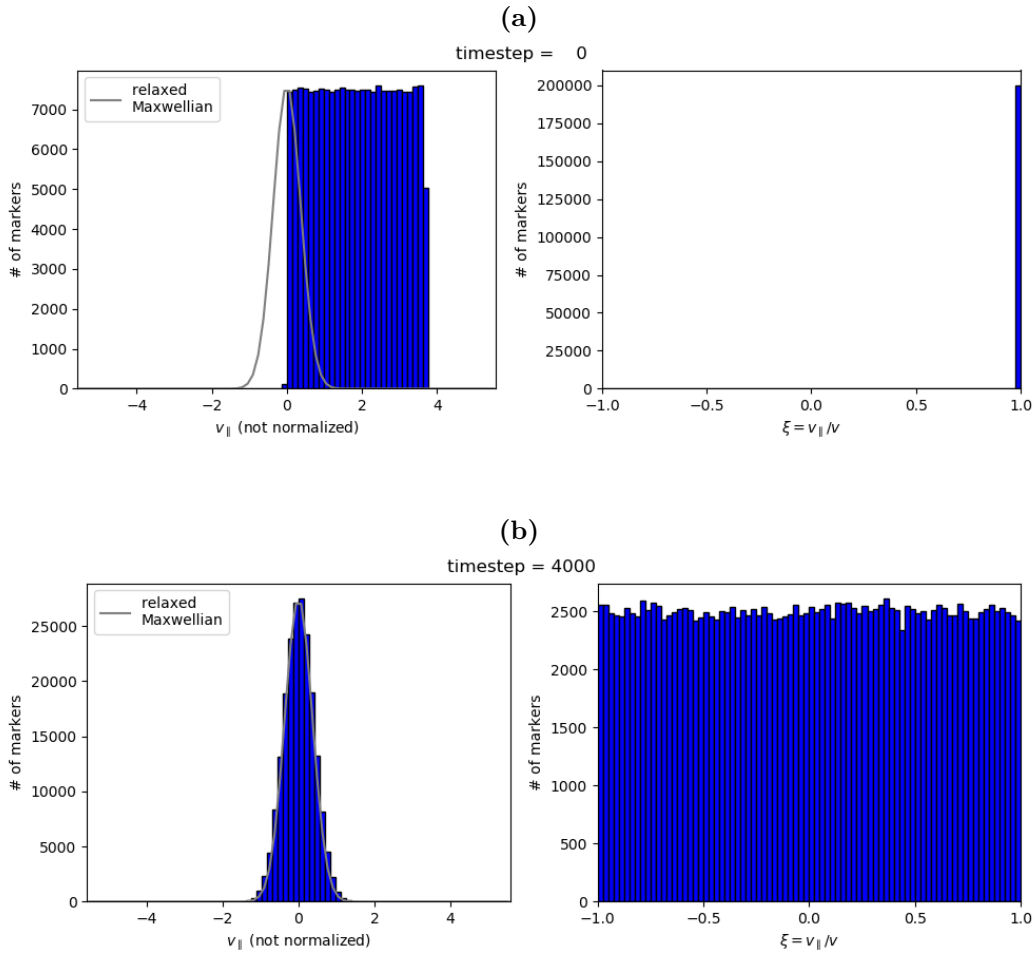


Figure 4.3: Marker distribution in $v_\parallel$ (non-normalized units) and $\xi = v_\parallel / v$ for simulations with constant $\mathbf{u}_\parallel$ and v_T is shown for (a) an arbitrary positive initial distribution with only parallel velocity components and (b) an equilibrated state which the system reaches (at 4000 time steps – arbitrarily chosen). In the equilibrated state, the distribution reaches a Maxwellian with the initially set $\mathbf{u}_\parallel$ and v_T. In ξ the distribution reaches an equal distribution.

momentum and kinetic energy. In figure 4.4, despite choosing an $\geq$3 times lower marker number per bin (of 100,000) than in the 1D heat pulse simulations in section 6.5, $\langle v_\parallel \rangle$ and $\langle v^2 \rangle$ are largely conserved and show only a variation of $< 2\%$ around their initial value (already published in [127]). Depending on the number of particles and the time step size Δt chosen, the deviations are smaller or larger. Therefore, in figure 4.5 the convergence of the conservation error for $\langle v_\parallel \rangle$ and $\langle v^2 \rangle$ is shown. The deviations decrease with increasing marker number and follow a $1/\sqrt{N_\mathrm{c}}$ dependence, where N_c is

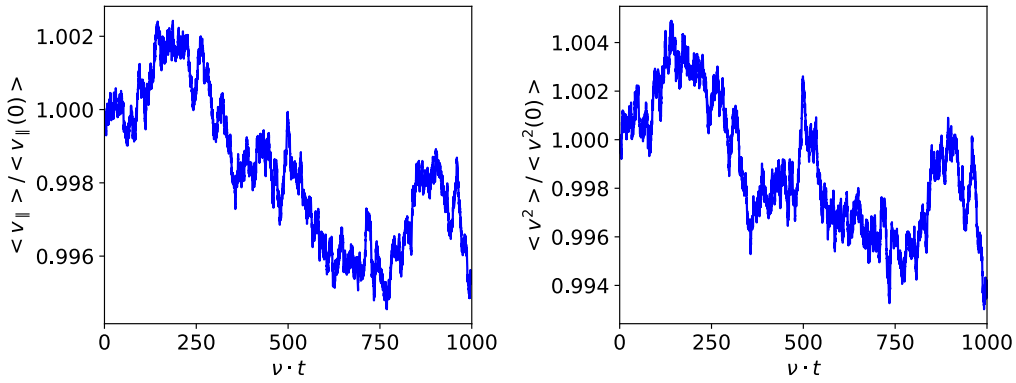


Figure 4.4: Time evolution of $\langle v_\parallel(t)\rangle/\langle v_\parallel(0)\rangle$ and $\langle v(t)^2\rangle/\langle v(0)^2\rangle$ for simulations with self-consistent calculation of $\mathbf{u}_\parallel$ and v_T and 100,000 particles per cell plotted for a total simulation time (in $\nu \cdot t$). Parallel momentum and kinetic energy are mostly conserved with $\sim 1\%$ deviation from initial values.

the number of markers per cell. In figure 4.6, again the marker distribution in $v_\parallel$ and

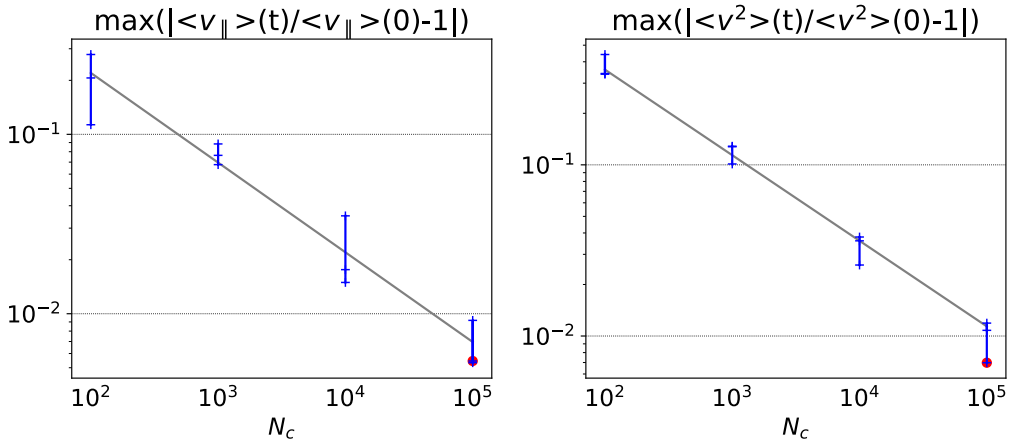


Figure 4.5: Convergence of the conservation error for $\langle v_\parallel\rangle$ and $\langle v^2\rangle$ with varying number of markers per cell (N_c). Each blue cross coincides with a conservation study similar to figure 4.4 for a specific N_c and a random initialization of markers (the data point of fig. 4.4 is indicated in red). The convergence in both cases is $\propto 1/\sqrt{N_\mathrm{c}}$ (indicated by the grey line).

$\xi = v_\parallel/v$ is shown (for $\sim 30,000$ markers per bin) for (a) an arbitrary positive initial distribution with only parallel velocity components and (b) an equilibrated state after

4000 time steps (arbitrarily chosen), to which the system relaxes. In the equilibrated state, the distribution in $v_\parallel$ again reaches a Maxwellian, but this time its maximum remains at the initialized $\mathbf{u}_\parallel$. Since $\mathbf{u}_\parallel$ is not fixed to a specific value and the parallel momentum conservation property of the collision operator holds, the particle velocities remain distributed around their initial $\mathbf{u}_\parallel$. In ξ the distribution is biased towards positive values, due to the choice of a positive initial particle distribution.

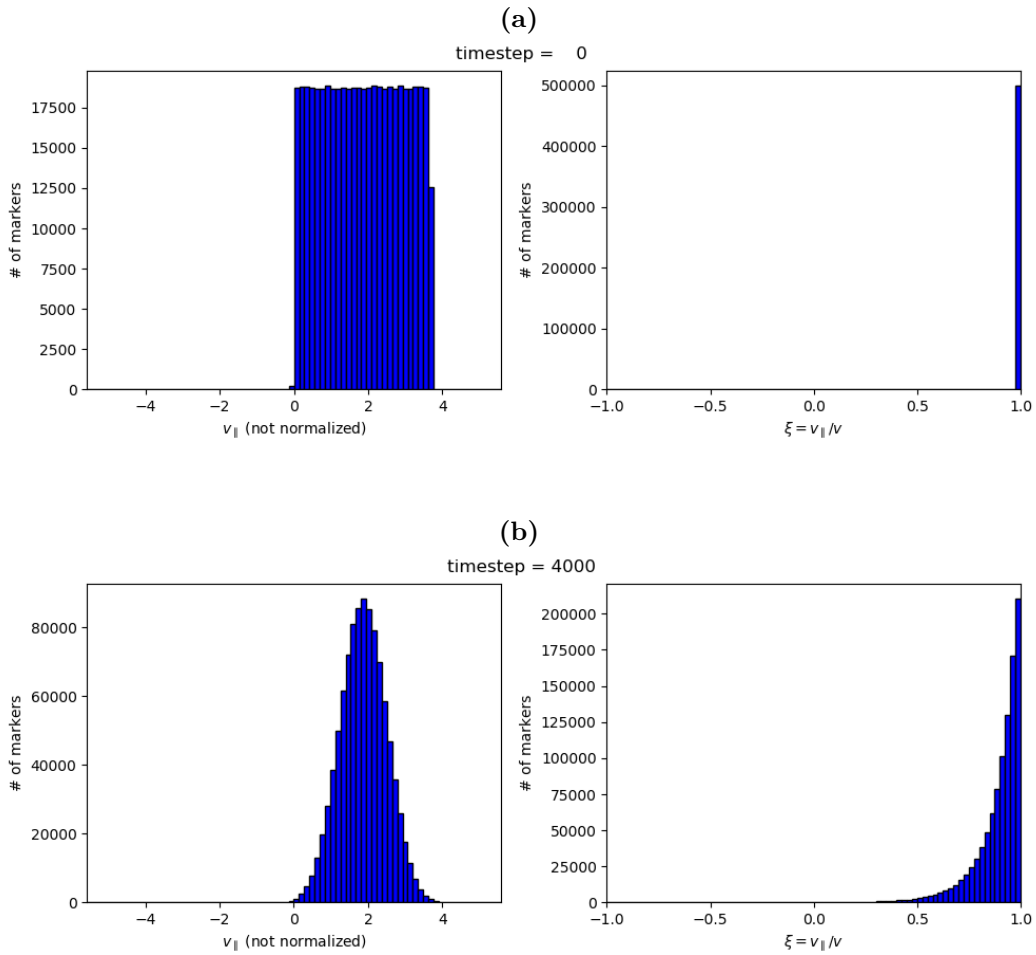


Figure 4.6: Marker distribution in $v_\parallel$ (non-normalized units) and $\xi = v_\parallel / v$ for simulations with self-consistent $\mathbf{u}_\parallel$ and v_T is shown for (a) an arbitrary positive initial distribution with only parallel velocity components and (b) an equilibrated state the system reaches (at 4000 time steps – arbitrarily chosen). In the equilibrated state, the distribution reaches the Maxwellian with the initialized $\mathbf{u}_\parallel$. In ξ the distribution reaches a steady state with a distribution biased towards positive values, due to the initial positive particle distribution.

4.6 Summary

In this chapter, we introduced the Lenard-Bernstein collision operator, which is the collisional model currently implemented in PICLS. This operator conserves particle number, momentum, and energy analytically and implies pitch-angle scattering. We then derived its particle-in-cell discretization via the Langevin approach. Here, we showed that in the discretized operator additional modifications are required to achieve conservation of parallel momentum and energy for finite values of particle number and time step length. Therefore, in our current implementation we added a correction factor to achieve conservation of moments for finite time steps, but not for a finite number of particles. We tested this operator against analytic solutions and achieved good agreement. Convergence studies with varying number of markers per cell (N_c) showed an expected $1/\sqrt{N_\mathrm{c}}$ dependence, which showed to be sufficient for our simulations. For the future, the implementation of a more advanced full-f collision operator (i.e. Fokker-Planck) is planned.

5 PICLS: Numerical methods

In this chapter the algorithmic scheme and the numerical methods used in PICLS are described. Since PICLS is a very recent code development and not part of an already existing plasma simulation code, some further understanding of its functionalities is essential. In the following, we will sketch the general algorithmic structure of PICLS and subsequently focus on the most relevant properties.

5.1 Background and general algorithmic structure

On a high level, PICLS can be described as a gyrokinetic Particle-in-Cell code for simulations in the scrape-off layer, or open field line regions. The code was initially designed to study sheath boundary models and heat and particle fluxes from the plasma towards the wall, but gets constantly extended towards more complicated open field line geometries. Just as a short side note, this is also the reason for its name "PICLS", a combination of "PIC" for Particle-In-Cell and "LS" for Logical Sheath[1].
The algorithmic structure of PICLS is mainly based on two PIC codes initially developed in the Swiss Plasma Center at EPFL Lausanne:

1. ORB5: Here I want to refer to the official ORB5 webpage (`https://www.epfl.ch/research/domains/swiss-plasma-center/research/theory/codes/research_theory_codes_orb5/`). ORB5 is a global, gyrokinetic, Lagrangian, Particle-In-Cell (PIC), finite element, electromagnetic, GPU-enabled code developed at SPC Lausanne, with important contributions from the Max-Planck IPP in Garching and Greifswald and the University of Warwick. (e.g., see [132]).

2. GK-engine: The so-called GK-engine is a gyrokinetic PIC code, written in a modular and portable structure, to provide a testbed for different numerical methods or data structures developed at SPC Lausanne (e.g., see [133]).

In general PIC is a numerical algorithm, that uses Monte Carlo methods to sample a finite collection of initial positions in phase space by a set of particles, or mostly called markers [69]. This is done via a random pull from the particle distribution function. In

[1]The pronunciation is similar to the food "pickles"

a very simplified picture, the PIC algorithm applied on gyrokinetic plasma simulations advances these particles according to the Euler Lagrange equations, with the help of a force field which is set up by a particle distribution. This distribution results from particle charge and current contributions ([134],[135]). Thus, every time a particle gets pushed by the force field towards a new position, the charge distribution resulting from these new particle positions is calculated and from the resulting potential, the new fields for the next particle advance are calculated (see figure 5.1). In section 5.2, we

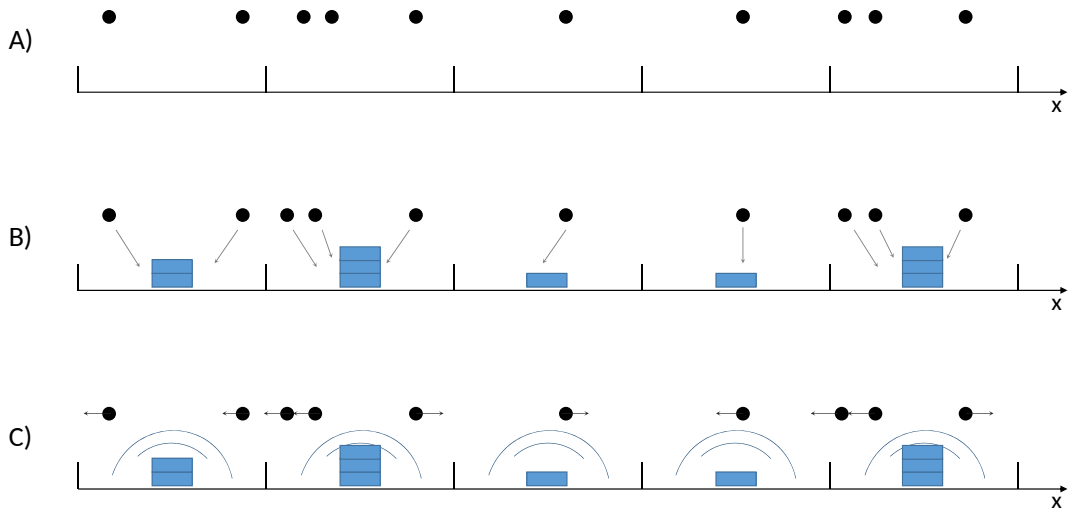


Figure 5.1: Schematic representation of main steps of a PIC algorithm. A) For each time step "particle" (indicated by black circles) positions within grid cells are calculated. B) "Particle" charges are deposited to grid cells where fields are calculated (indicated by blue boxes). C) These fields then act on the "particle" motions.

want to describe some of PICLS' main features.

5.2 Main properties and finite element methods

As mentioned in the previous section, PICLS is a gyrokinetic PIC code for simulations in open field line regions. In this part, we want to focus on the main numerical feature that are incorporated within its PIC engine. Specific details on its application in the SOL, or algorithms representing the plasma-wall interaction are not given here, but can be found in chapter 3.
Important to mention is that the algorithm is written in different variations for different spatial dimensions since several functions are not required in lower dimensionality cases. So far, a rather simple 1D (one spatial domain) case and a more advanced 3D case is

implemented in PICLS. In both cases, the algorithm basically starts with an initial loading of the particle distribution, on which it applies the high-level procedures shown in figure 5.2.

According to this, starting from the particle distribution the charges are deposited

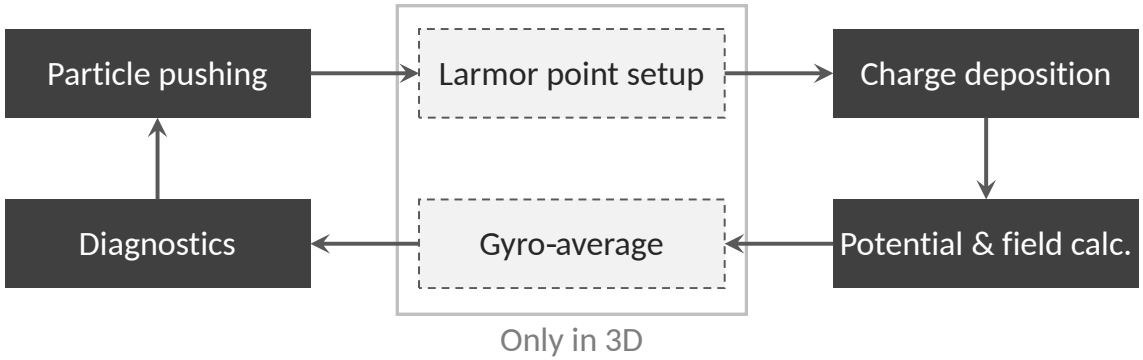


Figure 5.2: Schematic representation of main procedures applied at each time step of simulation. Procedures with bright grey background are only applied in 3D case if Larmor radius effects are considered. Other procedures are used for each dimensionality (1D and 3D).

on a field grid and from this resulting charge distribution, the potential and fields are calculated. In the simplified 1D case one now can already apply the diagnostics for system analysis and subsequently advance the particles towards their new positions according to their equations of motion. In the 3D case, however, one has to take into account that the particles are actually gyrating around their gyrocenter. This is done by setting up "Larmor points" on the Larmor radius around the gyrocenter before the charge deposition and by averaging across these Larmor points directly after the field calculation. The number of Larmor points depends on the thermal velocity of the particles and thus on the size of the gyroradius. A larger gyroradius requires a higher number of Larmor points, since the points are distributed further and thus the probability for the fields — that act on the particles — to significantly vary their value, increases.

These procedures are repeated again and again in so-called timesteps via a Runge-Kutta 4th order time integrator and for each time step system diagnostics can be extracted. If collisions are considered, the position of the collision operator depends on whether the FLR corrections are taken considered or not. In the case of our implemented LB collision operator they are not and thus the collision operator in our case acts at the beginning of the time step. In the following, we want to show the most relevant methods used within this algorithm.

5.2.1 Distribution function discretization

As noted in section 5.1 and shown in [102], the particle distribution function $f(x)$ in a PIC code is represented by discrete markers. In the full-f representation — where the whole distribution will be simulated — the particle distribution function thus can be expressed as:

$$f(\mathbf{R}, v_\parallel, t) = \sum_{n=1}^{N} w_n(t)\delta(\mathbf{R} - \mathbf{R}_n(t))\delta(v_\parallel - v_{\parallel_n}(t)) \tag{5.1}$$

with N the number of markers, w_n the marker weights, $\mathbf{R}_n$ their position and $v_{\parallel_n}$ their parallel velocity. With the definition for the initial number of physical particles, $N_{\mathrm{ph}} = \int n_0(\mathbf{R})\mathrm{d}\mathbf{R}$, the weights in our case are uniformly initialized with

$$w_n = \frac{1}{N} \tag{5.2}$$

for all markers. The weights for the full-f case are constant and thus do not change over time:

$$\frac{d}{dt} w_n = 0. \tag{5.3}$$

In the δf-case, where only the perturbed part of the distribution function is simulated, the weight evolution would be given by the Vlasov equation:

$$\frac{d}{dt} w_n = \frac{\Omega}{N_{\mathrm{ph}}} \left(\frac{d\mathbf{R}_n}{dt} \cdot \nabla f_{0,n} + \frac{dv_{\parallel_n}}{dt} \frac{\partial f_{0,n}}{\partial v_\parallel} \right), \tag{5.4}$$

with f_0 the background distribution and Ω the total phase space volume.

5.2.2 Finite Element method: B-splines

Here, we do not want to give an exhaustive overview on B-splines, but rather provide the most important properties for our application. For further and more detailed information, we refer to [136].
PICLS uses a Finite Element (FE) discretization scheme with B-splines as basis functions. To understand how B-splines are used in the code, we first need to introduce the Finite Element method. In general, for computer simulations which can just store a finite amount of information the simulation problem needs to be discretized. Several discretization schemes, like Finite Differences, Finite Volumes and Finite Elements exist. Our method of choice is the Finite element method (e.g., [137], [138]). The idea

behind this method, is to find a solution in a finite dimensional vector space. Hence, within a well-chosen vector space V_h, with basis $(\phi_i)_{0\leq i\leq N-1}$, the approximate solution $u_h(x)$ of the original function $u(x)$ has the form:

$$u_h(x) = \sum_{i=0}^{N-1} u_i \phi_i(x). \tag{5.5}$$

The approximate solution u_h is completely determined by its coefficients u_i, in the given basis. A basis that has specifically proven well for FE methods are so-called B-splines. For the introduction of B-splines, we first need a finite interval $[a, b]$ subdivided into N_x intervals:

$$a = t_0 \leq t_1 \leq ... \leq t_{N_x} = b. \tag{5.6}$$

This sequence t_i $(i = 0, ..., N_x)$ of grid points (or *knots*) can be irregularly spaced. The i-th spline of degree k Λ_i^k defined on this sequence can be constructed via the recurrence relation:

$$\Lambda_i^k(x) = w_i^k(x)\Lambda_i^{k-1}(x) + (1 - w_{i+1}^k(x))\Lambda_{i+1}^{k-1}(x), \tag{5.7}$$

with the $k = 0$ spline and the formula for w_i^k:

$$\begin{aligned} \Lambda_i^0(x) &= \begin{cases} 1 & t_i \leq x < t_{i+1} \\ 0 & otherwise, \end{cases} \\ w_i^k &= \frac{x - t_i}{t_{i+k} - t_i}. \end{aligned} \tag{5.8}$$

Starting from the $\Lambda_i^0(x)$-spline, all splines can be constructed according to equation (5.7). This procedure needs to be applied on all intervals $[t_i, t_{i+1}], i = 0, ..., N_x - 1$ to produce the sequence of $N_x + k$ degree k splines: $\Lambda_{-k}^k(x), ..., \Lambda_{N_x-1}^k(x)$.

B-spline properties

Some important properties of B-splines are listed in the following:

- Non-negative: $\Lambda_i^k(x) > 0$ in $t_i < x < t_{i+k+1}$
- Compact support for $\Lambda_i^k(x)$; contained in $[x_i, ..., x_{i+k+1}]$
- Piecewise polynomial of degree k

- Partition of unity: $\Sigma_{i=0}^{N_x-1}(x) = 1, \quad \forall x \in \mathbb{R}$
- Local linear independent
- C^{k-m} at knot x_i, with multiplicity $m = 1$ for all except boundary knots ($m = k+1$ at boundary)

B-spline derivative

The B-spline derivative of degree k can be written in the following way:

$$\frac{d}{dx}\Lambda_i^k(x) = k\left(\frac{\Lambda_i^{k-1}(x)}{t_{i+k} - t_i} - \frac{\Lambda_{i+1}^{k-1}(x)}{t_{i+k+1} - t_{i+1}}\right) \tag{5.9}$$

Hence, the derivative of a spline with degree k can directly be calculated by splines with degree $k-1$. This also leads to the C^{k-1} continuity of degree k splines.

Boundary conditions

For the generation of all splines on a finite domain $[t_0, t_{N_x}]$ via the recurrence (5.7), $N_x + k$ splines of degree k need to be constructed:

$$\Lambda_i^{-k}(x), \Lambda_i^{-k+1}(x), ..., \Lambda_i^{N_x-1}(x). \tag{5.10}$$

To generate all these splines, additional knots have to be defined beyond the ends of the interval $[t_0, t_{N_x}]$. There are generally two different options depending on the boundary conditions (BCs):

- **Periodic BCs:** additional knots are defined through periodicity:

$$\begin{aligned} t_{-\mu} &= t_{N_x-\mu} - (b-a) \\ t_{N_x+\mu} &= t_\mu + (b-a), \quad with\, \mu = 0, ..., k. \end{aligned} \tag{5.11}$$

  As a result, in eq. (5.10) the $k+1$ leftmost and rightmost splines are identical.

$$\Lambda_{-\mu}^k(x) = \Lambda_{N_x-\mu}^k(x) \tag{5.12}$$

- **Non-periodic BCs:** the choice we have taken in the non-periodic case is:

$$t_{-k} = ... = t_0 = a, \quad b = t_{N_x} = ... = t_{N_x+k} \tag{5.13}$$

The first spline $\Lambda^k_{-k}(x)$ in the first interval $[t_0, t_1]$ is thus constructed as:

$$\Lambda^k_{-k}(x) = \left(\frac{t_1 - x}{t_1 - t_0}\right)^k \Lambda^0_0, \tag{5.14}$$

and the last spline $\Lambda^k_{N_x-1}(x)$ in the last interval $[t_{N_x-1}, t_{N_x}]$ is constructed as:

$$\Lambda^k_{N_x-1}(x) = \left(\frac{x - t_{N_x-1}}{t_{N_x} - t_{N_x-1}}\right)^k \Lambda^0_{N_x-1}. \tag{5.15}$$

All non-periodic splines, except the first(last) one should vanish at $x = a$ ($x = b$), due to the fact that the sum of all splines is 1 and splines are non-negative:

$$\Lambda^k_j(a) = \delta_{j,-k}, \quad \Lambda^k_j(b) = \delta_{j,N_x-1}. \tag{5.16}$$

In figure 5.3, B-splines with degree $k = 1, 2, 3$ are displayed for the case of periodic boundary conditions in a domain with $N_x = 10$ equidistant grid cells. For comparison also the $k = 3$ B-spline for the non-periodic case is shown.

Spline expansion and ppform

Using a renumbering of spline index i, we can rewrite eq. (5.5) for the approximation of function $u(x)$ via B-splines to receive the so-called spline expansion:

$$u(x) \approx u_h(x) = \sum_{i=0}^{N+k-1} c_i \Lambda^k_i(x), \tag{5.17}$$

with support of $\Lambda^k_i(x)$ in $[t_{i-k}, t_{i+1}]$ and $t_i \leq t_{i+1} \rightarrow \Lambda^k_{i-k}(x), ..., \Lambda^k_i(x) \geq 0$.
Computing $u(x)$ according to equation (5.17) can be computationally costly, because of the spline evaluation. Therefore, by applying truncated Taylor series in each interval $[t_\nu, t_{\nu+1}]$ to expand $u(x)$, one can derive its Piecewise Polynomial Function representations (or **ppform**):

$$f(x) = \sum_{r=0}^{k} \Pi^k_{r\nu}(x - t_\nu)^r, \quad t_\nu \leq x \leq t_{\nu+1} \tag{5.18}$$

with

$$\Pi^k_{r\nu} = \frac{1}{r!}\frac{d^r}{dx^r}u(t_\nu) = \frac{1}{r!}\sum_j c_j \frac{d^r}{dx^r}\Lambda^k_j(t_\nu) = \sum_j c_j \alpha_{i\nu r}. \tag{5.19}$$

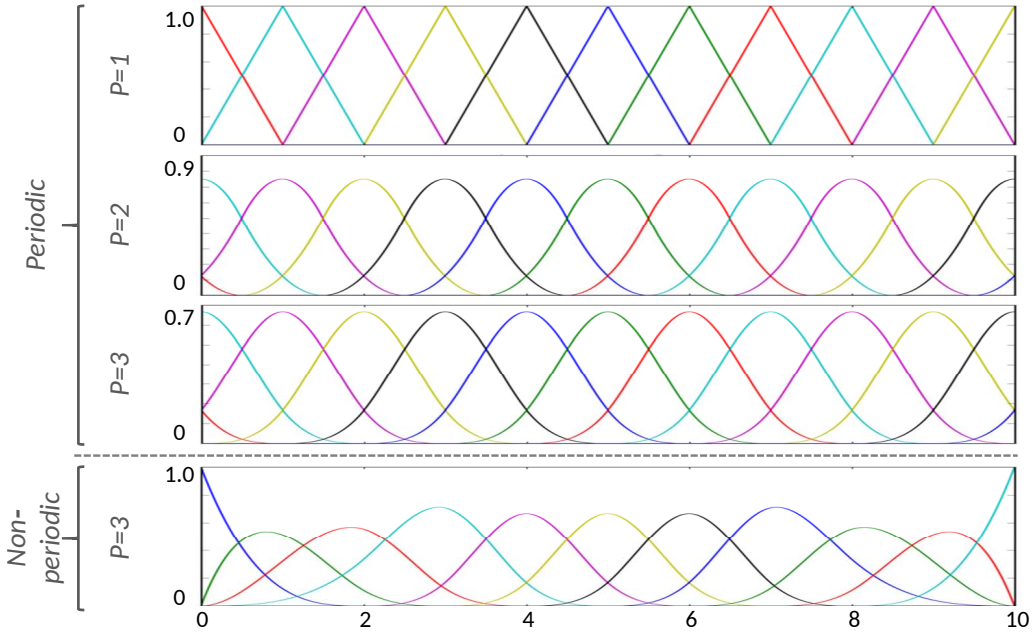


Figure 5.3: Representation of B-spline basis functions in a $N_x = 10$ equidistant grid. The upper three examples are B-splines of degree $p = 1, 2, 3$ for the case of periodic boundary conditions. For comparison, also the $k = 3$ B-spline for the non-periodic case is shown. Different B-spline functions are shown with different colors. Note that at every grid point, the sum of all B-splines at that point is exactly 1 (partition of unity).

It is important to notice that

$$\alpha_{i\nu r} = \frac{1}{r!}\frac{d^r}{dx^r}\Lambda_j^k(t_\nu) \tag{5.20}$$

only depends on spline specifications and thus can be pre-calculated. Eventually, only the spline coefficients are required to calculate $u(x)$ and its derivatives. In the code, we apply this ppform to reduce computational cost.

5.2.3 1D electric potential solver

A key task of the code is to solve the potential for a given particle distribution at each timestep and then calculate the electric field from this potential. In the following timestep particles are advanced with the help of this E-field. As mentioned in section 5.2, the potential is calculated based on the "Larmor point" distribution. After calculating the E-field values per "Larmor point", they are gyro-averaged to calculate their action on the gyrocenter. Therefore, the potential is calculated on specifically defined

field grid points and then the E-field is projected back to each marker position via the B-spline functions defined in 5.2.2.

Weak form

As shown in [102], to calculate the potential ϕ, we need to insert the discretized form of $f(x)$ from equation (5.1) into the polarization equation (2.40):

$$-\sum_p \nabla_\perp \frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \phi = \sum_p e_p \sum_{n=1}^{N} w_{p,n}(t) \delta(\mathbf{R_p} - \mathbf{R}_{p,n}) \tag{5.21}$$

with p the index for particle species and N the number of markers. We now multiply both sides with a test function ψ and integrate over the whole spatial domain V to get:

$$\int_V \psi \left(-\sum_p \nabla_\perp \frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \phi \right) \mathrm{d}\mathbf{R} = \sum_p e_p \sum_{n=1}^{N} w_{p,n}(t) \psi(\mathbf{R}_n). \tag{5.22}$$

By integrating by parts and applying the divergence theorem, the left-hand side can be simplified to:

$$\begin{aligned} & \int_V \psi (\sum_p \nabla_\perp (\frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \phi)) \mathrm{d}\mathbf{R} = \\ = \; & \int_V (\sum_p \nabla_\perp (\psi \frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \phi)) \mathrm{d}\mathbf{R} - \int_V (\sum_p \frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \psi \nabla_\perp \phi) \mathrm{d}\mathbf{R} \\ = \; & \underbrace{\int_{\partial V} (\sum_p \psi \frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \phi) \mathrm{d}\sigma}_{=0} - \int_V (\sum_p \frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \psi \nabla_\perp \phi) \mathrm{d}\mathbf{R}. \end{aligned} \tag{5.23}$$

Using that the integral over the domain boundary is zero and inserting this expression in equation (5.22) then yields:

$$\sum_p \int_V \frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \psi \nabla_\perp \phi \mathrm{d}\mathbf{R} = \sum_p e_p \sum_{n=1}^{N} w_{p,n}(t) \psi(\mathbf{R}_n). \tag{5.24}$$

Note that the same equation can be directly derived from the weak form of the polarization equation (2.36), by using an arbitrary test function for the potential variations $\delta\phi$.

Spline form

We now represent the potential ϕ with the B-spline basis functions defined in equation (5.17) to receive:

$$\phi(\mathbf{R}) = \sum_{\mu} \hat{\phi}_{\mu} \Lambda^k_{\mu}(\mathbf{R}), \tag{5.25}$$

where μ in the 1D case is a single index and ranges from 1 to the number of grid cells N_x ($\mu = 1, ..., N_x$). Whereas in the 3D case, $\mu = (i, j, r)$ is a tridimensional index, with $i = 1, ..., N_x$, $j = 1, ..., N_y$ and $r = 1, ..., N_z$. to represent the 3D spline $\Lambda^k_{i,j,r}(\mathbf{R}) = \Lambda^k_i(x)\Lambda^k_j(y)\Lambda^k_r(z)$. By also choosing $\psi(\mathbf{R}) = \Lambda^k_{\beta}(\mathbf{R})$ for the test function, we can rewrite eq. (5.24):

$$\sum_{\mu} \hat{\phi}_{\mu} \sum_{p} \int_V \frac{n_{p,0} m_p c^2}{B^2} \nabla_{\perp} \Lambda^k_{\beta}(\mathbf{R}) \nabla_{\perp} \Lambda^k_{\mu}(\mathbf{R}) \mathrm{d}\mathbf{R} = \sum_{p} e_p \sum_{n=1}^{N} w_{p,n}(t) \Lambda^k_{\beta}(\mathbf{R}_n). \tag{5.26}$$

By defining the matrix $P_{\mu\beta}$ and the right-hand-side ρ_{β} in the following way:

$$\begin{aligned} P_{\mu\beta} &= \hat{\phi}_{\mu} \sum_{p} \int_V \frac{n_{p,0} m_p c^2}{B^2} \nabla_{\perp} \Lambda^k_{\beta}(\mathbf{R}) \nabla_{\perp} \Lambda^k_{\mu}(\mathbf{R}) \mathrm{d}\mathbf{R}, \\ \rho_{\beta} &= \sum_{p} e_p \sum_{n=1}^{N} w_{p,n}(t) \Lambda^k_{\beta}(\mathbf{R}_n), \end{aligned} \tag{5.27}$$

we receive a system of equations of the form:

$$P\hat{\phi} = \rho. \tag{5.28}$$

For the calculation of both, P and ρ, the spline definitions from section 5.2.2 are essential. The so-called Poisson matrix P is not time-dependent and thus only needs to be calculated once at the beginning of the simulation. The right-hand-side ρ, however, needs to be recalculated at each step according to the particle positions and can be interpreted as the charge distribution of the particles. Once having calculated ρ, we can easily compute $\hat{\phi}$ from equation (5.28).

E-field

From the resulting potential, the electric field E can then be easily calculated from the relation:

$$E = -\nabla\phi. \tag{5.29}$$

Via B-spline functions, the E-field is eventually projected back from the potential/field grid on the particle positions to obtain the E-field at each particle position.

5.2.4 3D electric potential solver

Solving the 3D electric potential with finite elements in all three dimensions can be computationally extremely costly, depending on the number of grid cells in each direction (n_x, n_y, n_z). However, if the regarded problem is periodic in one of the three dimensions, a Fast Fourier Transform (FFT) can be applied to this periodic dimension. Depending on the size of the problem, this can reduce the computational cost of the simulation significantly. The remaining two dimensions can still be calculated based on a 2D finite element field solver. This is independent on whether those dimensions are periodic, or non-periodic. The electric field can then still be calculated for each cell of the 3D domain. In the following, we want to discuss the setup of such a so-called 2D1D solver, with an FFT in the periodic direction.

2D finite element solver for a 3D problem with 2D profiles

In 3D the electrostatic potential can be discretized with the finite-element method:

$$\phi(\boldsymbol{x}, t) = \sum_{w}^{N_{FE}} \phi_w(t)\widetilde{\Lambda}_w(\boldsymbol{x}) \ , \tag{5.30}$$

where from now on $\boldsymbol{x} = (x, z, y)$ for slab geometry and $\boldsymbol{x} = (r, \theta, \varphi)$ or $\boldsymbol{x} = (s, \theta, \varphi)$ for cylinder or screw pinch configurations. Here, s is the radius r of the cylinder normalized to its minor radius. We consider an axisymmetric geometry, where the equilibrium does not depend on one of the coordinates, typically y for slab, the toroidal angle φ for cylinder or screw pinch. The finite elements $\widetilde{\Lambda}_w$ are the tensor products of the usual B-splines $\Lambda_j(x)$, typically cubic:

$$\widetilde{\Lambda}_w(\boldsymbol{x}) = \Lambda_{wj'}(x)\Lambda_{wk'}(z)\Lambda_{wl'}(y) \ . \tag{5.31}$$

The tilde notations are used throughout this manuscript to indicate the tensor-product nature of both the finite elements and their indexes, illustrated by Eq. (5.31). Similarly,

we define:

$$\sum_{w}^{N_{FE}} = \sum_{j'} \sum_{k'} \sum_{l'} . \tag{5.32}$$

The integer indexes j, k, and l (or j', k', l') of the one-dimensional B-splines change from zero to the number of B-splines used in the respective direction. The electrostatic potential ϕ, can be represented in terms of the usual B-splines as follows:

$$\phi(x,z,y) = \sum_{l'=0}^{N_y-1} \sum_{j'k'} \phi_{j'k'l'}(t) \Lambda_{wj'}(x) \Lambda_{wk'}(z) \Lambda_{wl'}(y) \ , \tag{5.33}$$

with N_y being the number of the toroidal B-splines in y, the "axisymmentric" coordinate, assumed to be periodic. The number of splines in the other directions will depend on the choice of periodic or non-periodic splines. Therefore, the spline coefficients $\phi_{j'k'l'}$ can be Fourier transformed in the y coordinate. For simplicity, we assume that y is periodic in 2π. If not, a change of coordinates is required. Therefore, the Discrete-Fourier-Transform is used (see [139]):

$$\phi_{j'k'l'} = \sum_{n=0}^{N_y-1} \phi^{(n)}_{j'k'}(t) \ \exp\left[\frac{2\pi i}{N_y} \ nl'\right], \tag{5.34}$$

leading to

$$\phi(x,z,y) = \sum_{l'=0}^{N_y-1} \sum_{j'k'} \sum_{n=0}^{N_y-1} \phi^{(n)}_{j'k'}(t) \ \exp\left[\frac{2\pi i}{N_y} \ nl'\right] \Lambda_{wj'}(x) \Lambda_{wk'}(z) \Lambda_{wl'}(y). \tag{5.35}$$

The mass matrix problem

The mass matrix problem corresponds to the projection of a function ϕ on a spline basis. It is obtained by multiplying eq. (5.35) by a spline test function and integrating over the full space (the so-called weak formulation):

$$\begin{aligned} &\int d\boldsymbol{x} \ \phi(x,z,y) \widetilde{\Lambda}_t(\boldsymbol{x}) \\ &= \int d\boldsymbol{x} \left(\sum_{l'=0}^{N_y-1} \sum_{j'k'} \sum_{n=0}^{N_y-1} \phi^{(n)}_{j'k'}(t) \ \exp\left[\frac{2\pi i}{N_y} \ nl'\right] \Lambda_{wj'}(x) \Lambda_{wk'}(z) \Lambda_{wl'}(y) \right) \widetilde{\Lambda}_t(\boldsymbol{x}) \ . \end{aligned} \tag{5.36}$$

Where the test function $\widetilde{\Lambda}_t$ is once again the tensor products of the usual B-splines $\Lambda_j(x)$:

$$\widetilde{\Lambda}_t(\boldsymbol{x}) = \Lambda_{tj}(x)\Lambda_{tk}(z)\Lambda_{tl}(y) \ . \tag{5.37}$$

This is a matrix equation of the form:

$$\sum_w B_{wt}\phi_w(t) = c_t, \tag{5.38}$$

with

$$c_t = \int d\boldsymbol{x} \ \phi(x,z,y)\widetilde{\Lambda}_t(\boldsymbol{x}). \tag{5.39}$$

The Fast-Fourier-Transform is used (see [139]):

$$B_{wt} \ = \ \sum_{n=0}^{N_y-1} B_{wt}^{(n)} \ \exp\left[\frac{2\pi i}{N_y} \ nl\right], \tag{5.40}$$

$$c_t \ = \ \sum_{n=0}^{N_y-1} c_t^{(n)} \ \exp\left[\frac{2\pi i}{N_y} \ nl\right]. \tag{5.41}$$

For the Fourier coefficients $B_{jk}^{(n)}$, we have to compare eq. (5.36) and eq. (5.40):

$$\begin{aligned} B_{jkl} \ &= \ \sum_{n=0}^{N_y-1} B_{jk}^{(n)} \ \exp\left[\frac{2\pi i}{N_y} \ nl\right] \\ &= \ \sum_{n=0}^{N_\varphi-1} M^{(n)} \exp\left[\frac{2\pi i}{N_y} \ nl\right] \sum_{j'k'} \phi_{j'k'}^{(n)} \int \Lambda_{j'}(x)\Lambda_{k'}(z)\Lambda_j(x)\Lambda_k(z) \ J(x,z) \, dx \, dz. \end{aligned}$$

Thus, for the individual toroidal modes, we have to compute:

$$\begin{aligned} B_{jk}^{(n)} \ &= \ M^{(n)} \sum_{j'k'} \phi_{j'k'}^{(n)} \int \Lambda_{j'}(x)\Lambda_{k'}(z)\Lambda_j(x)\Lambda_k(z) \ J(x,z) \, dx \, dz \\ &= \ M^{(n)} \sum_{j'k'} \phi_{j'k'}^{(n)} b_{jkj'k'} \ . \end{aligned} \tag{5.42}$$

The matrix equation becomes, for a specific mode n:

$$\sum_{j'k'} \phi^{(n)}_{j'k'} b_{jkj'k'} = \frac{1}{M^{(n)}} c^{(n)}_{jk}. \tag{5.43}$$

This is constructed via the product of the Fourier transform of the spline coefficients, $\phi^{(n)}_{j'k'}$, the two-dimensional mass matrix,

$$b_{jkj'k'} = \int \Lambda_{j'}(x)\Lambda_{k'}(z)\Lambda_j(x)\Lambda_k(z)\, J(x,z)\, dx\, dz\ , \tag{5.44}$$

and the quantity $M^{(n)}$, defined by:

$$M^{(n)} \exp\left[\frac{2\pi i}{N_y}\, nl\right] = \sum_{l'=0}^{N_y-1} \int_0^{2\pi} dy\ \Lambda_{l'}(\varphi)\, \Lambda_l(\varphi)\ \exp\left[\frac{2\pi i}{N_\varphi}\, nl'\right]\ . \tag{5.45}$$

Calculation of matrix $M^{(n)}$ for slab geometry

By using the properties of the B-splines, we can rewrite eq. (5.45):

$$M^{(n)} \exp\left[\frac{2\pi i}{N_y}\, nl\right] = \exp\left[\frac{2\pi i}{N_y}\, nl\right] \sum_{k=-p}^{p} \int_0^{2\pi} dy\ \Lambda_{l+k}(y)\, \Lambda_l(y)\ \exp\left[\frac{2\pi i}{N_y}\, nk\right]$$

where p is the order of the B-splines. Hence, we obtain:

$$M^{(n)} = \sum_{k=-p}^{p} \exp\left[\frac{2\pi i}{N_y}\, nk\right] \int_0^{2\pi} dy\ \Lambda_{l+k}(y)\, \Lambda_l(y).$$

Now, we can define a new variable t, with $y = (2\pi/N_y)\, t$ to write:

$$m_k = \int_0^{2\pi} dy\ \Lambda_{l+k}(y)\, \Lambda_l(y)\ = \frac{2\pi}{N_y} \int_0^{N_y} dt\ \Lambda_{l+k}(t)\, \Lambda_l(t)\ . \tag{5.46}$$

Using the B-spline polynomials formulation of degree p defined within a grid cell, with the notation of [140], we can reformulate the expressions for m_k and $M^{(n)}$:

$$m_k = \frac{2\pi}{N_y} \int_0^{N_y} \Lambda_{l+k}(t)\, \Lambda_l(t)\, dt = \frac{2\pi}{N_y} \sum_{j=0}^{p-k} \int_0^1 P^{(p)}_{j+k}(t)\, P^{(p)}_j(t)\, dt \quad , \tag{5.47}$$

$$M^{(n)} = \sum_{k=-p}^{p} \exp\left[\frac{2\pi i}{N_y}\, nk\right] \; \frac{2\pi}{N_y} \sum_{j=0}^{p-k} \int_0^1 P_{j+k}^{(p)}(t)\, P_j^{(p)}(t)\, dt \quad .$$

By using the symmetry of the splines (and of the complex exponential) we then obtain:

$$M^{(n)} = \frac{2\pi}{N_y} \left(2 \sum_{k=1}^{p} \cos\left(\frac{2\pi}{N_y}\, nk\right) m_k + m_0 \right) \; ,$$
$$m_k = \sum_{j=0}^{p-k} \int_0^1 P_{j+k}^{(p)}(t)\, P_j^{(p)}(t)\, dt \; .$$

The coefficients can be calculated analytically. For the linear B-splines, $p = 1$, we thus can derive:

$$m_0 = \frac{2}{3} \; , \quad m_1 = \frac{1}{6} \; , \tag{5.48}$$

$$M^{(n)} = \frac{2\pi}{N_y} \left(\cos\left(\frac{2\pi}{N_y} n\right) + 2 \right) \frac{1}{3} \; .$$

For the quadratic ($p = 2$) B-splines holds:

$$m_0 = \frac{11}{20} \; , \quad m_1 = \frac{13}{60} \; , \quad m_2 = \frac{1}{120} \; , \tag{5.49}$$

$$M^{(n)} = \frac{2\pi}{N_y} \left(\frac{1}{60} \cos\left(\frac{2\pi}{N_y} 2n\right) + \frac{13}{30} \cos\left(\frac{2\pi}{N_y} n\right) + \frac{11}{20} \right) \; ,$$

which can be more conveniently rewritten by using:

$$\cos\left(\frac{2\pi}{N_y} 2n\right) = 2\cos^2\left(\frac{2\pi}{N_y} n\right) - 1. \tag{5.50}$$

Leading to:

$$M^{(n)} = \frac{2\pi}{N_y} \left(\frac{1}{30} \cos^2\left(\frac{2\pi}{N_y} n\right) + \frac{13}{30} \cos\left(\frac{2\pi}{N_y} n\right) + \frac{8}{15} \right) \; .$$

For the cubic ($p = 3$) B-splines we obtain:

$$m_0 = \frac{151}{315} \ , \quad m_1 = \frac{397}{1680} \ , \quad m_2 = \frac{1}{42} \ , \quad m_3 = \frac{1}{5040} \ , \tag{5.51}$$

$$M^{(n)} = \frac{2\pi}{N_y}\left(\frac{1}{2520}\cos\left(\frac{2\pi}{N_y}3n\right) + \frac{1}{21}\cos\left(\frac{2\pi}{N_y}2n\right) + \frac{397}{84}\cos\left(\frac{2\pi}{N_y}n\right) + \frac{151}{315}\right) \ ,$$

which again can be more conveniently rewritten by using eq. (5.50) and

$$\cos\left(\frac{2\pi}{N_y}3n\right) = 4\cos^3\left(\frac{2\pi}{N_y}n\right) - 3\cos\left(\frac{2\pi}{N_y}n\right) . \tag{5.52}$$

Leading to:

$$M^{(n)} = \frac{2\pi}{N_y}\left(\frac{1}{630}\cos^3\left(\frac{2\pi}{N_y}n\right) + \frac{2}{21}\cos^2\left(\frac{2\pi}{N_y}n\right) + \frac{33}{70}\cos\left(\frac{2\pi}{N_y}n\right) + \frac{136}{315}\right) \ .$$

2D1D Poisson solver in slab geometry

In PICLS, one general setup is a slab geometry, where the ignorable coordinate is now part of the perpendicular gradient. We can think of: z parallel direction, x radial direction, and y ignorable periodic direction. This also applies to cylindrical geometries, if the ignorable coordinate lies in the perpendicular plane (e.g., in θ-direction). The GK polarization (Poisson) equation is an elliptic equation given by:

$$-\nabla\cdot\left(\sum_{sp=\mathrm{i}}\frac{n_{sp}(x)m_{sp}c^2}{B(x,z)^2}\nabla_\perp\phi\right) = \sum_{sp=\mathrm{i,e}} q_{sp}n_{sp} \ . \tag{5.53}$$

The index sp corresponds to a sum on the different ion (i) and electron (e) species. Note that the polarization density, left-hand-side of eq. (5.53), does not contain electron contributions (drift-kinetic approximation for electrons). The gyro-averaged density is $n_{sp}(\boldsymbol{x}) = \int d^6Z\, f_{sp}\,\delta(\boldsymbol{R}+\boldsymbol{\rho}-\boldsymbol{x})$ (gyrocenter densities), $\rho_{sp} = \sqrt{m_{sp}T_{sp}}/(q_{sp}B)$ is the thermal gyroradius, q_{sp} is the charge of the particle, and $d^6Z = B_\parallel^*\, d\boldsymbol{R}\, dv_\parallel\, d\mu\, d\alpha$ is the phase-space volume, more details about the GK model can be found, for example, in [141]. The perpendicular plane in slab geometry is given by (x, y), leading to:

$$\nabla_\perp^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \tag{5.54}$$

The weak formulation of eq. (5.53) leads to the following Polarization equation:

$$-\int d\boldsymbol{x}\ \nabla\cdot\left(\sum_{sp=\mathrm{i}}\frac{n_{sp}(x)m_{sp}c^2}{B(x,z)^2}\nabla_\perp\phi\right)\widetilde{\Lambda}_t(\boldsymbol{x})\ =\int d\boldsymbol{x}\left(\sum_{sp=\mathrm{i,e}}q_{sp}n_{sp}\right)\widetilde{\Lambda}_t(\boldsymbol{x})\ . \quad (5.55)$$

Integrating it by parts leads to:

$$\sum_{sp=i}\int d\boldsymbol{x}\ c_{sp}(x,z)\nabla_\perp\phi\nabla_\perp\widetilde{\Lambda}_t(\boldsymbol{x})\ +\left(\int dydz\ c_{sp}(x,z)\frac{\partial}{\partial x}\phi\frac{\partial}{\partial x}\widetilde{\Lambda}_t(\boldsymbol{x})\right)\Bigg|_{x_{\min}}^{x_{\max}}$$
$$=\int d\boldsymbol{x}\left(\sum_{sp=i,e}q_{sp}n_{sp}\right)\widetilde{\Lambda}_t(\boldsymbol{x}), \quad (5.56)$$

with the definition:

$$c_{sp}(x,z)=\sum_{sp=\mathrm{i}}\frac{n_{sp}(x)m_{sp}c^2}{B(x,z)^2}.$$

The second term on the RHS of eq. (5.56) corresponds to the so-called natural boundary condition. Inserting eq. (5.35) into the Polarization equation we get, for the left-hand side:

$$B_{jkl}\ =\ \sum_{l'=0}^{N_{y-1}}\sum_{n=0}^{N_{y-1}}\exp\left[\frac{2\pi i}{N_y}\,nl'\right]\int_0^{2\pi}\frac{\partial}{\partial y}\Lambda_{l'}(y)\,\frac{\partial}{\partial y}\Lambda_l(y)\,dy \quad (5.57)$$
$$\sum_{j'k'}\phi^{(n)}_{j'k'}\int c_{sp}(x,z)\frac{\partial}{\partial x}\Lambda_{j'}(x)\Lambda_{k'}(z)\frac{\partial}{\partial x}\Lambda_j(x)\Lambda_k(z)\,J(x,z)\,dx\,dz$$

Here, the indexes j, k, l and n are integers. Again, we use the Fast-Fourier-Transform (see [139]) to obtain:

$$B_{jkl}\ =\ \sum_{n=0}^{N_y-1}B^{(n)}_{jk}\ \exp\left[\frac{2\pi i}{N_y}\,nl\right], \quad (5.58)$$

with:

$$c_t=\int d\boldsymbol{x}\left(\sum_{sp=i,e}q_{sp}n_{sp}\right)\widetilde{\Lambda}_t(\boldsymbol{x}). \quad (5.59)$$

For the Fourier coefficients $B_{jk}^{(n)}$, we have to compare eq. (5.57) and eq. (5.58) to get:

$$\sum_{n=0}^{N_y-1} B_{jk}^{(n)} \exp\left[\frac{2\pi i}{N_y}\, nl\right] = \sum_{n=0}^{N_y-1} DD^{(n)} \exp\left[\frac{2\pi i}{N_y}\, nl\right] \cdot$$
$$\sum_{j'k'} \phi_{j'k'}^{(n)} \int c_{sp}(x,z) \frac{\partial}{\partial x} \Lambda_{j'}(x) \Lambda_{k'}(z) \frac{\partial}{\partial x} \Lambda_j(x) \Lambda_k(z)\, J(x,z)\, dx\, dz.$$

Thus, for the individual toroidal modes, we have to compute:

$$B_{jk}^{(n)} = DD^{(n)} \sum_{j'k'} \phi_{j'k'}^{(n)} \int c_{sp}(x,z) \frac{\partial}{\partial x} \Lambda_{j'}(x) \Lambda_{k'}(z) \frac{\partial}{\partial x} \Lambda_j(x) \Lambda_k(z)\, J(x,z)\, dx\, dz\ .$$

The matrix equation becomes, for a specific mode n:

$$\sum_{j'k'} \phi_{j'k'}^{(n)} b_{jkj'k'} = \frac{1}{DD^{(n)}} c_{jk}^{(n)}. \tag{5.60}$$

Practically, it is convenient to solve this equation by applying the usual charge assignment, perform an FFT on the spline coefficients in y to get the Fourier components $c_{jk}^{(n)}$ and divide them by $DD^{(n)}$. The equation is constructed via the product of the Fourier transform of the spline coefficients, $\phi_{j'k'}^{(n)}$, the two-dimensional mass matrix,

$$b_{jkj'k'} = \int c_{sp}(x,z) \frac{\partial}{\partial x} \Lambda_{j'}(x) \Lambda_{k'}(z) \frac{\partial}{\partial x} \Lambda_j(x) \Lambda_k(z)\, J(x,z)\, dx\, dz\ , \tag{5.61}$$

and the quantity $DD^{(n)}$, defined by

$$DD^{(n)} \exp\left[\frac{2\pi i}{N_y}\, nl\right] = \sum_{l'=0}^{N_y-1} \int_0^{2\pi} dy\ \frac{\partial}{\partial y} \Lambda_{l'}(y)\ \frac{\partial}{\partial y} \Lambda_l(y)\ \exp\left[\frac{2\pi i}{N_y}\, nl'\right]\ . \tag{5.62}$$

Once the $\phi_{j'k'}^{(n)}$ Fourier coefficients are known, the electrostatic potential is given by eq. (5.35).

Calculation of matrix $DD^{(n)}$ for slab geometry

Similar to the derivation of $M^{(n)}$, by using the properties of the B-splines we obtain:

$$DD^{(n)} \exp\left[\frac{2\pi i}{N_y}\, nl\right] = \exp\left[\frac{2\pi i}{N_y}\, nl\right] \sum_{k=-p}^{p} \int_0^{2\pi} dy\ \frac{\partial \Lambda_{l+k}(y)}{\partial y}\ \frac{\partial \Lambda_l(y)}{\partial y}\ \exp\left[\frac{2\pi i}{N_y}\, nk\right],$$

where p is the order of the B-splines.
Therewith we can write:

$$DD^{(n)} = \sum_{k=-p}^{p} \exp\left[\frac{2\pi i}{N_y}\, nk\right] \int_0^{2\pi} dy\, \frac{\partial \Lambda_{l+k}(y)}{\partial y}\, \frac{\partial \Lambda_l(y)}{\partial y}.$$

Now, we can define a new variable t, $y = (2\pi/N_y)\, t$ to get:

$$dd_k = \int_0^{2\pi} dy\, \frac{\partial \Lambda_{l+k}(y)}{\partial y}\, \frac{\partial \Lambda_l(y)}{\partial y} \;= \frac{N_y}{2\pi} \int_0^{N_y} dt\, \frac{\partial \Lambda_{l+k}(t)}{\partial t}\, \frac{\partial \Lambda_l(t)}{\partial t}. \tag{5.63}$$

Using again the B-spline polynomials formulation of degree p defined within a grid cell with the notation of [140], we can rewrite dd_k and $DD^{(n)}$ as:

$$dd_k = \frac{N_y}{2\pi} \int_0^{N_y} \frac{\partial \Lambda_{l+k}(t)}{\partial t}\, \frac{\partial \Lambda_l(t)}{\partial} dt = \frac{N_y}{2\pi} \sum_{j=0}^{p-k} \int_0^1 \frac{\partial P_{j+k}^{(p)}(t)}{\partial t}\, \frac{\partial P_j^{(p)}(t)}{\partial t}\, dt\ , \tag{5.64}$$

$$DD^{(n)} = \sum_{k=-p}^{p} \exp\left[\frac{2\pi i}{N_y}\, nk\right]\ \frac{N_y}{2\pi} \sum_{j=0}^{p-k} \int_0^1 \frac{\partial P_{j+k}^{(p)}(t)}{\partial t}\, \frac{\partial P_j^{(p)}(t)}{\partial t}\, dt\ .$$

By using the symmetry of the splines (and of the complex exponential) we then obtain:

$$\begin{aligned} DD^{(n)} &= \frac{N_y}{2\pi} \left(2 \sum_{k=1}^{p} \cos\left(\frac{2\pi}{N_y}\, nk\right) dd_k + dd_0 \right)\ , \\ dd_k &= \sum_{j=0}^{p-k} \int_0^1 \frac{\partial P_{j+k}^{(p)}(t)}{\partial t}\, \frac{\partial P_j^{(p)}(t)}{\partial t}\, dt\ . \end{aligned}$$

Also here, the coefficients can be calculated analytically.
For the linear B-splines, $p = 1$, we can derive:

$$dd_0 = 2\ ,\quad dd_1 = -1\ , \tag{5.65}$$

$$DD^{(n)} = \frac{N_y}{2\pi} \left(-2 \cos\left(\frac{2\pi}{N_y} n\right) + 2 \right)\ .$$

For the quadratic ($p = 2$) B-splines holds:

$$dd_0 = 1 \ , \ \ dd_1 = -\frac{1}{3} \ , \ \ dd_2 = -\frac{1}{6} \ , \tag{5.66}$$

$$DD^{(n)} = \frac{N_y}{2\pi}\left(-\frac{1}{3}\cos\left(\frac{2\pi}{N_y}2n\right) - \frac{2}{3}\cos\left(\frac{2\pi}{N_y}n\right) + 1\right) \ ,$$

which can be more conveniently rewritten by using:

$$\cos\left(\frac{2\pi}{N_y}2n\right) = 2\cos^2\left(\frac{2\pi}{N_y}n\right) - 1. \tag{5.67}$$

Leading to:

$$DD^{(n)} = \frac{N_y}{2\pi}\left(-2\cos^2\left(\frac{2\pi}{N_y}n\right) - 2\cos\left(\frac{2\pi}{N_y}n\right) + 4\right)\frac{1}{3} \ .$$

For the cubic ($p = 3$) B-splines we obtain:

$$dd_0 = \frac{2}{3} \ , \ \ dd_1 = -\frac{1}{8} \ , \ \ dd_2 = -\frac{1}{5} \ , \ \ dd_3 = -\frac{1}{120} \ , \tag{5.68}$$

$$DD^{(n)} = \frac{N_y}{2\pi}\left(-\frac{1}{60}\cos\left(\frac{2\pi}{N_y}3n\right) - \frac{2}{5}\cos\left(\frac{2\pi}{N_y}2n\right) - \frac{1}{4}\cos\left(\frac{2\pi}{N_y}n\right) + \frac{2}{3}\right) \ ,$$

which again is more conveniently rewritten by using eq. (5.67) and

$$\cos\left(\frac{2\pi}{N_y}3n\right) = 4\cos^3\left(\frac{2\pi}{N_y}n\right) - 3\cos\left(\frac{2\pi}{N_y}n\right). \tag{5.69}$$

Leading to:

$$DD^{(n)} = \frac{N_y}{2\pi}\left(-\cos^3\left(\frac{2\pi}{N_y}n\right) - 12\cos^2\left(\frac{2\pi}{N_y}n\right) - 3\cos\left(\frac{2\pi}{N_y}n\right) + 16\right)\frac{1}{15} \ .$$

Note that the $DD^{(n)}$ matrix vanishes for $n = 0$. A special treatment is therefore required for the $n = 0$ mode in slab geometry.

2D1D Poisson solver in tokamak geometries

The same exact procedure as described in the previous section can be applied to the polarization equation:

$$-\nabla \cdot \left[\left(\sum_{s=\mathrm{i}} \frac{q_\mathrm{s}^2 n_s}{T_s} \rho_s^2 \right) \nabla_\perp \phi \right] = \sum_{s=\mathrm{i,e}} q_\mathrm{s} n_{1s}, \tag{5.70}$$

where $n_{1s} = \int d^6Z\, f_s\, \delta(\boldsymbol{R} + \boldsymbol{\rho} - \boldsymbol{x})$ is the gyrocenter density, $\rho_s = \sqrt{m_s T_s}/(q_\mathrm{s} B)$ is the thermal gyroradius, q_s is the charge of the particle, and $d^6Z = B_\parallel^*\, d\boldsymbol{R}\, dv_\parallel\, d\mu\, d\alpha$ is the phase-space volume. The weak formulation, "Poisson Matrix", after integration by parts, leads to the usual polarization equation. The right-hand side (charge density) can thus be written as:

$$b_{jkl} \;=\; \int \left(\sum_{s=i,e} q_\mathrm{s} n_{1s} \right) (s,\theta,\varphi) \Lambda_j(s) \Lambda_k(\theta) \Lambda_l(\varphi)\, J(s,\theta)\, ds\, d\theta\, d\varphi. \tag{5.71}$$

Whereas, for the left-hand side (GK Poisson/Polarization equation) we get:

$$B_{jkl} = \int N(s,\theta)\, \nabla_\perp \phi(s,\theta,\varphi) \nabla_\perp (\Lambda_j(s)\Lambda_k(\theta)) \Lambda_l(\varphi)\, J(s,\theta)\, ds\, d\theta\, d\varphi, \tag{5.72}$$

with:

$$N(s,\theta) = \sum_{s=\mathrm{i}} \frac{q_\mathrm{s}^2 n_s}{T_s} \rho_s^2. \tag{5.73}$$

Note that it is crucial that N does not depend on φ. This could be a problem for the nonlinear Poisson case, but for the moment we can deal with it using a linear Poisson equation.
We can now use the Fourier representation of eq. (5.35) to write:

$$\begin{aligned} B_{jkl} \;=\; & \sum_{l'=0}^{N_\varphi - 1} \sum_{n=0}^{N_\varphi - 1} \exp\left[\frac{2\pi i}{N_\varphi}\, nl' \right] \int_0^{2\pi} \Lambda_{l'}(\varphi)\, \Lambda_l(\varphi)\, d\varphi \sum_{j'k'} \phi_{j'k'}^{(n)}(t) \\ & \int N(s,\theta) \nabla_\perp (\Lambda_{j'}(s)\Lambda_{k'}(\theta)) \nabla_\perp (\Lambda_j(s)\Lambda_k(\theta))\, J(s,\theta)\, ds\, d\theta. \end{aligned} \tag{5.74}$$

The Poisson equation in matrix form, $B_{jkl} = b_{jkl}$, with fast Fourier is:

$$\sum_{n=0}^{N_\varphi - 1} B_{jk}^{(n)} \exp\left[\frac{2\pi i}{N_\varphi} nl\right] = \sum_{n=0}^{N_\varphi - 1} b_{jk}^{(n)} \exp\left[\frac{2\pi i}{N_\varphi} nl\right]. \tag{5.75}$$

This means that for individual toroidal modes we have to solve:

$$B_{jk}^{(n)} = M^{(n)} \sum_{j'k'} \phi_{j'k'}^{(n)} \int N(s,\theta) \nabla_\perp (\Lambda_{j'}(s)\Lambda_{k'}(\theta)) \nabla_\perp (\Lambda_j(s)\Lambda_k(\theta))\, J(s,\theta)\, ds\, d\theta = b_{jk}^{(n)},$$

with the same $M^{(n)}$ calculated previously for the mass matrix case. Practically, it is more convenient to solve:

$$\sum_{j'k'} \phi_{j'k'}^{(n)} \int N(s,\theta) \nabla_\perp (\Lambda_{j'}(s)\Lambda_{k'}(\theta)) \nabla_\perp (\Lambda_j(s)\Lambda_k(\theta))\, J(s,\theta)\, ds\, d\theta = \frac{b_{jk}^{(n)}}{M^{(n)}}. \tag{5.76}$$

This is done by performing the usual charge assignment, then performing an FFT on the spline coefficients in φ to get the Fourier components $b_{jk}^{(n)}$ and dividing them by $M^{(n)}$. Meanwhile, the 2D matrix

$$A_{j'k'jk} = \int N(s,\theta) \nabla_\perp (\Lambda_{j'}(s)\Lambda_{k'}(\theta)) \nabla_\perp (\Lambda_j(s)\Lambda_k(\theta))\, J(s,\theta)\, ds\, d\theta \tag{5.77}$$

has to be constructed (as usual). Note that the matrix does not depend on φ. Finally a set of N_φ matrix equations, one for each toroidal mode n must be solved (Lapack):

$$\sum_{j'k'} \phi_{j'k'}^{(n)} A_{j'k'jk} = \frac{b_{jk}^{(n)}}{M^{(n)}}. \tag{5.78}$$

Once the $\phi_{j'k'}^{(n)}$ Fourier coefficients are known, the electrostatic potential is given by eq. (5.35).

To get a better understanding of the structure of the 2D matrix $A_{j'k'jk}$ we want to refer to figure 5.4. Here, the logarithm of the absolute values of the matrix elements of an exemplary $A_{j'k'jk}$ matrix is plotted without boundary conditions applied to the matrix. The logarithm was chosen to better distinguish between the values of the individual elements. The matrix has 121×121 elements and corresponds to a field grid with $n_x \times n_y = 8 \times 8$ with a spline degree of $k = 3$. The matrix clearly exhibits a banded structure with $1 + 2k$ bands. With non-periodic boundary conditions (see section 5.2.2) applied to the matrix in the first direction, several matrix elements have to be adjusted. The result can be seen in figure 5.5. As visible, always the first and last cells of the

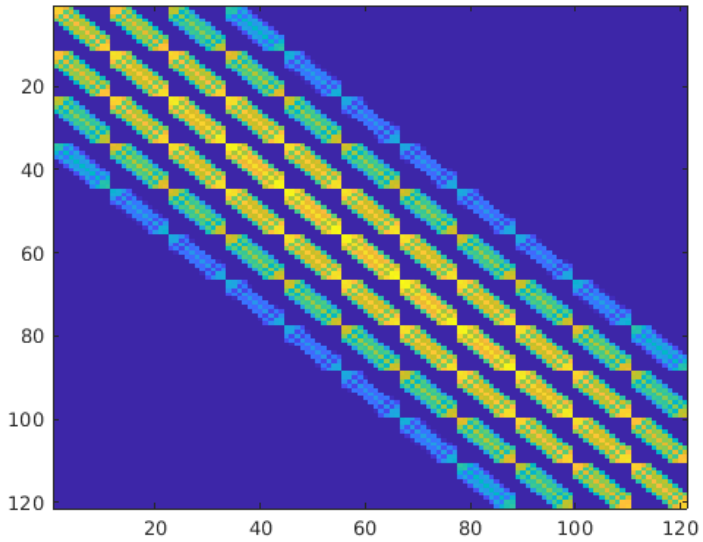


Figure 5.4: Plot of the logarithm of the absolute values of the matrix elements of the $A_{j'k'jk}$ matrix, without boundary conditions applied to the matrix. The matrix has 121×121 elements and corresponds to a field grid with $n_x \times n_y = 8 \times 8$ with a spline degree of $k = 3$.

matrix blocks are affected. The reason for this is that the boundary conditions were chosen to be on the first index of the matrix (x-direction). A matrix with boundary conditions in the y-direction exhibits a different structure. For an even clearer picture, the difference of the two matrices is plotted in figure 5.6. To be more specific, what is actually plotted is the logarithm of the differences of the absolute values of the matrix elements of the two matrices shown in figure 5.4 and figure 5.5. The matrix shown here are only exemplary for the tokamak case, but the same principle applies to the 2D matrix in the slab case.

Calculation of matrix $M^{(n)}$ for tokamaks

Similar to the previous calculation of the matrix $M^{(n)}$, by using the properties of the B-splines, we can write:

$$M^{(n)} \exp\left[\frac{2\pi i}{N_\varphi}\, nl\right] = \exp\left[\frac{2\pi i}{N_\varphi}\, nl\right] \sum_{k=-p}^{p} \int_0^{2\pi} d\varphi\, \Lambda_{l+k}(\varphi)\, \Lambda_l(\varphi)\, \exp\left[\frac{2\pi i}{N_\varphi}\, nk\right].$$

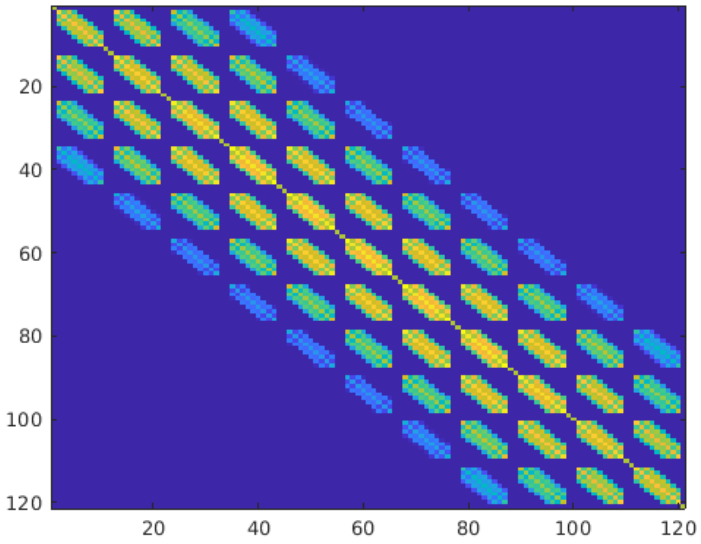


Figure 5.5: Plot of the logarithm of the absolute values of the matrix elements of the $A_{j'k'jk}$ matrix (compare to figure 5.4), with boundary conditions applied to the first index of the matrix. The matrix has 121×121 elements and corresponds to a field grid with $n_x \times n_y = 8 \times 8$ with a spline degree of $k = 3$.

Here, p is the order of the B-splines. Hence, we achieve:

$$M^{(n)} = \sum_{k=-p}^{p} \exp\left[\frac{2\pi i}{N_\varphi}\, nk\right] \int_0^{2\pi} d\varphi\, \Lambda_{l+k}(\varphi)\, \Lambda_l(\varphi).$$

Now, we define a new variable t, $\varphi = (2\pi/N_\varphi)\, t$ to obtain:

$$m_k = \int_0^{2\pi} d\varphi\, \Lambda_{l+k}(\varphi)\, \Lambda_l(\varphi) \;= \frac{2\pi}{N_\varphi} \int_0^{N_\varphi} dt\, \Lambda_{l+k}(t)\, \Lambda_l(t). \tag{5.79}$$

Using the B-spline polynomials formulation of degree p defined within a grid cell, with the notation of [140], m_k and $M^{(n)}$ can be reformulated as:

$$m_k = \frac{2\pi}{N_\varphi} \int_0^{N_\varphi} \Lambda_{l+k}(t)\, \Lambda_l(t)\, dt = \frac{2\pi}{N_\varphi} \sum_{j=0}^{p-k} \int_0^1 P_{j+k}^{(p)}(t)\, P_j^{(p)}(t)\, dt\ , \tag{5.80}$$

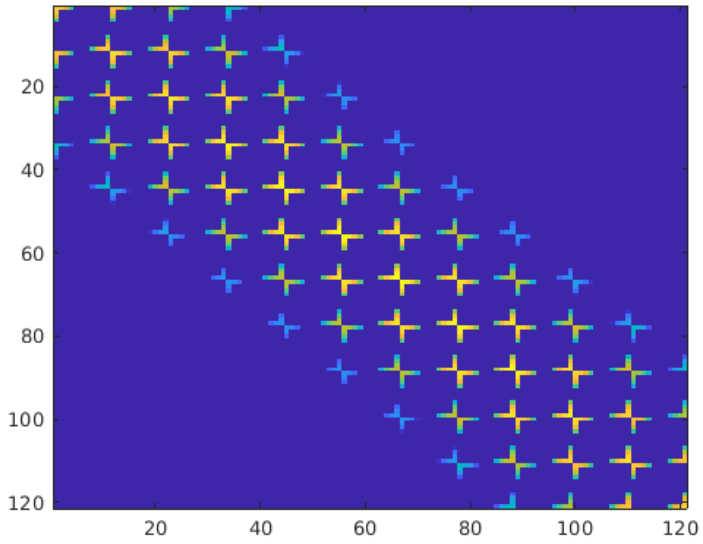


Figure 5.6: Plot of the logarithm of the differences of the absolute values of the matrix elements of the two matrices shown in figure 5.4 and figure 5.5.

$$M^{(n)} = \sum_{k=-p}^{p} \exp\left[\frac{2\pi i}{N_\varphi}\, nk\right] \frac{2\pi}{N_\varphi} \sum_{j=0}^{p-k} \int_0^1 P_{j+k}^{(p)}(t)\, P_j^{(p)}(t)\, dt \ .$$

By using the symmetry of the splines (and of the complex exponential) we can then write:

$$M^{(n)} = \frac{2\pi}{N_\varphi} \left(2 \sum_{k=1}^{p} \cos\left(\frac{2\pi}{N_\varphi}\, nk\right) m_k + m_0 \right) \ ,$$

$$m_k = \sum_{j=0}^{p-k} \int_0^1 P_{j+k}^{(p)}(t)\, P_j^{(p)}(t)\, dt \ .$$

The coefficients again can be calculated analytically. For the linear B-splines, $p = 1$, we thus obtain:

$$m_0 = \frac{2}{3} \ , \quad m_1 = \frac{1}{6} \ , \tag{5.81}$$

$$M^{(n)} = \frac{2\pi}{N_\varphi} \left(\cos\left(\frac{2\pi}{N_\varphi} n\right) + 2 \right) \frac{1}{3} \ .$$

For the quadratic ($p = 2$) B-splines holds:

$$m_0 = \frac{11}{20} \ , \quad m_1 = \frac{13}{60} \ , \quad m_2 = \frac{1}{120} \ , \tag{5.82}$$

$$M^{(n)} = \frac{2\pi}{N_\varphi} \left(\frac{1}{60} \cos\left(\frac{2\pi}{N_\varphi} 2n \right) + \frac{13}{30} \cos\left(\frac{2\pi}{N_\varphi} n \right) + \frac{11}{20} \right) \ ,$$

which is more conveniently rewritten by using:

$$\cos\left(\frac{2\pi}{N_\varphi} 2n \right) = 2\cos^2\left(\frac{2\pi}{N_\varphi} n \right) - 1. \tag{5.83}$$

Leading to:

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

For the cubic ($p = 3$) B-splines we can derive:

$$m_0 = \frac{151}{315} \ , \quad m_1 = \frac{397}{1680} \ , \quad m_2 = \frac{1}{42} \ , \quad m_3 = \frac{1}{5040} \ , \tag{5.84}$$

$$M^{(n)} = \frac{2\pi}{N_\varphi} \left(\frac{1}{2520} \cos\left(\frac{2\pi}{N_\varphi} 3n \right) + \frac{1}{21} \cos\left(\frac{2\pi}{N_\varphi} 2n \right) + \frac{397}{84} \cos\left(\frac{2\pi}{N_\varphi} n \right) + \frac{151}{315} \right) \ ,$$

which is more conveniently rewritten by using Eq. (5.83) and

$$\cos\left(\frac{2\pi}{N_\varphi} 3n \right) = 4\cos^3\left(\frac{2\pi}{N_\varphi} n \right) - 3\cos\left(\frac{2\pi}{N_\varphi} n \right). \tag{5.85}$$

Leading to:

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

Special case in slab geometry, n=0

Let us define a $\bar{\phi} = \frac{1}{L_y} \int_0^{L_y} \phi \ \mathrm{d}y$, equivalent to the $n = 0$ component of the Fourier Transform of ϕ, $\phi^{(0)}$ (with periodicity L_y). Taking the integral in y of the polarization

equation, the polarization equation for $\bar{\phi}$ reads:

$$-\nabla \cdot c_{sp}(x,z)\nabla_{\perp}\bar{\phi} = \sum_{sp=i,e} q_{sp}\bar{n}_{sp}. \tag{5.86}$$

Since none of the terms depends on y anymore, all the terms containing a derivative in y are identically zero. Therefore, the perpendicular gradient reduces to:

$$\nabla_{\perp} = \frac{\partial}{\partial x}\mathbf{e_x}. \tag{5.87}$$

We can now use the Fourier representation of Eq. (5.35) for $n = 0$:

$$\begin{aligned} B_{jkl} \;=\; & \sum_{l'=0}^{N_y-1}\sum_{n=0}^{N_y-1}\int_0^{2\pi}\Lambda_{l'}\Lambda_l(y)\,dy\sum_{j'k'}\phi^{(n)}_{j'k'}(t) \\ & \int c_{sp}(x,z)\frac{\partial\Lambda_{j'}(x)}{\partial x}\Lambda_{k'}(z)\frac{\partial\Lambda_j(x)}{\partial x}\Lambda_{k'}(z)\,J(x,z)\,dx\,dz. \end{aligned} \tag{5.88}$$

The Poisson equation in matrix form, $B_{jkl} = b_{jkl}$, with fast Fourier Transformation for $n = 0$ can be written as:

$$B^{(0)}_{jk} = MM^{(0)}\sum_{j'k'}\phi^{(0)}_{j'k'}\int c_{sp}(x,z)\frac{\partial\Lambda_{j'}(x)}{\partial x}\Lambda_{k'}(z)\frac{\partial\Lambda_j(x)}{\partial x}\Lambda_k(z)\,J(x,z)\,dx\,dz = b^{(0)}_{jk},$$

where we still have to calculate the new coefficients $MM^{(0)}$ as shown previously. It is more convenient to solve:

$$\phi^{(0)}_{j'k'}\int c_{sp}(x,z)\frac{\partial\Lambda_{j'}(x)}{\partial x}\Lambda_{k'}(z)\frac{\partial\Lambda_j(x)}{\partial x}\Lambda_k(z)\,J(x,z)\,dx\,dz = \frac{b^{(0)}_{jk}}{MM^{(0)}}.$$

Which again is done by performing the usual charge assignment, then applying an FFT on the spline coefficients in φ to get the Fourier components $b^{(0)}_{jk}$ and dividing them by $MM^{(0)} = 2\pi/N_y$ (after having performed the usual change of variable from $y = [0 : L_y]$ to $\hat{y} = [0 : 2\pi]$, using the factor $2\pi/L_y$).

Fourier filter

In summary, the main steps of the PIC solver to get the electric potential ϕ_{jkl} from the right-hand side (or charge density) b_{jkl} can be written in the following way:

$$b_{jkl} \xrightarrow{\text{DFT}} b^{(n)}_{jk} \xrightarrow{\text{backsolve}} \phi^{(n)}_{jk} \xrightarrow{\text{IDFT}} \phi_{jkl}, \tag{5.89}$$

where DFT stands for the discrete Fourier transform and IDFT for its inverse operation. Solving for the Fourier transformed potential $\phi_{jk}^{(n)}$ from $b_{jk}^{(n)}$ is called "backsolve". Statistical noise which is created by the discretization of markers can now be filtered by a numerical trick, where the density Fourier modes that are not physical are filtered. Therefore, this Fourier filtering needs to be applied after the right hand side has been Fourier transformed, at the beginning of the "backsolve". These non-physical modes correspond to high frequency components in the FFT-direction. These pre-conditions can be formulated as:

$$|n| \leq n_{\max}. \tag{5.90}$$

To avoid truncation of physical modes, it is important to choose $n_{\max}$ carefully.

Geometries

With the previously described tokamak and slab 2D1D solvers several geometries can be studied. As discussed, the main difference between the two solver types is that once the FFT-direction lies along the main B-field direction and once it is directed perpendicular to the B-field. Using this property, we define three different geometries that are implemented in PICLS (see also figure 5.7 for illustration):

1. **Cylinder:** In this case, the geometry is cylindrical with the coordinates (s, θ, z), where s is the radius r of the cylinder normalized to the minor radius a, θ the poloidal angle and z goes in the "toroidal" direction. The cylinder geometry can also be used as cylindrical tokamak, for which z is replaced by the toroidal angle ϕ. In this case, the magnetic field B has only a component in z direction and also the FFT direction is along z. Hence, for solving the field equation of this geometry, equation (5.78) has to be used.

2. **Pinch:** This geometry in general is similar to the "Cylinder" case, except that the B-field can have an additional component in the poloidal θ direction.

3. **Slab**: Here, we use a 3D slab geometry, with the spatial coordinates (x, y, z). The magnetic field is aligned along the z coordinate, but different from the "Cylinder" and "Pinch" case, the FFT direction is along the perpendicular y direction. The choice of y as FFT-coordinate here is arbitrary, x would be equally viable. To solve the field equation for this setup, equation (5.60) has to be used.

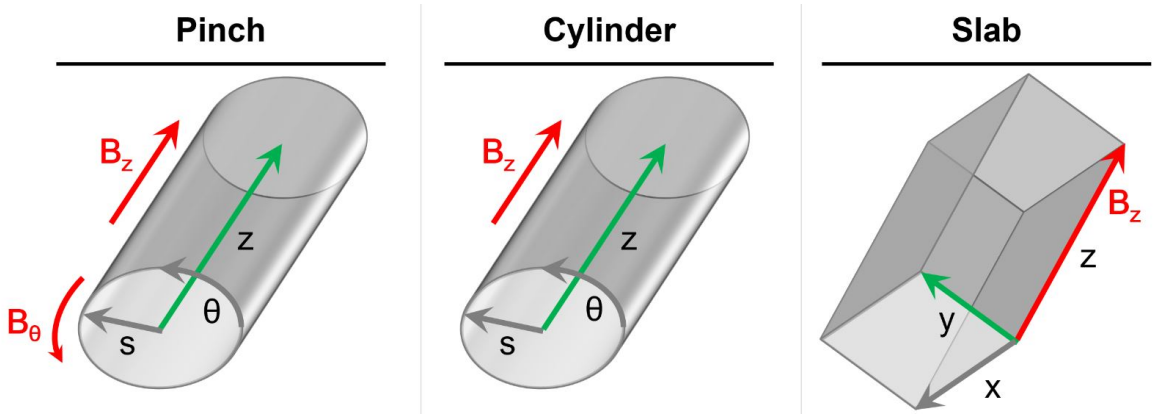


Figure 5.7: Illustration of the three different geometries "Pinch", "Cylinder", and "Slab" that are currently implemented in PICLS. "Pinch" and "Cylinder" are implemented in cylindrical coordinates (s, θ, z), whereas "Slab" uses (x, y, z) coordinates. The red arrows indicate the direction of the B-field of the respective geometry. The green arrows indicate the direction of the coordinate for the FFT of the electric field solver.

5.3 Data structures

Our code is generally based on a data structure centric approach, which was chosen to guarantee a smooth transition to GPU architectures in the future. In the following the main data types we have implemented are shown for the 3D case. In the 1D case the data types are generally the same but smaller, due to the reduced dimensionality. The main types are:

- **Marker attributes:** A structure of arrays for the marker attributes (called `particle_att`) of size $25 \times N$, where N is the number of markers. Each array represents a specific attribute of the markers and contains N entries for all marker positions. The stored attributes are the marker's position in 3D space, parallel and perpendicular velocity, weight, electric field, etc. The 5 attributes for position in 3D, parallel velocity and weight are replicated three times. One time to store the system state at the beginning of a timestep, one time to compute their derivatives and to do the virtual Runge-Kutta step, and one time to update the effective displacement.

- **Poisson matrix:** The time-independent band matrix $A_{j'k'jk}$ is calculated at the beginning and stored for the rest of the simulation (called `poisson_matrix`, or `mass_matrix`). Only the bands of the matrix have to be stored to decrease memory requirements.

- **RHS and potential:** To store the right-hand side and the potential for each time step, 6 $n_x \times n_y \times n_z$ arrays need to be calculated. The actual RHS and potential, as well as their Fourier transforms (with complex type) and transposed copies of both needed by the parallelization scheme have to be stored therefore.

5.4 Parallelization

For the parallelization scheme the two main PIC principles of calculating particle trajectories and solving the field equations need to be taken into account. Both require a specific parallel treatment to save computational cost and thus will be discussed in the following.

5.4.1 Parallelization: particles

Running high resolution plasma simulations with a gyrokinetic PIC code, in general means that a significant number of markers and grid cells is required. However, the more markers and cells we introduce, the higher the computational costs become. As a result, an efficient parallelization scheme needs to be implemented, to share data between multiple processors to join their computation resources and to optimize the memory use.
In PICLS, we use a hybrid OpenMP and MPI approach. The specific methods we apply for the parallelization of the MPI tasks are called domain decomposition and domain cloning.
In the domain decomposition approach, the grid cells are split up into different domains. Each domain is then contributed to a MPI task and also the particles that are present within the cell at the specific timestep are contributed to this task. For our domain decomposition, we split the cells in the physical y-direction, along the periodic FFT-direction.
In the domain cloning approach, however, the number of particles is divided between the number of defined MPI clones. Each clone has a copy of the whole domain to deposit the charges. The sums of the charges of each clone are then added up via MPI communication to calculate the fields for the total particle distribution.
Each MPI task eventually is responsible for a specific clone and a specific domain. Within each of these MPI tasks, the particle operations are performed in parallel and OpenMP is applied, to optimally use computational resources. Since the particles can move from one domain to the other, based on the forces acting on it, a function "particle_move" is implemented that moves the particles from the exiting to the entering MPI domain for the calculations. In figure 5.8 this parallelization scheme is sketched.

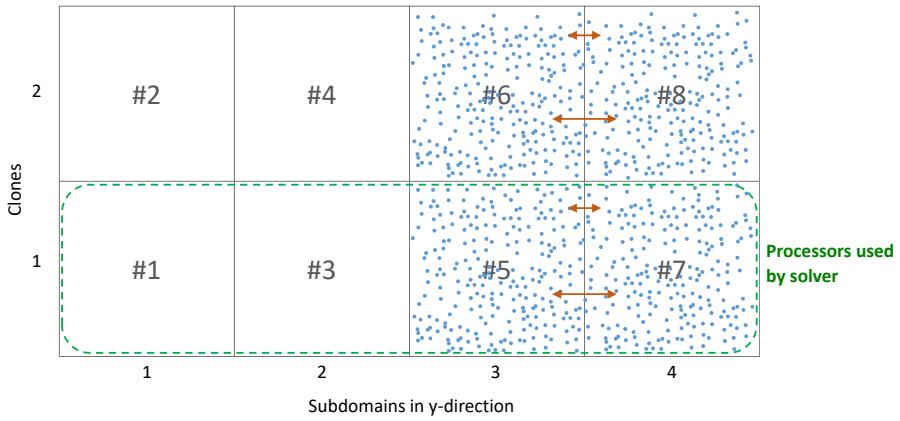


Figure 5.8: Illustration of the PICLS parallelization scheme with 4 domains and two clones. The blue dots represent the particles and the orange arrows the communication between the MPI domains. In the example, the communication is displayed exemplary for domains 3 & 4 and clones 1 & 2. With the green box we indicated to which processors the particle sums are reduced for the field solve. The total number of processors (indicated by #) is equal to the number of domains times the number of clones.

5.4.2 Parallelization: field solver

As mentioned in section 5.4.1, MPI domain decomposition and domain cloning are applied and within each domain the total charge distribution (right-hand side) is calculated and sent to the first clone. For more detailed information on domain cloning, we refer to [142] Some of the spline coefficients are common to several subdomains, since splines of order p are $\neq 0$ on $p+1$ intervals. These spline values that are contributing to different subdomains have to be stored in so-called guard cells. At the end of the charge deposition step, all p guard cells of a subdomain are sent to its neighbors and added up there.
To execute the Fourier transforms on the ignorable dimension, the so-called parallel data transpose has to be applied, where the partitioning of the RHS is switched with another dimension. In general, a parallel data transpose is always performed before and after each Fourier transform. How the partitioning is done during the different steps of the solver can be explained by the following workflow, where y stands for the

ignorable coordinate:

$$\texttt{rhs}(i,j,[k]) \xrightarrow{\text{remove guard cells}} \texttt{rhs}(i,j,[k[) \xrightarrow{\text{data transpose}} \texttt{rhs}(j,i,[k[)$$
$$\xrightarrow{\text{DFT in } y} \texttt{rhs}(n,i,[k[) \xrightarrow{n \text{ filter}} \texttt{rhs}(n,i,[k[) \xrightarrow{\text{data transpose}} \texttt{rhs}(i,n,[k[)$$
$$\xrightarrow{\text{backsolve}} \texttt{phi}(i,n,[k[) \xrightarrow{\text{data transpose}} \texttt{phi}(n,i,[k[) \xrightarrow{\text{IDFT in } y} \texttt{phi}(j,i,[k[)$$
$$\xrightarrow{\text{data transpose}} \texttt{phi}(i,j,[k[) \xrightarrow{\text{add guard cells}} \texttt{phi}(i,j,[k])$$

Here, (I)DFT is the (Inverse) Discrete Fourier Transform and the brackets indicate the partitioned dimension, with (or without) guard cells if they are closed (or open). Note that the n filter step includes the division of the RHS by the $M^{(n)}$ or $DD^{(n)}$ matrix as explained in section 5.2.4.

5.5 Scaling tests

To get an indication for the computational efficiency of the code implementation we will in the following discuss the strong scaling with OpenMP of PICLS in a more or less realistic test case. We do not want to provide a very detailed analysis of the codes speed up, but only want to show its key characteristics for a given test case. The strong scaling can be identified by fixing a specific test case and increasing the number of processors for the execution. With the increasing number of processors, the speed-up of the execution time is measured to achieve the speed-up per additional processor. The problem size is not changed to get strong scaling. Due to its hybrid OpenMP/MPI parallelization, we will also consider the scaling of the MPI parallelization separately. However, here we will increase the problem size together with the number of nodes. We measure the time elapsed for 100 time steps, once the system has overcome the initial start up phase of the simulation.
The OpenMP test case is designed to represent the actual open field line simulations we performed in three-dimensional slab and cylindrical setups so far. Therefore, we choose a electron deuterium plasma with a reduced mass ratio of $m_\mathrm{e}/m_\mathrm{i} = 1/400$, a domain size of $(l_x, l_y, l_z = 1m, 1m, 10m)$, 8 cells in each dimension and ~ 1000 particles per cell. For the MPI test case, we increase the number of cells and the length in the y-direction together with the number of nodes to maintain the number of particles per cell. The simulation domain is periodic in each direction, which means that no sheath boundary routines are measured. However, due to the generally low number of particles crossing the sheath within our simulations the time consumption of this routine is not critical. In addition, we do not apply sheath BCs in all of our simulations and therefore their impact would not be meaningful for parts of our runs. Only drift-kinetic particles,

without gyro-rings are considered and collisions are introduced via a Lenard-Bernstein collision operator (see 4.3). The diagnostics routines are not considered, since they are not part of each time step, but only applied after several cycles. The B-field is chosen to be constant with a z-component of $B_z = 2.0\text{T}$ only.
The scaling tests are performed on the MARCONI Eurofusion HPC system, which consists of $2 \times 24-$cores Intel Xeon 8160 CPUs (Skylake) at 2.10 GHz.
For the future, we plan to port our code to GPUs to further exploit modern HPC architectures. This could be of specific interest, since in the current code version three routines are the most time consuming: the charge deposition, the mapping of the E-field to the markers and the particle pushing. These routines account for $> 90\%$ of the elapsed computational time with similar shares for each of them. With the help of GPUs and further code optimization this is expected to decrease significantly.

5.5.1 OpenMP scaling / intra-node

For the OpenMP scaling, only one single node is used and only one single MPI task. Starting from one single OpenMP process, the scaling test is performed for up to 48 processes. In figure 5.9 we therefore plot the speed up against the number of used OpenMP threads N_{omp}. The speed up is defined as $t_1/t_{N_{\text{omp}}}$, where t_1 is the elapsed time with only one core and $t_{N_{\text{omp}}}$ the time elapsed with N_{omp} cores. The blue line shows the speed up for OpenMP only and already with ~ 16 cores is well below the optimal speed up of $t_1/t_{N_{\text{omp}}} = N_{\text{omp}}$ shown by the grey line. This mainly comes from the too small problem size for the increasing number of OpenMP processes. By introducing for example 8 MPI tasks introduced by clones with 6 OpenMP processes for each of these tasks the speed up can be increased significantly. In the graph this is shown by the red point for 48 processes, which is close to the optimal speed up.
From figure 5.9, we can see that the problem is obviously not perfect for our test, but we already get a good indication, that the full node can be exploited with a good speed up, when the OpenMP and MPI parallelization is applied correctly.

5.5.2 MPI scaling / inter-node

To test the MPI scaling with several nodes, we now fix the number of MPI clones (see 5.4) per node to 4 and the OpenMP threads per MPI clone to 12. As mentioned earlier, for this test case we keep the markers per cell ~ 1000 and increase the problem size (total number of particles, length and cells in y) together with the number of nodes N_{node}. The total number of MPI tasks then is the number of nodes multiplied by 4 clones. In figure 5.10, the time elapsed for each run is plotted against the number of nodes used. The time is normalized to the time required for the run with only one

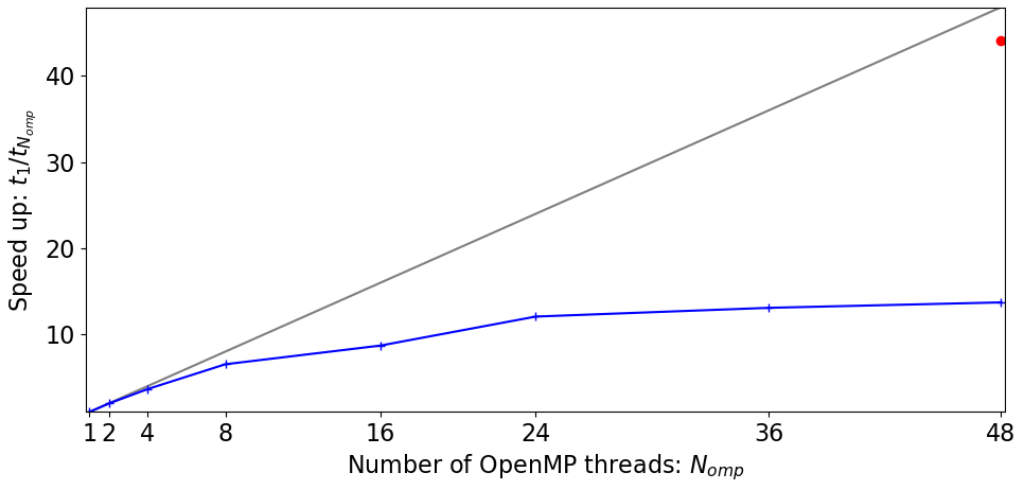


Figure 5.9: The plot shows the speed up against the number of used OpenMP threads N_{omp}. The speed up is defined as $t_1/t_{N_{\text{omp}}}$, where t_1 is the elapsed time with only one core and $t_{N_{\text{omp}}}$ the time elapsed with N_{omp} cores. The speed up for OpenMP only (blue line) with ~ 16 cores is already well below the optimal speed up of $t_1/t_{N_{\text{omp}}} = N_{\text{omp}}$ (grey line). The main reason for this is the too small problem size for the increasing number of OpenMP cores. The red point shows a test run with 8 MPI tasks with 6 OpenMP processes for each and shows a huge speed up.

node.

We see that after a small decrease for 2 nodes with increasing number of nodes, also the required time increases slightly. However, the increase is still rather small for up to 32 nodes, where the elapsed time is only $\sim 9\%$ higher than for the run with 1 node.

In general, we have to say that these scaling tests were done rather superficial and could still be significantly improved by optimizing the test case, the core pinning and the OpenMP vs. MPI ratio. But for a quick overview of the code's current scaling behavior, they already give a sufficient indication. A more detail investigation is planned for the future, together with a code optimization effort.

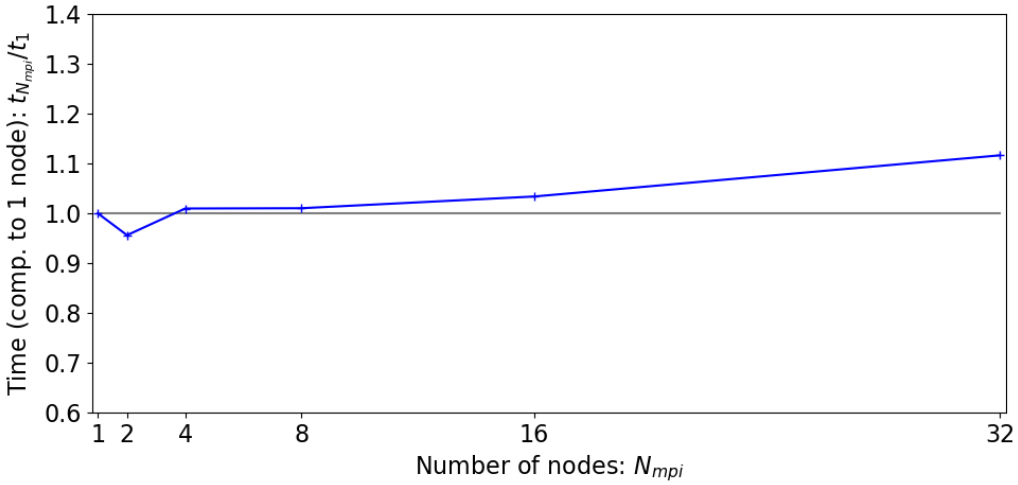


Figure 5.10: The plot shows the time elapsed for each run for a different number of applied nodes N_{node}. With increasing number the problem size increases simultaneously. The time is normalized to the time required for the run with only one node. The number of MPI clones per node is fixed to 4 and the number of OpenMP threads per MPI clone is 12.

6 1D SOL simulations

To focus on the sheath physics and reduce complexity of the system, we start with investigating a one-dimensional electrostatic model with straight open field lines and sheath boundaries at both sides of the domain. We apply a gyrokinetic full-f model with a linearization in the polarization equation. The specific problem we want to look at was already studied by two gyrokinetic continuum codes by E. Shi et al. 2015 and Pan et al. 2016. Here, a central hot source, which resembles an edge-localized-mode (ELM) heat-pulse, heats the domain and thus the parallel transport in the SOL onto the divertor plate is simulated. The applied parameters are chosen to model an ELM heat pulse in the JET tokamak. Principally, in our simulations the same set-up as in the two continuum code studies is used to investigate the similarities and differences of the continuum approaches compared to our PIC based approach. Also PIC specific subtleties and challenges, such as statistical noise, for studying this specific problem will be presented. This initial step is needed to be able to go to more complex geometries with the required confidence of correctly functioning key features such as sheath algorithms. Due to the GK model and the logical sheath boundary conditions we are no longer bound to the spatial (λ_{De}) and temporal (w_{ce}^{-1}) resolution requirements, but can increase these significantly. In the following, we will show results for the collisionless as well as the collisional case.

6.1 Introduction

A key issue for high-power tokamaks like ITER and beyond is the power load on plasma facing materials (such as the divertor or first wall components) from periodic energy outbursts into the SOL in the high-confinement mode. Type I ELMs (or "giant" ELMs) [143] are mainly causing these outbursts at high-power level. An ELM can be described as a MHD instability which causes a loss of stored energy and a profile relaxation and is triggered by steep pressure gradients in the plasma edge region (such as the H-mode pedestal) [144]. The eroded material potentially will have to be replaced more frequently in ITER and eroded atoms that penetrate into the plasma thus deteriorate the energy gain. As shown by previous studies, in only one single ITER discharge hundreds of

ELMs are expected to occur [30]. Due to this high number of plasma wall interaction via ELMs, to limit the damage on plasma facing components the suppression of ELMs is essential. Here, mainly two schemes are currently under investigation for ELM suppression: ELM control coils within the vessel and pellet injection [145]. However, an accurate prediction of heat transport in the SOL for future devices by simulation is required, to further develop these and other concepts.
With ELM heat pulse simulations decisive information on material erosions, due to loads on divertor plates and peak surface temperatures due to ELMs can be gained. Pitts et al. studied fully kinetic collisional 1D3V PIC simulations of the parallel propagation of ELM heat pulses with different energies, temperatures, densities and durations already in 2007 [123]. The typical heat-flux increase time on the order of the fast-ion sound-transit time as measurable on several machines could be investigated with their simulations. They were able to correctly predict the quantity range of the fraction of energy that was deposited in a JET measurement before the heat flux peak (0.25-0.35). Based on these results for low ELM energies, they felt confident to also make a prediction for a small Type I ELMs of 2.46MJ in ITER ($\sim$ 30). According to the authors the main motivation they had not use fluid, but to use kinetic simulations was the usual application of assumed constant sheath heat-transmission coefficients and approximate flux limiters on parallel heat fluxes in fluid models.
The ELM heat flux problem again was addressed by Havlíčková et al. in 2012, where they performed a benchmark on three different code types: a fluid, a kinetic Vlasov and a kinetic PIC code [146]. Their simulation focused on only one spatial dimension parallel to the field line. A rather simple model was used by them for the ELM, where it was modeled as a hot source near the midplane without trying to calculate the MHD instability and reconnection processes that drive the ELM. This setup is viable to simulate divertor heat fluxes and parallel transport in the SOL, although it is a quite reduced model.
With this simplified model we can also focus on testing the general code features. In general, the same problem was chosen by E. Shi et al. [115] and Pan et al. [94] for their ELM simulations with gyrokinetic continuum codes, except for the initial conditions. In addition to kinetic electrons, E. Shi [52] also implemented an adiabatic electron response. Both implemented logical-sheath boundary conditions [114], which are able to model the effects of a Debye sheath, but have the advantage, that it actually does not have to be resolved (see section 3.2). Compared to kinetic codes, this combination of gyrokinetics and a logical sheath provides a significant speed up, since the restriction towards small time steps of $\sim w_{\mathrm{ce}}^{-1}$ and spatial resolutions of $\sim \lambda_{\mathrm{De}}$ is lifted. This approach is surely computationally heavier than some fluid models, but on the other hand ensures that kinetic effects are included.
We chose to implement exactly the same model for our gyrokinetic PIC simulations as

previously used by E. Shi et al. [115] and Pan et al. [94] for kinetic electrons to have the ability to benchmark its consistency with the previously published results. Therewith, we can gain confidence of the correct code implementation. This is important, since for this type of open field line systems analytic solutions do not exist. This one-dimensional setup also prepares for studying plasma physics phenomena of real linear devices in higher dimensions, such as LAPD [147]. This machine was previously studied by the full-f gyrokinetic continuum codes Gkeyll [90] and GENE [95]. As PIC codes have their own subtleties, such as statistical noise or different methods for the particle distribution initialization, we also want to address these in our work.
In section 6.2 we describe the specific electrostatic gyrokinetic equations implemented in PICLS for the 1D heat pulse problem. In section 6.3 the setup of the heat pulse simulation is described. The corresponding results for the non-collisional and the collisional cases are shown in sections 6.4 and 6.5. The summary and some conclusions are shown in the last section 6.6.

6.2 Physical model implemented

The model we chose for this chapter consists of the gyrokinetic equations in the full-f case in one spatial dimension, with a linearization in the field equation. We neglect Larmor radius effects (due to the 1D set-up) and focus on long-wavelengths in the drift-kinetic limit. The electric potential is calculated based on the Polarization equation introduced in eq. 2.40, with additional adaptations required for the 1D case.

6.2.1 Gyrokinetic electrostatic model in 1D1V with kinetic electrons

The Hamiltonian of the applied electrostatic full-f gyrokinetic model (with a linearized field equation) is $H_{p,0\&1} = \frac{m_p {v_\parallel}^2}{2} + \mu B + e_p J_{p,0}\phi$ as already shown in equation 2.29. Therewith the equations of motion in slab geometry can be derived for the 3D case (see eqs. 2.60). Within this chapter the 1D1V versions of these equations are required to evolve the markers according to the PIC algorithms applied. These can be written as:

$$\begin{aligned} \dot{\mathbf{R}} &= v_\parallel \mathbf{b}, \\ \dot{v_\parallel} &= -\frac{e_p}{m_p} \mathbf{b} \cdot \nabla J_{p,0}\phi. \end{aligned} \tag{6.1}$$

The B-field in the 1D case is parallel to the z-direction of the domain and in our case $B = \text{const}$. Thus, in the $\dot{\mathbf{R}}$ equation the $\mathbf{B} \times \nabla B$, as well as the $\mathbf{B} \times \nabla J_{p,0}$ term cancel

out. For the 1V case also $\mu = 0$ applies and thus also the first term in the $\dot{v}_\parallel$ equation disappears.
The equation we use for the calculation of the electric potential is the polarization equation (eq. 2.40):

$$-\sum_p \nabla_\perp \frac{n_{p,0} m_p c^2}{B^2} \nabla_\perp \phi = \sum_p \int \mathrm{d}W e_p J_{p,0} f. \tag{6.2}$$

But additional modifications are necessary to adjust the equation to the 1D heat pulse problem, we want to study (see section 6.2.2).
The total conserved energy of the system consists of a field part and a kinetic part, as shown in section 2.4 [111]:

$$\mathcal{E}_{tot} = \mathcal{E}_k + \mathcal{E}_f = \sum_p \int \mathrm{d}W \mathrm{d}V H_{p,0} f_p + \sum_p \frac{1}{2} \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} \phi f_p, \tag{6.3}$$

with $H_{p,0} = \frac{1}{2} m_p {v_\parallel}^2 + \mu B$ in the kinetic part.

6.2.2 Modification of the Polarization equation

We modify the Polarization equation, following the derivations in [52]. Thus, in the following we only want to sketch the modifications without going into detail of the derivation.
In a first step the Polarization equation 6.2 and the Hamiltonian evolution equation $\frac{\partial F_p}{\partial t} = \{H_p, F_p\}$ are linearized and Fourier transformed to obtain a wave dispersion relation. Assuming that $q_\mathrm{e} = q_\mathrm{i}$ and neglecting ion perturbations, one obtains:

$$k_\perp^2 \rho^2 + [1 + \epsilon Z(\epsilon)] = 0, \tag{6.4}$$

with $\rho^2 = T_\mathrm{e}/(m_\mathrm{i}\Omega_\mathrm{ci}^2)$, $\epsilon = w/(\sqrt{2} k_\parallel v_\mathrm{te})$, $v_\mathrm{te} = \sqrt{T_\mathrm{e}/m_\mathrm{e}}$ and the plasma dispersion function $Z(\epsilon) = \pi^{-1/2} \int dt \, \exp(-t^2)/(t - \epsilon)$. The solution to this dispersion relation for $\epsilon \gg 1$ is a wave with frequency:

$$w_\mathrm{H} = \frac{k_\parallel v_\mathrm{te}}{|k_\perp| \rho}. \tag{6.5}$$

From the equation, we see that for $k_\perp \rho \ll 1$ this wave must be treated with care to guarantee numerical stability. The resulting wave is the electrostatic limit of the shear Alfvén wave ([148]) and due to its high frequency does not affect parallel transport in the SOL.

Thus, in the electrostatic case a modification of the Polarization equation is required, to set a minimum value for $k_\perp$ and thus slow down the electrostatic shear Alfvén wave. By introducing the shielding factor $s_\perp(z,t) = k_\perp^2(z)\epsilon_\perp(z,t)$, with $\epsilon_\perp = \sum_p \frac{n_{p,0}m_pc^2}{B^2}$, one can write the modified polarization equation as:

$$-\nabla_\perp C_\epsilon \nabla_\perp \phi + s_\perp(z,t)(\phi - \langle\phi\rangle) = \sum_p \int \mathrm{d}W e_p J_{p,0} f. \tag{6.6}$$

Where we use the flux-surface-averaged, dielectric-weighted potential:

$$\langle\phi\rangle = \frac{\int dz s_\perp \phi}{\int dz s_\perp}. \tag{6.7}$$

This equation can be further simplified towards an algebraic form by setting the fixed coefficient C_ϵ to $C_\epsilon = 0$ as a limit. This discards the usual differential term and should work well for low frequency dynamics. Also $k_\perp(z)$ can be set to a constant value to only allow a single wave number. Setting this value is crucial since it needs to be small enough that the frequency of the wave from eq. 6.5 is high enough to not interact with other relevant dynamics, and in addition, small enough to prevent a too small time step. Both, E. Shi [52] and Pan et al. [94] found that simulation results within this range are not sensitive to the perpendicular wave number and chose $k_\perp\rho = 0.2$ as an appropriate value.
In eq. 6.6, the flux-surface-averaged potential $\langle\phi\rangle$ is subtracted from ϕ to assure that the gauge invariance of the modified polarization equation is maintained. Also the applied logical sheath boundary conditions are a reason for this choice. In the chosen 1D case specifically, the net guiding center charge vanishes, $\int \sigma_{tot} dz = \int \sum_p \int \mathrm{d}W e_p J_{p,0} f dz = 0$, because the net flux is set to $j_\parallel = 0$. The polarization charge density averages to 0 as well, which can be shown by applying the integral over dz on the left-hand-side of equation 6.6. Nevertheless, this subtraction of $\langle\phi\rangle$ can be neglected for calculating the E-field and does not influence the gyrokinetic equations of motion. The modified Polarization equation used in our model as already presented in [102] is:

$$s_\perp(z)\phi(z) = k_\perp^2(z) \sum_p \frac{n_{p,0}m_pc^2}{B^2}\phi(z) = \sum_p \int \mathrm{d}W e_p J_{p,0} f. \tag{6.8}$$

Eventually, in our PIC algorithm with finite-element discretization the Poisson matrix, has to be changed to the mass matrix with an additional factor of $k_\perp^2$.
$k_\perp$ is normalized to the inverse of the electron cyclotron radius $1/\rho_{\mathrm{ce}}$ and can be expressed as $k_\perp = \bar{k_\perp}/\rho_{\mathrm{ce}}$. By setting $l_{\mathrm{norm}} = \rho_{\mathrm{ce}}$ in our simulations, the normalized

form of the Fourier transformed Polarization equation can be obtained as:

$$\sum_p \bar{k_\perp}^2 \frac{n_0 M \bar{m}_p}{\bar{B}^2 B_{\text{norm}}^2} \phi(z) = \sum_p \int \mathrm{d}W e \bar{Z}_p J_{p,0} f, \tag{6.9}$$

with $\bar{k_\perp} = 0.2$. For the calculation of the conserved energy, the $\langle\phi\rangle$ term would have to be taken into account, as shown in [115]. However, in section 6.2.1 we showed the conserved energy for the more general case of the original Polarization equation 2.40.

6.2.3 Geometry of the ELM heat-pulse problem

As described in section 6.1, we want to study an ELM heat pulse occurring at the center or mid-plane of the plasma and from there propagating through the SOL towards the divertor targets. In 1D the associated open field lines become straight lines along the dimension z, that force the particles to propagate along this direction. The actual curved structure of the field lines thus is not taken into account. An illustration of the applied simplification is sketched in figure 6.1.

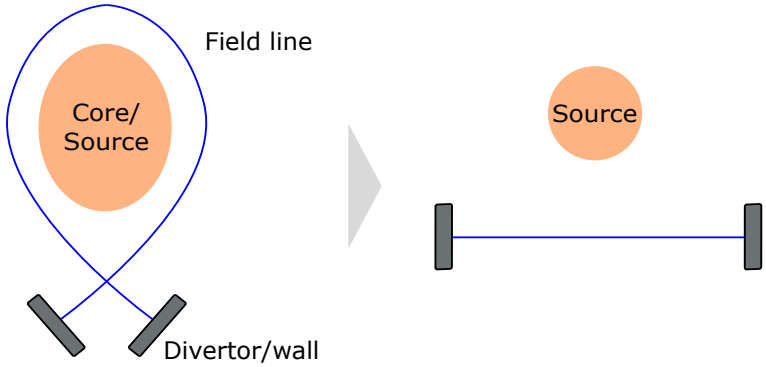


Figure 6.1: Illustration of the simplified geometry applied for the 1D1V ELM heat-pulse model. The in reality curved field lines are reduced to straight lines along the z-direction with a hot source in the center of the simulation domain.

6.3 Simulation setup

At the beginning of the simulation, electrons and ions are set to predefined initial conditions, similar to the setup shown in [115] and [94] to be able to benchmark with their work. At time $t = 0$ an intense particle source, which is supposed to resemble an ELM, is switched on. At $t = 200\mu$s the source is significantly decreased to model a less

intense inter-ELM phase until the end of the simulation run at $t = 350\mu$s. This setup was already shown in our previous publication [102].

6.3.1 Initial conditions

Initially, electrons and ions are set to a fixed spatial and velocity distribution, to be able to benchmark with previous studies. The simulation results, however, are mainly insensitive to the initial background distribution. In their fully kinetic PIC simulations for example, Pitts et al. [123] performed runs with a weaker initial pre-ELM source to achieve a quasi-steady state.

Electron initial conditions

The initial electron distribution function can be written as:

$$f_{\mathrm{e0}}(z, v_{\parallel}, T_{\mathrm{e0}}) = n_{\mathrm{e0}}(z) F_{\mathrm{M}}(v_{\parallel}, T_{\mathrm{e0}}), \tag{6.10}$$

where $F_{\mathrm{M}}(v, T_{p0}) = \frac{1}{\sqrt{2\pi T_{p0}/m_p}} \exp\left(\frac{-m_p {v_{\parallel}}^2}{2T_{p0}}\right)$ is the Maxwellian distribution in 1D for species p (in this case p stands for electrons). The initial temperature is fixed to $T_{\mathrm{e0}} = 75$eV and the electron density profile (in $10^{13}\mathrm{cm}^{-3}$) can be formulated as:

$$n_{\mathrm{e0}}(z) = 0.7 + 0.3\left(1 - \left|\frac{z}{L}\right|\right) + 0.5\cos\left(\frac{\pi z}{L_{\mathrm{s}}}\right) H\left(\frac{L_{\mathrm{s}}}{2} - |z|\right) \tag{6.11}$$

with L, which is half the size of the simulation domain $[-L; L]$, the source size L_{s} and the Heaviside step function $H(.)$.

Ion initial conditions

In the case of the ions, the initial distribution function is expressed as a combination of right and left half-Maxwellian distribution functions:

$$\begin{aligned} F_{\mathrm{L}}(z, v_{\parallel}, T_{\mathrm{i0}}) &= 2n_{\mathrm{i0}}(z) F_{\mathrm{M}}(v_{\parallel}, T_{\mathrm{i0}}) H(-v_{\parallel}) \\ F_{\mathrm{R}}(z, v_{\parallel}, T_{\mathrm{i0}}) &= 2n_{\mathrm{i0}}(z) F_{\mathrm{M}}(v_{\parallel}, T_{\mathrm{i0}}) H(v_{\parallel}) \end{aligned} \tag{6.12}$$

with the initial ion density $n_{\mathrm{i}0}$, which is set equal to the initial electron density $n_{\mathrm{e}0}$. Depending on the particle's position z, the ion distribution function reads:

$$f_{\mathrm{i}0}(z, v_{\parallel}, T_{\mathrm{i}0}) = \begin{cases} F_{\mathrm{L}} & z < -\frac{L_{\mathrm{s}}}{2}, \\ \left(\frac{1}{2} - \frac{z}{L_{\mathrm{s}}}\right) F_{\mathrm{L}} + \left(\frac{1}{2} + \frac{z}{L_{\mathrm{s}}}\right) F_{\mathrm{R}} & -\frac{L_{\mathrm{s}}}{2} < z < \frac{L_{\mathrm{s}}}{2}, \\ F_{\mathrm{R}} & \frac{L_{\mathrm{s}}}{2} < z. \end{cases} \tag{6.13}$$

The ion temperature profile is defined as (in eV):

$$T_{\mathrm{i}0}(z) = 100 + 45\left(1 - \left|\frac{z}{L}\right|\right) + 30\cos\left(\frac{\pi z}{L_{\mathrm{s}}}\right) H\left(\frac{L_{\mathrm{s}}}{2} - |z|\right). \tag{6.14}$$

These initial conditions for electrons and ions are equal to the set-up in [94]. Except for the ion particle distribution, this setup is also equal to the one chosen in [115]. Here, in order to minimize the excitation of high-frequency shear Alfvén waves in the electrostatic limit, the ion density profile initially is determined in a way that electrons are distributed according to the Boltzmann relation. But similar to Pan et al. [94], by setting the initial ion density equal to the initial electron density also in our simulations no numerical issues occur. As stated in [94], replacing the $cos(\pi z/L_{\mathrm{s}})$ combined with the Heaviside function $H(L_{\mathrm{s}}/2 - |z|)$ which are used in the initial and source term profiles by an exponential function $\sqrt{2/\pi}\exp(-(\pi z/L_{\mathrm{s}})^2/2)$, changes the simulation results only marginally.

6.3.2 ELM and inter-ELM phase

The parameters and set-up we use for the ELM and inter-ELM phases were already applied by Havlíčková et al. [146] and are based on a simplified case from the JET tokamak. Generally, a plasma blob exhausted in an ELM crash is represented by this model and is simulated via a hot electron-deuterium source at the SOL midplane. This source in our 1D setup is placed at the center of the domain $[-L, L]$ and two divertor plates are positioned at each end as boundaries. After 200μs at the end of the ELM phase, an inter-ELM phase with a colder and weaker source follows. The source function S_{ELM} depends on time t, position z and velocity $v_{\parallel}$ and is expressed by:

$$S_{\mathrm{ELM}}(z, v_{\parallel}, t) = g(t)S(z)F_{\mathrm{M}}(v_{\parallel}, T_s(t)) \tag{6.15}$$

with

$$S(z) = S_0 \cos\left(\frac{\pi z}{L_\mathrm{s}}\right) H\left(\frac{L_\mathrm{s}}{2} - |z|\right), \tag{6.16}$$

$$g(t) = \begin{cases} 1 & 0 \leq t \leq 200\mu\mathrm{s} \\ 1/9 & 200\mu\mathrm{s} < t, \end{cases} \tag{6.17}$$

$$T_e = \begin{cases} 1500\mathrm{eV} & 0 \leq t \leq 200\mu\mathrm{s} \\ 210\mathrm{eV} & 200\mu\mathrm{s} < t, \end{cases} \tag{6.18}$$

$$T_i = \begin{cases} 1500\mathrm{eV} & 0 \leq t \leq 200\mu\mathrm{s} \\ 260\mathrm{eV} & 200\mu\mathrm{s} < t. \end{cases} \tag{6.19}$$

The particle source intensity S_0, which scales with the pedestal density and temperature, is set to $S_0 = An_\mathrm{ped}c_\mathrm{s,ped}/L_\mathrm{s} = 9.066 \times 10^{17}\mathrm{cm}^{-3}\mathrm{s}^{-1}$. A, the proportionality constant, is set to $1.2\sqrt{2} \approx 1.7$. The particle source within the code is modeled via a Monte-Carlo generation of particles based on the introduced source function.

Relevant parameters

The key parameters for the simulation are taken from [146] and are shown in table 6.1.

Parameter	Value	Description
$2L$	80m	Length of simulation domain
L_s	25m	Length of source region
t_ELM	200μs	Duration of ELM phase
τ_i	149μs	Ion transit time ($L/c_\mathrm{s,ped}$)
τ_e	2.5μs	Electron transit time ($L/v_\mathrm{te,ped}$)
T_ped	1500eV	Ion and electron temperature at ELM pulse
S_0	$9.066 \times 10^{17}\mathrm{cm}^{-3}\mathrm{s}^{-1}$	Particle source intensity
$k_\perp\rho$	0.2	Perpendicular wave number
B	2T	B-field strength in parallel direction

Table 6.1: Simulation parameters used for 1D1V ELM heat pulse simulations.

To test convergence, simulations with varying numbers of field grid cells, particles per cell and perpendicular wave numbers were executed. We chose a minimum value of $\sim 10,000$ particles per cell to achieve high enough resolution for the relevant sheath parameters (sheath potential, heat flux, etc.). With a lower number of particles per

cell the physical results are still obtainable, however the signal to noise ratio can be affected. We did not want to study numerical convergence in detail here, thus a rather high number of markers was chosen to ensure good resolution. To ensure high enough spatial resolution of the fields in z-direction, a minimum value of $n_z \geq 16$ grid cells should be implemented. The following results were performed with $\sim 100,000$ particles per cell, $n_z = 32$ and $k_\perp \rho = 0.2$. We discuss the effect on the simulation results of varying $k_\perp \rho$ $(0.05 - 1.0)$. We obviously use an excessive number of particles per cell for this simulation, however, since our runs finish within hours, there was no necessity to decrease resolution. One must keep in mind that generally for full-f codes a higher number of particles per cell ($> 1,000$) is required than for delta-f. For the 3D simulations, we reduced the number of particles per cell and plan to implement further noise-reduction techniques to save computational time, if required.

6.4 Simulation results: collisionless 1D1V model

In this section we study the plasma behavior during the ELM phase and beyond. Therefore, the spatial profiles at the end of the ELM phase, together with the time-dependent heat flux towards the divertor as well as the sheath potential ϕ_{sh} development over time will be investigated. Our problem is symmetric, thus, ϕ_{sh} can be measured deliberately at the right or left boundary; we therefore chose the right boundary. In this section only simulations without collisions are regarded, which we already published in [102]; for collisional studies we refer to section 6.5 or to our respective publication [127].

6.4.1 ELM phase spatial profiles

For the heat flux problem the relevant spatial profiles for species p are density n_p, parallel temperature $T_{\parallel,p}$, parallel particle flux Γ_p and parallel heat flux Q_p and can be written as:

$$n_p = \int_{-\infty}^{\infty} f_p dv, \tag{6.20}$$

$$\Gamma_p = \int_{-\infty}^{\infty} f_p v dv, \tag{6.21}$$

$$Q_p = \frac{1}{2} m_p \int_{-\infty}^{\infty} f_p v^3 dv + T_\perp \int_{-\infty}^{\infty} f_p v dv, \tag{6.22}$$

$$T_{\parallel,p} = \frac{1}{n_p} m_p \int_{-\infty}^{\infty} f_p (v - \langle v \rangle_p)^2 dv. \tag{6.23}$$

Here, f_p is the particle distribution function and the perpendicular temperature $T_\perp$ is set to the pedestal temperature $T_\perp = T_{\text{ped}} = 1500\text{eV}$ (equal to previous publications).

For all figures within this section, values for the n_z grid cells were calculated and we applied a cubic spline interpolation on 128 diagnostics cells, to smooth the shape of the profiles. The mentioned profiles are plotted in figure 6.2 for $k_\perp \rho = 0.2$ shortly before the system comes into the post-ELM phase when the source is switched off (at 200μs).

We want to emphasize that in the upper left plot in figure 6.2 the gyrocenter density

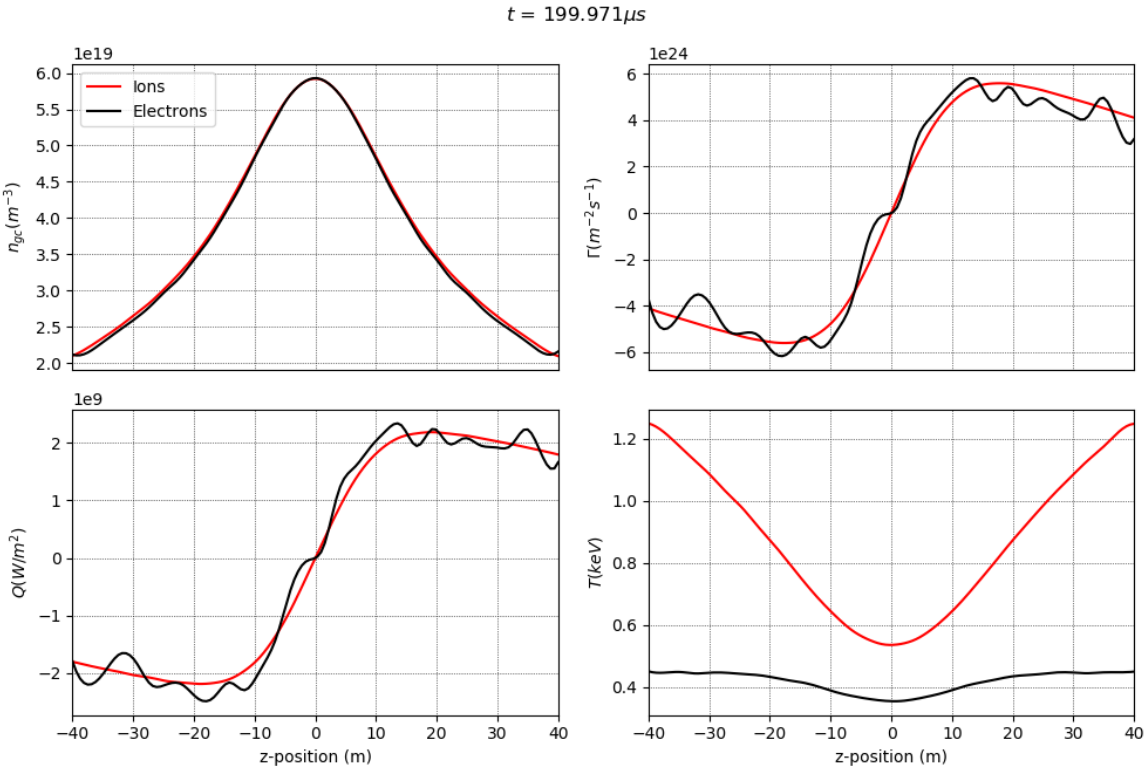


Figure 6.2: Spatial profiles of electrons (black) and ions (red) of gyrocenter density n_{gc}, parallel particle flux Γ, parallel heat flux Q and parallel temperature $T_\parallel$ within the 1D simulation domain for $k_\perp \rho = 0.2$. The snapshot of the profiles is taken shortly before the end of the ELM phase at 200μs.

is shown. This means that in the electron case the polarization term of equation 6.8 is added. The overlap of the ion and electron gyrocenter density implies that the electrons are mostly bound to the ions and are transported towards the target with the same flux rate, which is a good indicator for the accuracy of the simulation. This largely agrees with the results seen in Pan et al. [94]. Directly at the boundaries a small excess of electrons can be detected, which is due to the logical sheath boundaries, that lead to an absorption of ions and reflection of electrons by the boundary.

The remaining spatial profiles are in good agreement with Pan et al.'s results, as well. The electron temperature for example is slightly higher than in Pan et al.'s case, however, the overall lower value for electrons compared to ions still is due to the same origin of selective loss of high-energy electrons to the wall. This slightly higher value indicates that higher energy electrons are introduced by our particle sources. A lower limit for the electron velocity in the source distribution function thus can decrease the electron temperature. In case of the parallel particle and heat flux the electron fluxes oscillate around the ion profiles, however, still a good overlap was obtained. The reason for this are PIC-inherent noise effects. The profiles can be further flattened and an even better overlap between both species can be achieved by increasing the number of particles, or averaging over the diagnostics cells.
The drop of the fluxes towards the domain boundary in both cases clearly indicate, that the system has not yet reached an equilibrium between the lost particles at the walls and the introduced particles from the source after 200μs. In equilibrium, towards the domain boundary the flux profiles outside the source region are flat.
The densities for ions and electrons (without the Polarization term) together with the Polarization $s_\perp(z)\phi(z)$ (divided by e to achieve comparability), are plotted in figure 6.3 for values of $k_\perp\rho = 0.05 - 1.0$, to discuss the effects of varying $k_\perp\rho$. The ion (electron) density is decreasing (increasing) with decreasing $k_\perp\rho$. The electrons are the lighter and more mobile species, hence the effect on their density profile is stronger. This in turn implies that the difference between the ion and electron densities decreases, as directly seen in the decreasing Polarization term $s_\perp(z)\phi(z)$, where $s_\perp(z)\phi(z)/e$ is equal to the density difference. Or in other words the electrons are stronger bound to the ions. Hence, varying $k_\perp\rho$ has hardly any effect on the other profiles: temperature, particle and heat flux.

6.4.2 Divertor heat flux and sheath potential

For calculating the parallel heat flux on the divertor targets in gyrokinetic simulations additional sheath effects have to be taken into account, compared to the calculation of the heat flux spatial profile (see eq. 6.22), to obtain:

$$Q_p = \frac{1}{2} m_p \int_{v_{c,p}}^{\infty} f_p {v_\parallel}^3 dv_\parallel + (T_\perp + q_p\phi_{sh}) \int_{v_{c,p}}^{\infty} f_p v_\parallel dv_\parallel. \tag{6.24}$$

The electrons that are reflected back into the plasma are not taken into account, hence the lower boundary of the integrals is $v_{c,p} = \sqrt{\max(-2q_p\phi_{sh}/m_p, 0)}$. This is based on the definition of the cut-off velocity as shown in eq. (3.1). The same would be valid for the reflected ions, in the less probable case where more ions than electrons hit the divertor. To account for the acceleration (deceleration) of outgoing ions (electrons) by

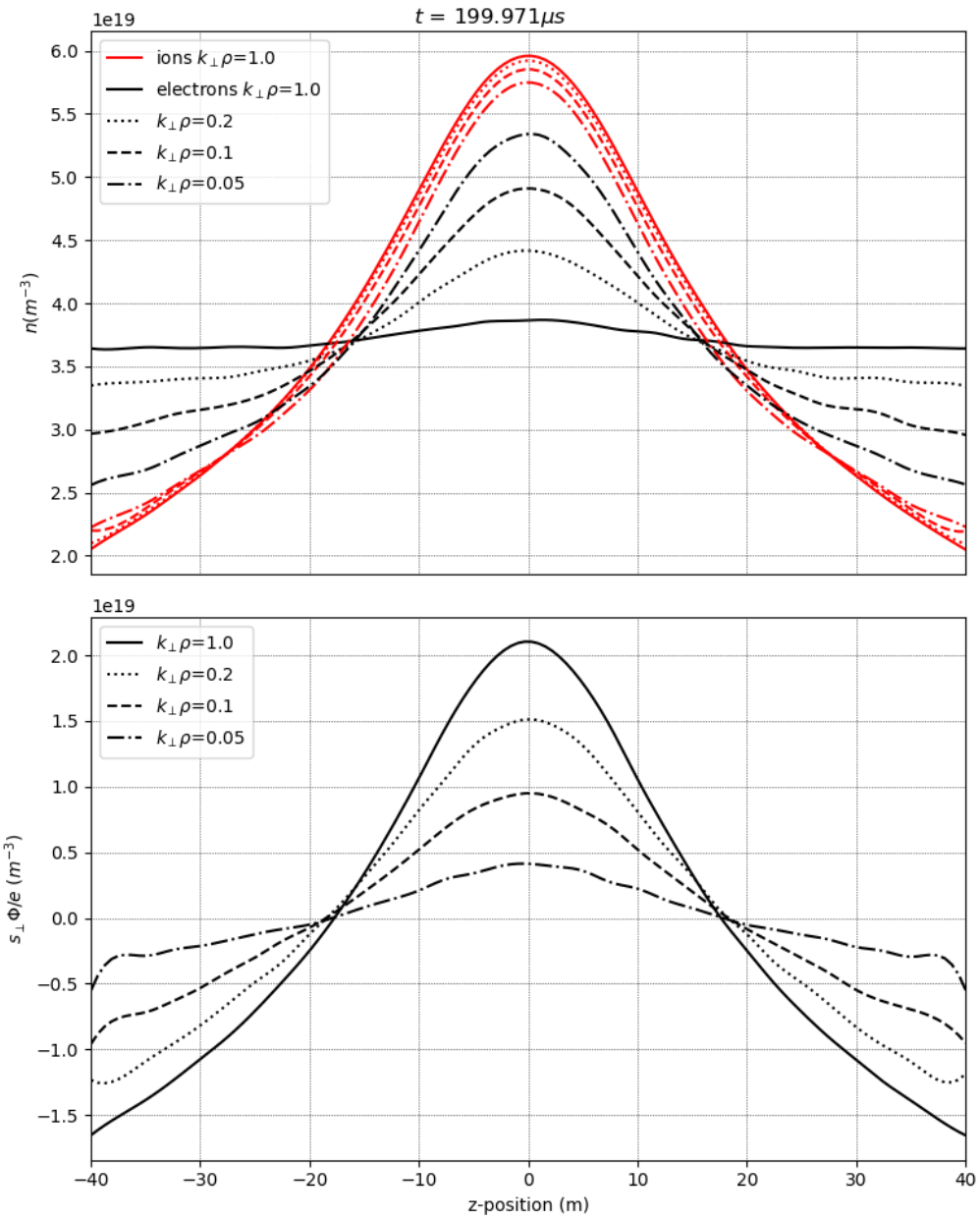


Figure 6.3: Spatial density profiles for electrons (without the polarization term) and ions and the polarization $s_\perp(z)\phi(z)$ divided by e for varying $k_\perp\rho = 0.05 - 1.0$.

the sheath, the additional ϕ_{sh} term has to be added.
A moving average of about 50 time steps ($\sim 0.1\mu$s) was chosen, for all graphs within

this section, to be able to decrease the number of required particles per cell and thus achieve fast simulations and results with low enough noise. The heat flux on the right sheath boundary and the number of particles that hit this boundary are displayed in figure 6.4 for the same $k_\perp \rho = 0.2$ run as in the previous section. The majority of electrons is reflected and only as many electrons as ions are eventually leaving the domain, due to the applied logical sheath boundary conditions. Looking at the bottom plot of figure 6.4, we see that for our simulations more electrons than ions are hitting the wall at each point in time. This means, that the number of electrons and ions that are absorbed by the wall is forced to be equal to the number of ions that hit the wall.
For values smaller than $0.5\tau_\mathrm{e} = 0.5L/\sqrt{T_\mathrm{ped}/m_\mathrm{e}}$, a quick rise of the parallel electron heat flux can be seen, which results from the first fast electrons of the hot ELM source hitting the wall. Shortly after, the electron heat flux slightly decreases before staying at a constant level of $\sim 0.4 \times 10^9 \mathrm{W/m}^2$ until $\sim 0.5\tau_\mathrm{i}$. During the same time period, a very slight increase of the ion heat flux can be measured initially and the heat flux remains at a rather low value of $\sim 0.1 \times 10^9 \mathrm{W/m}^2$. This leads to a constant total heat flux of $\sim 0.5 \times 10^9 \mathrm{W/m}^2$ until shortly before $0.5\tau_\mathrm{i}$. The sheath potential that builds up at $\sim 0.5\tau_\mathrm{e}$ is responsible for the slight increase (decrease) of the ion (electron) heat flux at this time (see figure 6.5). In previous studies with continuum codes (Fig. 2 in [94] and Fig. 3 in [52]), however, the ion heat flux increases as much as the electron heat flux drops at $0.5\tau_\mathrm{e}$ and remains higher for the rest of their simulations. But their results show a similar increase of the total heat flux to $\sim 0.5 \times 10^9 \mathrm{W/m}^2$ in this time period. In case of simulations with a fully-kinetic PIC code (Fig. 2 in [146]), similar to our case, for this time period the electron heat flux stays continuously above the ion heat flux.
Then, at about half of the ion transit time, when the first fast ions from the ELM source arrive, the electron as well as the ion heat fluxes increase steeply until $t = 200\mu\mathrm{s}$ when the ELM crashes. Here, a peak total heat flux of $\sim 5.1 \times 10^9 \mathrm{W/m}^2$ is obtained. The electron heat flux drops immediately at this moment, whereas a significantly slower ion response is observable, which leads to a deferred drop. Again, we can compare this period to the previous simulations with continuum codes, to obtain a good qualitative agreement. However, the peak total heat flux for electrons and ions and thus also the total heat flux is slightly higher in our case ($\sim 5.1 \times 10^9 \mathrm{W/m}^2$ compared to $\sim 4.1 \times 10^9 \mathrm{W/m}^2$). The main reason for this difference is the differing source implementation of our PIC compared to the continuum codes of previous studies. The same maximum values of the heat fluxes can actually be constructed, by cutting-off the electron velocities at lower values or narrowing the particle sources' velocity distributions. The peak total heat flux in case of the fully-kinetic PIC code is very similar to our result, but the peak ion (electron) heat flux is significantly higher (lower) than in our simulations. This could be due to the differences between their fully-kinetic

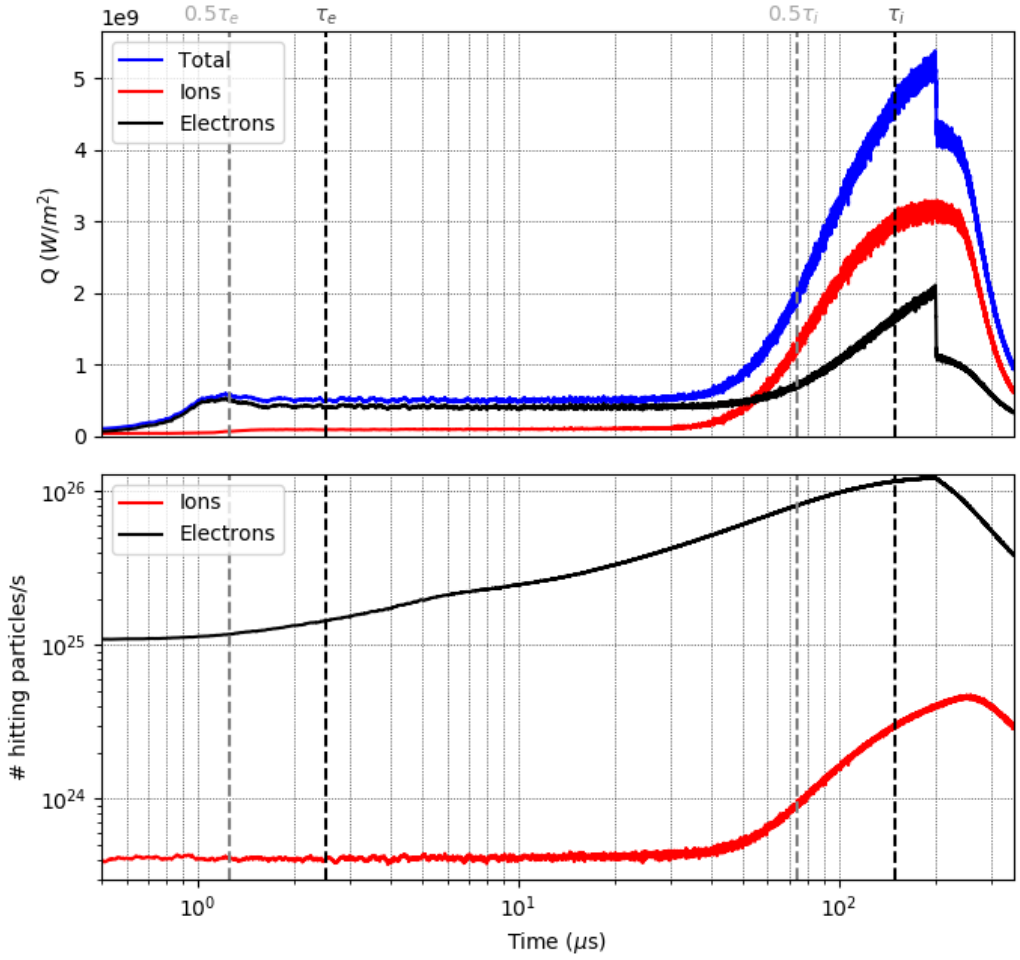


Figure 6.4: Evolution of ion (red), electron (black) and total (blue) heat flux, according to eq. (6.24), and number of hitting particles on right sheath boundary. The (half of the) thermal ion and electron transit times τ_e and τ_i are indicated by (grey) black vertical lines.

and our gyrokinetic model.
The total ELM energy delivered to the divertor for each species can be calculated by integrating the fluxes over time. This is 66.6% (33.4%) of the total ELM energy in case of the ions (electrons) and again shows very similar values compared to the previous continuum code simulations.
Different from the SOL heat fluxes in section 6.4.1, the heat fluxes on the domain wall include deceleration (acceleration) of electrons (ions). By looking at the number of hitting particles in figure 6.4, we can comprehend the reason for the strong increase of the heat flux at $\sim 0.5\tau_\mathrm{i}$. The graph clearly shows a steep increase of incoming ions at

the sheath boundary. The number of incident ions is still rising after 200μs, which is a good indicator for the deferred ion heat flux drop, too. And in case of the electrons as expected the increase of hitting particles starts directly at $\sim 0.5\tau_\mathrm{e}$ and immediately drops after switching off the ELM source.
An even more complete physical picture can be obtained by looking at the time-dependent evolution of the sheath potential ϕ_sh in figure 6.5. At the start of the simulation, only the cold initial distribution determines ϕ_sh. Around $\sim 0.5\tau_\mathrm{e}$, due to the incoming suprathermal electrons from the ELM source, the potential quickly rises to $\sim$ 3keV. Then, the potential stays mainly constant, until the suprathermal ions arrive at $\sim 0.5\tau_\mathrm{i}$. The large sheath potential causes a deceleration and reflection of electrons at the divertor plates. Most of the electrons are prevented from hitting the divertor, which stops the increase of the electron heat flux at $\sim 0.5\tau_\mathrm{e}$ and slightly inverts it. The sheath potential then drops steadily after the arrival of suprathermal ions and allows an increase of ion as well as electron heat flux. The qualitative behavior is very similar to the results in Pan et al.[94], with the main difference, that the increase is even steeper in the previous study and the absolute value slightly below ours ($<$ 3keV). This again is an indicator for the introduction of faster electrons into the system by our PIC specific ELM source.
We already mentioned before that the quantitative differences of our results compared to

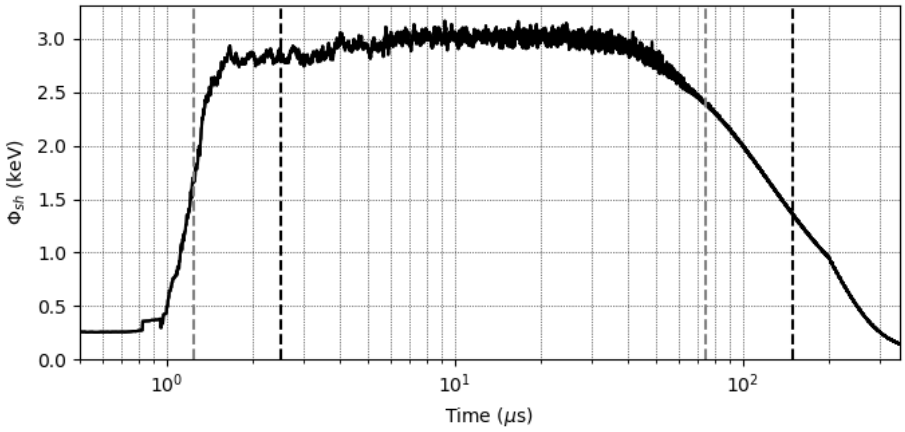


Figure 6.5: Time-dependent evolution of the potential at the right sheath boundary. The (half of the) thermal ion and electron transit times τ_e and τ_i are indicated by (grey) black vertical lines.

the previous continuum code results from [94] and [52] are mainly due to our PIC-specific implementation of the velocity distribution functions. The sources we implemented

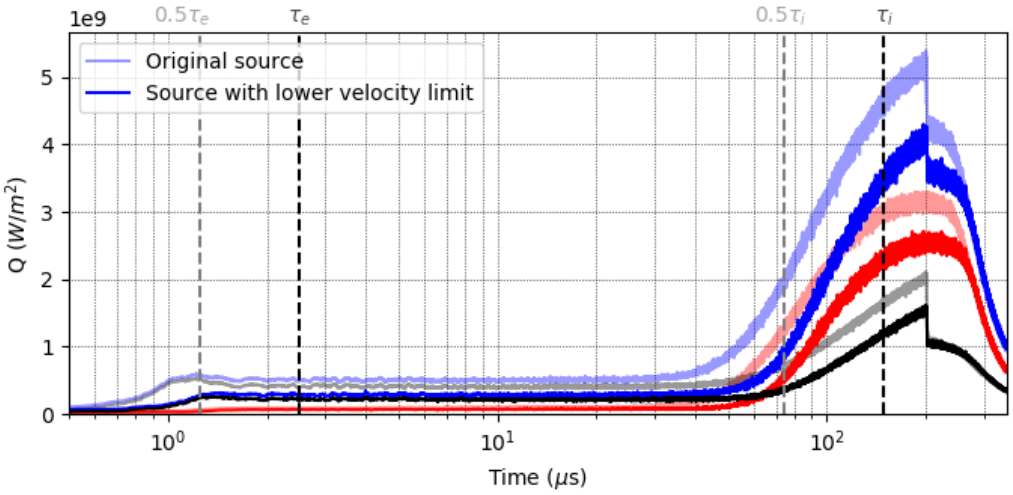


Figure 6.6: Comparison of the heat flux on the divertor for the originally implemented source (transparent) and for a source with a decreased limit for the maximum of the velocity distribution (opaque), where $v \leq 2.2 v_{\mathrm{th}}$ is set.

are assumed to allow faster particles than the sources used in the mentioned studies. Therefore, in figure 6.6 the heat flux on the divertor for a simulation with a lower maximal limit for the particle sources' velocity distribution is plotted. We thus only initialize electrons and ions up to a lower maximum velocity of $v_{\parallel} \leq 2.2 v_{\mathrm{th}} = 2.2\sqrt{k_{\mathrm{B}} T_p / m_p}$ (vs. $v_{\parallel} \leq 3.7 v_{\mathrm{th}}$), or with $f_p(v_{\parallel}) / f_{p,\mathrm{max}} = 10\%$ (vs. 0.1%), compared to our previous simulation runs. The qualitative behavior of the heat flux can still be maintained compared to the previous run. The peak values, however, are decreased because of the limited velocity of the fastest introduced particles.

By narrowing the distribution function and subsequently limiting the very high velocity particles, too, a similar picture can be achieved (not shown here). Again, the sheath potential ϕ_{sh} can thus only reach lower peak values by setting a lower velocity limit. However, the spatial profiles are only slightly affected, since most of the particles anyways are below the velocity limit. Hence, the effects of the fastest particles are less drastic.

6.5 Simulation results: collisional 1D2V model

The simulations we performed so far are based on a collisionless model in the 1D1V $(z, v_{\parallel})$ phase-space. However, to perform realistic SOL simulations particle collisions need to be introduced, especially in the 3D case. As described in section 4.1, collisions

are a key driver for particle transport across closed magnetic flux surfaces which would otherwise be confined. Due to collisions plasma is caused to diffuse from the confined region into the SOL and from there the plasma is eventually transported towards the vessel wall (or divertor) [20]. Within this section we therefore use the Lenard-Bernstein collision operator, described in section 4.3, for same-species collisions for electrons as well as ions. In addition, we added the magnetic moment μ as a second velocity component to study the previous 1D1V heat pulse problem in a domain with one spatial and two velocity coordinates $(z, v_\parallel, \mu)$ with and without collisions. We already published these results in our previous work [127]. Inter-species collisions between electrons and ions are neglected to achieve comparability with the work done by E. Shi et al. 2015.
By introducing μ, also $T_\perp$ changes over time due to collisions and is no longer a simulation parameter. The perpendicular temperature for the source however is fixed at T_{ped}, which even applies after the ELM heat pulse ends. The exact same setup as described in 6.3 will be employed for the parallel temperature. The equation for the parallel heat flux (6.24) needs to be altered, due to the evolving $T_\perp$ and the additional velocity component μ:

$$Q_p = \int_{v_{\mathrm{c},p}}^{\infty} f_p v_\parallel \left(\frac{1}{2} m_p {v_\parallel}^2 + \mu B \right) dv_\parallel + q_p \phi_{\mathrm{sh}} \int_{v_{\mathrm{c},p}}^{\infty} f_p v_\parallel dv_\parallel. \tag{6.25}$$

Within this section for all simulations we initialize the marker distribution with $\sim$ 300,000 markers per bin. This is $\geq$3 times higher than in the presented conversion study in figure 4.4 in section 4.5.2. Thus, we can ensure that the conservation error is small enough. Once the particle sources introduce particles into the simulation domain, the total number of markers even increases further. Obviously, this high number of markers per cell is excessive, however, since only some hours are required to run the simulations, we decided to not allow an increase of the noise by lowering the marker number.
The heat flux on the divertor target for collisionless and collisional 1D2V simulations is compared in figure 6.7. We want to stress the fact that the non-collisional case differs from the 1D1V simulations in section 6.4.2, because of the source applied for the μ initialization. The collision frequencies are slightly decreased for ions and electrons in the collisional case, by multiplying eqs. (4.12) with 0.5. Looking at the start of the simulation, one differences we encounter between both graphs is a lower initial heat flux before $\sim 0.5\tau_{\mathrm{i}}$ for the non-collisional run ($\sim$ 50% of the collisional case). As soon as the suprathermal ions hit the wall, the ion heat flux in the collisional case rises even higher than in the non-collisional case. For the electrons, however, a slight decrease of maximum heat flux values is visible. In the collisional case, the total heat

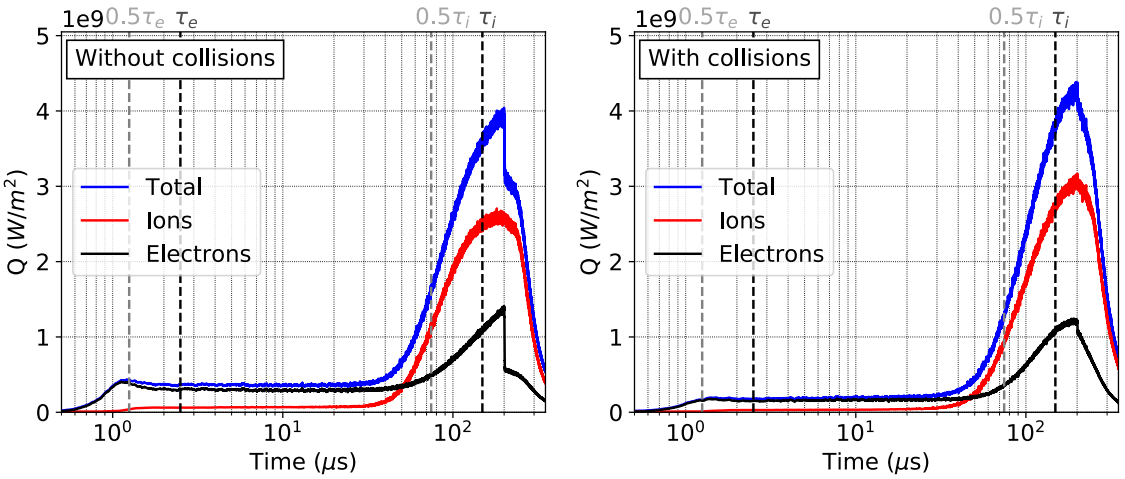


Figure 6.7: Comparison of the evolution of ion (red), electron (black) and total (blue) heat flux in the 1D2V case, according to eq. (6.25), with and without same-species Lenard-Bernstein collisions. Thermal ion and electron transit times $\tau_\mathrm{e} = 2.5\mu$s and $\tau_\mathrm{i} = 149\mu$s are indicated by black vertical lines ($0.5\tau_\mathrm{e}$ and $0.5\tau_\mathrm{i}$ are indicated by grey lines).

flux thus reaches an $\sim 8\%$ higher maximal value ($4.04 \cdot 10^9 \mathrm{W/m}^2$ vs. $4.38 \cdot 10^9 \mathrm{W/m}^2$). Further investigation of the heat flux in the collisional (non-collisional) case reveals more differences. For the share of the total heat flux over time deposited before the peak at 200μs we achieve 55% (61%) and for the total heat flux deposited by ions vs. electrons we obtain shares of 74% vs. 26% (72% vs. 28%). In addition, a 9% higher total heat flux deposited over time in the collisional case is measurable, compared the non-collisional case. The collisions introduced clearly lead to an increase in the ion heat flux and cause a higher heat flux on the target in total over time. The increase largely depends on the increased particle flux, but for a better understanding of the heat flux evolution, in figure 6.8 we show a comparison of the sheath potential ϕ_sh with and without collisions. ϕ_sh in both cases is only determined by the cold initial distribution, at the beginning. Then, around $\sim$ 0.5-1 τ_e both curves rapidly increase, due to the arriving suprathermal electrons from the ELM source. ϕ_sh immediately rises to $\sim$ 3keV in the non-collisional case, where it stays mainly steady until suprathermal ions arrive at $\sim 0.5\tau_\mathrm{i}$. In the collisional case on the other hand ϕ_sh rapidly increases to $\sim$ 1.5keV and then gradually rises up to a maximum of $\sim$ 2.5keV, until the arrival of the suprathermal ions at τ_i. This indicates, that due to drag on ions or thermalization through self-collisions, the collision operator lowers the high-$v_\parallel$ tail of the velocity distribution. Differing from our results in [52] the collisional ϕ_sh mostly follows the

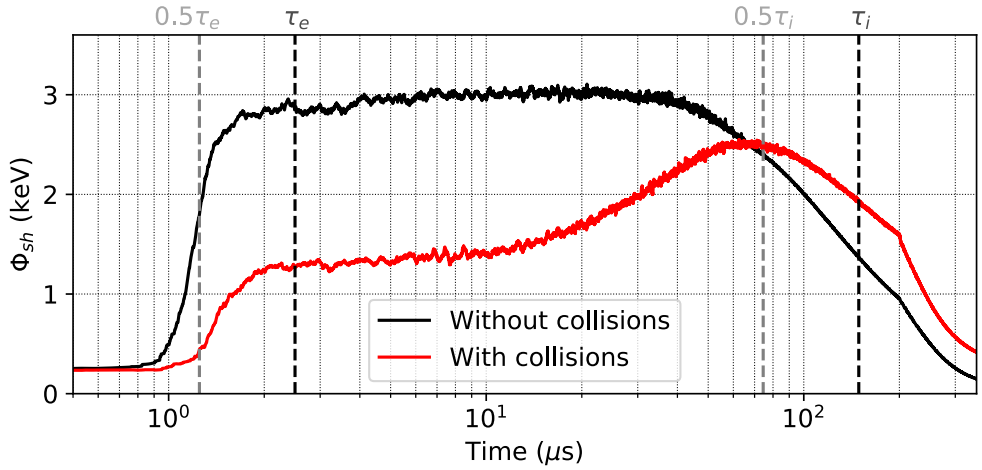


Figure 6.8: Comparison of the time-dependent evolution of the potential at the right sheath boundary in the 1D2V case with and without same-species Lenard-Bernstein collisions. Thermal ion and electron transit times $\tau_\mathrm{e} = 2.5\mu$s and $\tau_\mathrm{i} = 149\mu$s are indicated by black vertical lines ($0.5\tau_\mathrm{e}$ and $0.5\tau_\mathrm{i}$ are indicated by grey lines).

non-collisional one and already surpasses it at $\sim 20\mu$s. The sheath potential causes a deceleration and reflection of electrons at the wall in both cases. Most of the electrons are prevented from leaving the domain, which leads to a stop in the increase of the electron heat flux at $\sim 0.5\tau_\mathrm{e}$. After the inflow of suprathermal ions, the sheath potential decreases constantly and hence an increase of the ion as well as the electron heat flux can be seen. However, in the collisional case after $\sim 0.5\tau_\mathrm{i}$ until the end of the simulation ϕ_sh still remains higher. Here again the collision operator apparently is able to replenish high-$v_\parallel$ electrons through pitch-angle scattering.
As mentioned before, the simulation results shown in figures 6.7 and 6.8 only contain intraspecies collisions, i.e. electron-electron and ion-ion collisions. Therefore, in figure 6.9 we show the same collisional heat flux on the divertor as in figure 6.7 on the right, but we compare it with the heat flux of a collisional run that includes electron-ion interspecies collisions on the left. Due to their small collision frequency, ion-electron collisions are still neglected. From the plot we can see that adding electron-ion collisions hardly has an effect on the heat flux of both the electrons and the ions. Only for electrons a slight decrease of the heat flux can be seen at the beginning, until $\sim 0.5\tau_\mathrm{i}$ when the first suprathermal ions arrive at the wall. After $\sim 0.5\tau_\mathrm{i}$ both graphs show a very similar behavior.

This initial difference of both simulations can also be seen when looking at the

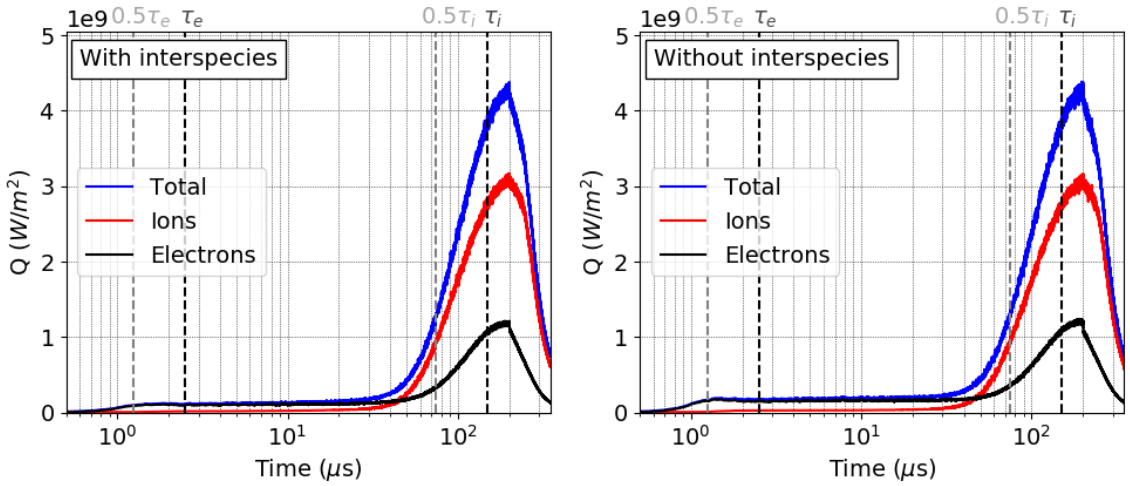


Figure 6.9: Comparison of the evolution of ion (red), electron (black) and total (blue) heat flux in the collisional 1D2V case, according to eq. (6.25), with and without electron-ion interspecies collisions. The heat flux graph without interspecies collisions on the right is exactly the same as the heat flux with collisions on the right of figure 6.9. Thermal ion and electron transit times $\tau_e = 2.5\mu s$ and $\tau_i = 149\mu s$ are indicated by black vertical lines ($0.5\tau_e$ and $0.5\tau_i$ are indicated by grey lines).

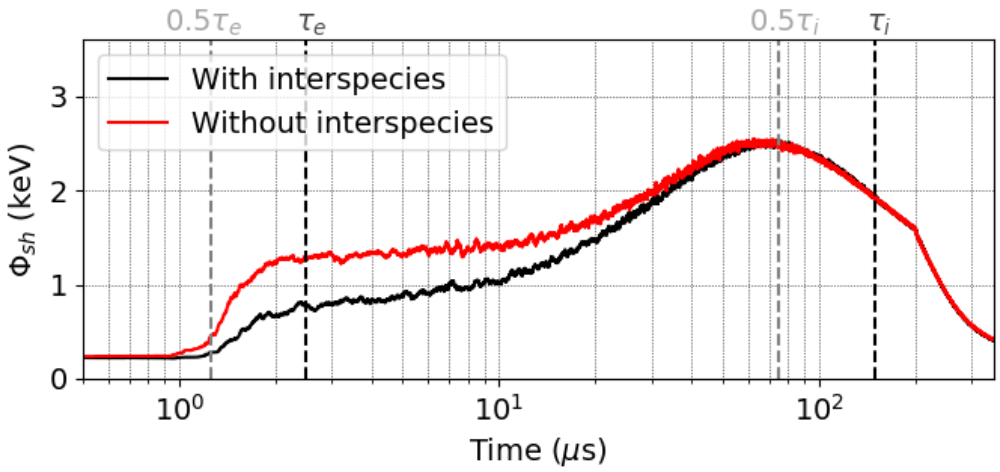


Figure 6.10: Comparison of the time-dependent evolution of the potential at the right sheath boundary in the collisional 1D2V case with and without interspecies collisions. The sheath potential without interspecies collisions (red) is exactly the same as ϕ_{sh} with collisions (red) shown in figure 6.8. Thermal ion and electron transit times $\tau_{\mathrm{e}} = 2.5\mu$s and $\tau_{\mathrm{i}} = 149\mu$s are indicated by black vertical lines ($0.5\tau_{\mathrm{e}}$ and $0.5\tau_{\mathrm{i}}$ are indicated by grey lines).

comparison of the sheath potential ϕ_{sh} for these two cases in figure 6.10. The electron-ion collisions apparently lead to a slower increase of the sheath potential until the first suprathermal ions arrive at $\sim 0.5\tau_{\mathrm{i}}$ again. This could be an indicator, that the collisions of electrons with the slower ion species leads to a deceleration of the fastest electrons and thus to a lower ϕ_{sh}. After $\sim 0.5\tau_{\mathrm{i}}$, the sheath potential is mainly determined by the ion dynamics and both plots overlap almost perfectly.

We also performed simulations where instead of using the Lenard-Bernstein collision operator for interspecies electron-ion collisions, we simply used pitch angle scattering [125, 130]. However, the differences in the heat flux and the sheath potential between both cases were only marginal and thus we do not show the results here. For the future, it could still be interesting to study the difference of both methods further.

6.6 Conclusions

As a first test case for PICLS, we implemented a well-studied 1D ELM heat pulse problem for the collisionless ([115], [94]), as well as collisional case ([52]). Here, a central heat pulse that propagates along the field line towards the divertor target was modeled and for the collisions, a Lenard-Bernstein collision operator was implemented.

We already published most of the results in this chapter previously (for more details see [102] and [127]).
In the non-collisional case, the results we achieved for the heat flux on the divertor are consistent with previous simulation results of fully kinetic continuum, PIC, fluid and gyrokinetic continuum codes. Our results could reproduce the key aspects of the previous findings: the ELM heat flux loading on the divertor mainly happens on the ion transit time scale; deceleration (acceleration) of electrons (ions) by the sheath causes an asymmetric heat flux on the divertor of electrons and ions; the majority of electrons are confined due to a strong negative potential that builds on the electron transit time. Because of the code specific particle source implementation, a differences in the peak values occur compared to previous gyrokinetic continuum simulations. In addition to previous results, it was shown that varying $k_\perp \rho$ mainly affects the electric potential and therewith the ion polarization. However, the heat flux on the divertor and the sheath potential are hardly affected (at least for values of $1.0 \geq k_\perp \rho \geq 0.05$).
In the collisional case, we performed 1D2V heat pulse simulations and compared the results with non-collisional results. The effect of the collisions on the heat flux deposited on the sheath, lead to a total increase of 9%. Due to collisions, also the sheath potential ϕ_{sh} is changed and thus shows a deferred increase and lower maximum value of ~ 2.5keV (compared to ~ 3.0keV). Including interspecies electron-ion collisions, however, only had a very mild effect on the heat flux and ϕ_{sh} at electron time scales, but the total heat flux over time was hardly affected.
Due to the implementation of the non-linear Poisson equation (6.8) only a single $k_\perp$ mode was assumed. We were tempted to re-run the simulation setup with the more accurate linearized Polarization equation (2.40), which would account for the coupling of several $k_\perp$ modes. For more realistic edge plasma simulations or real 3D devices (such as linear experiments or tokamaks) a coupling of different $k_\perp$ modes is essential. However, just expanding the given test case to three spatial dimensions would not provide significant gradients in x and y. Thus, also the Polarization equation cannot be used as a field solver in such a system. Hence, we decided to move on to simulate more appropriate and realistic setups with the full 3D code setup. The collision operator which we investigated in this section is also an important ingredient for simulations with more than one dimension.

7 ITG simulations in screw pinch

Going from a 1D to a 3D spatial domain is not trivial and requires careful implementation and testing. Several main aspects of our PIC algorithm, especially the field solver and the particle pusher, need to be adjusted to allow for higher dimensional simulations. Therefore, we initially want to study a known problem of ITG (Ion Temperature Gradient) instabilities in a screw pinch geometry (see for example [149]) and test our results against the pre-existing results. As mentioned in section 5.2.4, the screw pinch is a cylindrical pinch device with the coordinates (r, θ, z), where r is the radius, θ the poloidal angle and z the coordinate along the cylindrical axis (or toroidal direction). Its name comes from the helical shape of the magnetic field with components in the z and θ direction. The electric potential solver we use for the screw pinch is the 2D1D solver for tokamak geometries described in section 5.2.4, with an FFT in the periodic toroidal z-direction, periodic boundary conditions in the poloidal direction and non-periodic Dirichlet zero boundary conditions in the radial direction. The particles leaving the domain in radial direction are reflected back into the domain. Thus, no sheath boundary conditions will be studied within this chapter.
In section 7.1, we will give further details on the physical model regarded within these simulations. Here, we will focus on the setup of the magnetic field components (see 7.1.1) and a theoretical background on ITG modes in slab geometry (see 7.1.2) as studied in this section. The specific setup of our test case is mainly based on a study done by Brunner within his PhD thesis (see [149]) and will be presented in section 7.2. In section 7.3, we will present the results for the particle trajectory and potential solver test, as well as the slab-ITG simulations. The results will be shortly summarized in section 7.4. This is a valuable test case to study the key features of PICLS' 3D code setup and delivers an interesting full-f ITG-slab study in screw pinch geometry.

7.1 Model details

In this section we want to focus on two important aspects of the model used for the ITG-slab instability studies in screw pinch geometry that we simulate in this chapter. First, we introduce the specific B-field with its helical shape, due to components in the z and θ direction. We also discuss all necessary derivatives to achieve the required

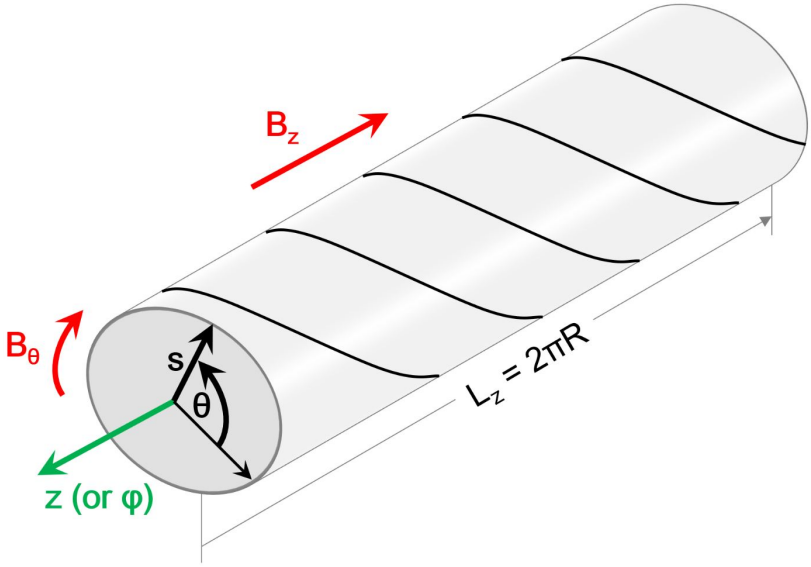


Figure 7.1: Illustration of the screw pinch geometry with the shape of the helical field lines sketched by black lines.Where s is the radius r normalized to the minor radius a of the pinch.

Euler-Lagrange equations. In a second step, we focus on theoretical aspects of ITG-slab instabilities, which we will mostly base on the work of Brunner ([149]).

7.1.1 B-field and Euler Lagrange equations

An important aspect of the screw pinch is its helical shape of the magnetic field with components in the z and θ direction (see also figure 7.1):

$$\mathbf{B} = B_0(-\frac{r}{Rq_s(r)}\boldsymbol{\theta} + \mathbf{z}), \tag{7.1}$$

with the radius r, the q-factor $q_s(r) = \alpha + \beta(r/a)^2$, the normalized radius $s = r/a$, the constant magnetic field B_0 and $R = \frac{L_z}{2\pi}$ (L_z is the length of the cylinder in z-direction). The coordinates r and θ can also be expressed in Cartesian coordinates by:

$$r^2 = (x - x_0)^2 + (y - y_0)^2, \tag{7.2}$$

$$\theta = \tan^{-1}(\frac{y - y_0}{x - x_0}), \tag{7.3}$$

$$\boldsymbol{\theta} = -\sin(\theta)\mathbf{x} + -\cos(\theta)\mathbf{y} \tag{7.4}$$

and the derivatives of r and θ can be expressed as:

$$\begin{aligned}
\frac{dr}{dx} &= \frac{x - x_0}{r}, && (7.5)\\
\frac{dr}{dy} &= \frac{y - y_0}{r}, && (7.6)\\
\frac{d\theta}{dx} &= -\frac{y - y_0}{r^2}, && (7.7)\\
\frac{d\theta}{dy} &= \frac{x - x_0}{r^2}. && (7.8)
\end{aligned}$$

With the definition:

$$g(r) = -\frac{rB_0}{Rq_{\mathrm{s}}(r)}, \tag{7.9}$$

we can rewrite the magnetic field $\mathbf{B}$ from equation (7.1) as:

$$\mathbf{B} = \sin(\theta)g(r)\mathbf{x} + \cos(\theta)g(r)\mathbf{y} + B_0\mathbf{z}. \tag{7.10}$$

The magnetic field is required for the calculation of the Euler-Lagrange equations for the screw pinch case, which can be written as:

$$\begin{aligned}
\dot{\mathbf{R}} &= v_\parallel \frac{\mathbf{B}^*}{B_\parallel^*} + \frac{c}{e_p B B_\parallel^*}\mathbf{B} \times \nabla B,\\
\dot{v_\parallel} &= -\frac{\mathbf{B}^*}{B_\parallel^*} \cdot \frac{\mu}{m}\nabla B. && (7.11)
\end{aligned}$$

Here we used the previous definitions for $\mathbf{B}^*$ and $B_\parallel^*$:

$$\begin{aligned}
\mathbf{B}^* &= \mathbf{B} + \frac{m_p c}{e_p} v_\parallel \nabla \times \mathbf{b}, && (7.12)\\
B_\parallel^* &\equiv \mathbf{B}^* \cdot \mathbf{b} = B + \frac{c m_p}{e_p} v_\parallel \nabla \times \mathbf{b} \cdot \mathbf{b}. && (7.13)
\end{aligned}$$

By using $\frac{\nabla \times (B\mathbf{b})}{B} = \nabla \times \mathbf{b} + \frac{\nabla B \times \mathbf{b}}{B}$, we can rewrite equation (7.12) as:

$$\mathbf{B}^* = \mathbf{B} + \frac{m_p c}{e_p} v_\parallel \left(\frac{\nabla \times \mathbf{B}}{B} + \frac{\mathbf{B} \times \nabla B}{B^2} \right). \tag{7.14}$$

To calculate the required quantities for

$$\begin{aligned}
\nabla \times \mathbf{B} &= \mathbf{x}\left(\frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z}\right) - \mathbf{y}\left(\frac{\partial B_z}{\partial x} - \frac{\partial B_x}{\partial z}\right) \\
&\quad + \mathbf{z}\left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}\right), \\
\nabla B &= \frac{\partial B}{\partial x}\mathbf{x} + \frac{\partial B}{\partial y}\mathbf{y} + \frac{\partial B}{\partial z}\mathbf{z}, \\
\mathbf{B} \times \nabla \times \mathbf{B} &= \mathbf{B} \times (\nabla \times \mathbf{B}),
\end{aligned} \tag{7.15}$$

first the relevant derivatives need to be calculated. We start with the derivatives of B, using that $B^2 = g^2(r) + B_0^2$ to obtain:

$$\frac{dB}{dx} = \frac{dB}{dr}\frac{dr}{dx} = 2g(r)\frac{dg}{dr}\frac{dr}{dx} = \frac{x - x_0}{Br} g g', \tag{7.16}$$

$$\frac{dB}{dy} = \frac{y - y_0}{Br} g g', \tag{7.17}$$

$$\frac{dB}{dz} = 0, \tag{7.18}$$

with

$$g_s'(r) = \frac{B_0}{Rq_\mathrm{s}} + \frac{rB_0}{Rq_\mathrm{s}^2} q_\mathrm{s}', \tag{7.19}$$

$$q_\mathrm{s}'(r) = 2r. \tag{7.20}$$

Using equations (7.5, 7.6, 7.7, 7.8), the derivatives of the x, y and z components of the magnetic field $B_x = -\sin(\theta)g(r)$, $B_y = \cos(\theta)g(r)$ and $B_z = B_0$ can be calculated:

$$\begin{aligned}
\frac{dB_x}{dx} &= \frac{dB_x}{d\theta}\frac{d\theta}{dx} + \frac{dB_x}{dr}\frac{dr}{dx} = \cos(\theta)g(r)\frac{y-y_0}{r^2} - \sin(\theta)g'(r)\frac{x-x_0}{r}, && (7.21)\\
\frac{dB_x}{dy} &= -\cos(\theta)g(r)\frac{x-x_0}{r^2} - \sin(\theta)g'(r)\frac{y-y_0}{r}, && (7.22)\\
\frac{dB_x}{dz} &= 0, && (7.23)\\
\frac{dB_y}{dx} &= \sin(\theta)g(r)\frac{y-y_0}{r^2} + \cos(\theta)g'(r)\frac{x-x_0}{r}, && (7.24)\\
\frac{dB_y}{dy} &= \sin(\theta)g(r)\frac{x-x_0}{r^2} + \cos(\theta)g'(r)\frac{y-y_0}{r}, && (7.25)\\
\frac{dB_y}{dz} &= 0, && (7.26)\\
\frac{dB_z}{dx} &= 0, \quad \frac{dB_z}{dy} = 0, \quad \frac{dB_z}{dz} = 0. && (7.27)
\end{aligned}$$

Inserting these derivatives in equations (7.15), we obtain:

$$\begin{aligned}
\nabla \times \mathbf{B} &= \mathbf{z}\left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}\right) \equiv C_z\mathbf{z},\\
\nabla B &= \frac{\partial B}{\partial x}\mathbf{x} + \frac{\partial B}{\partial y}\mathbf{y} \equiv G_x\mathbf{x} + G_y\mathbf{y},\\
\mathbf{B} \times \nabla \times \mathbf{B} &= B_y C_z \mathbf{x} - B_x C_z \mathbf{y}, && (7.28)
\end{aligned}$$

which can finally be used to calculate the required expression for B^* in eq. (7.12).

7.1.2 ITG instabilities in slab geometry

As mentioned earlier, the theoretical considerations on ITG modes within this section are based on the work of Brunner [149], for more details please refer to their publication. We are considering a slab geometry without toroidal effects to investigate the destabilizing effects of ion temperature gradients (ITG). A local dispersion relation can be derived by using a single plane wave component:

$$\phi = \hat{\phi}\exp\left(\mathbf{k}\cdot\mathbf{r} - \omega t\right). \tag{7.29}$$

The gyrocenter trajectories reduce to the parallel motion for the ions and thus the gyrokinetic equations can be solved. With the definitions:

$$\omega^* = \omega_\mathrm{n}(1 + \eta T\partial/\partial T) \quad \text{and} \quad \eta = d\ln T/d\ln n. \tag{7.30}$$

we can obtain:

$$\bar{g}\mathrm{i} = \frac{e}{T_\mathrm{i}}\frac{\omega - \omega_\mathrm{i}^*}{\omega - k_\parallel v_\parallel} f_{\mathrm{M,i}} J_0\left(\frac{k_\perp v_\perp}{\Omega_\mathrm{i}}\right)\hat{\phi}\exp\left(\mathbf{k}\cdot\mathbf{r} - \omega t\right). \tag{7.31}$$

Respectively, the diamagnetic drift frequencies related to density and temperature gradients can be written as:

$$\omega_\mathrm{n} = \frac{T}{qB}\frac{d\ln n}{dx}k_y \quad \text{and} \quad \omega_\mathrm{T} = \eta\omega_\mathrm{n} = \frac{1}{qB}\frac{dT}{dx}k_y. \tag{7.32}$$

By assuming that in the standard case n and T decrease along $\mathbf{e_x}$, we can define the characteristic lengths:

$$L_\mathrm{n} = -\left(\frac{d\ln n}{dx}\right)^{-1} \quad \text{and} \quad L_\mathrm{T} = -\left(\frac{d\ln T}{dx}\right)^{-1}. \tag{7.33}$$

The ion density can be obtained by replacing $\exp\left(i\mathbf{k}\cdot\mathbf{r}\right)$ by $\exp\left(i\mathbf{k}\cdot\left(\mathbf{r} + \mathbf{v}\times\mathbf{e}_\parallel/\Omega_\mathrm{i}\right)\right)$, which means we are returning to the particle variables:

$$\bar{n}_\mathrm{i} = \int d\mathbf{v}\bar{f}_\mathrm{i} = -\frac{ne}{T_\mathrm{i}}\left[1 - \int d\mathbf{v} J_0^2\left(\frac{k_\perp v_\perp}{\Omega_\mathrm{i}}\right)\frac{\omega - \omega_\mathrm{i}^*}{\omega - k_\parallel v_\parallel}\frac{f_{\mathrm{M,i}}}{n}\right], \tag{7.34}$$

with $d\mathbf{v} = 2\pi v_\perp dv_\perp dv_\parallel$. The integration over the gyroangle led to a second Bessel function J_0. The term $-Nq\phi/T$ corresponds to the adiabatic response. Now, we need to introduce the scaled modified Bessel function $\Lambda_p(x) = e^{-x}I_p(x)$ to write the integrals over $v_\parallel$ and $v_\perp$, and formulate the plasma dispersion function $W(z)$:

$$W(z) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{+\infty}\frac{x}{x - z}\exp\left(-\frac{x^2}{2}\right)dx, \quad \Im(z) > 0. \tag{7.35}$$

Hence, by considering adiabatic electrons, the quasi-neutrality relation gives the required dispersion relation:

$$\begin{aligned} 0 &= \frac{1}{\tau} + 1 + \left(1 - \frac{\omega_{\mathrm{i}}^*}{\omega}\right)(W-1)\Lambda_0 \\ &= \frac{1}{\tau} + 1 + \left[1 - \frac{\omega_{\mathrm{n,i}}}{\omega}\left(1 - \frac{\eta_{\mathrm{i}}}{2}\right)\right](W-1)\Lambda_0 \\ &\quad - \frac{\omega_{\mathrm{n,i}}}{\omega}\eta_{\mathrm{i}}\left[\frac{z_{\mathrm{i}}^2}{2}W\Lambda_0 + (W-1)\zeta(\Lambda_1 - \Lambda_0)\right], \end{aligned} \tag{7.36}$$

with the notations $\tau = T_e/T_i$ for the temperature ratio, $\zeta = (k_\perp \lambda_{\mathrm{L}})^2$, $z = \omega/|k_\parallel| v_{\mathrm{th}}$, $\Lambda_p = \Lambda_p(\zeta_{\mathrm{i}})$ and $W = W(z_i)$. Following [149], for cold ions the dispersion relation 7.36 can be further simplified. For $|k_\parallel c_{\mathrm{s}}| \gg \omega_{\mathrm{n,e}}$ the ion-acoustic or ion sound wave can be obtained:

$$\omega^2 \simeq \frac{(k_\parallel c_{\mathrm{s}})^2}{1 + (k_\perp \lambda_{\mathrm{L}}^*)^2}, \tag{7.37}$$

where $c_{\mathrm{s}}^2 = T_{\mathrm{e}}/m_{\mathrm{i}}$ is the sound velocity squared and $\lambda_{\mathrm{L}}^* = c_{\mathrm{s}}/\Omega_{\mathrm{i}}$ the ion Larmor radius. For $|k_\parallel c_{\mathrm{s}}| \ll \omega_{\mathrm{n,e}}$ one of the two roots becomes:

$$\omega \simeq \omega_{\mathrm{n,e}}, \tag{7.38}$$

which is the electron drift mode. This mode can be destabilized by non-adiabatic electron dynamics.
In a second case, a finite inhomogeneous ion temperature with $T_{\mathrm{i}} \simeq T_{\mathrm{e}}$ is considered. Here, an ion driven instability can arise. Following the considerations of [149] again and assuming a flat density profile yields:

$$1 - \left(\frac{k_\parallel v_{\mathrm{th,i}}}{\omega}\right)\left(1 - \frac{\omega_{\mathrm{T,i}}}{\omega}\right) = 0. \tag{7.39}$$

For $|k_\parallel v_{\mathrm{th,i}}| \ll \omega_{\mathrm{T,i}}$, this relation contains the slab-ITG instability or η_{i}-instability, an unstable root $\omega = \omega_{\mathrm{r}} + i\gamma$, with $\gamma > 0$, which can be written as:

$$\omega \simeq 1/2(1 + i\sqrt{3})\left[(k_\parallel v_{\mathrm{th,i}})^2 \omega_{\mathrm{T,i}}\right]^{1/3}. \tag{7.40}$$

At the limit of applicability of this result we get $\omega_{\mathrm{r}} \simeq \gamma \simeq \omega_{\mathrm{T,i}}$. However, accurate results can only be obtained by solving the full kinetic dispersion relation (7.36), since maximal growth rates appear for longitudinal phase velocities of the order $v_{\mathrm{th,i}}$ and

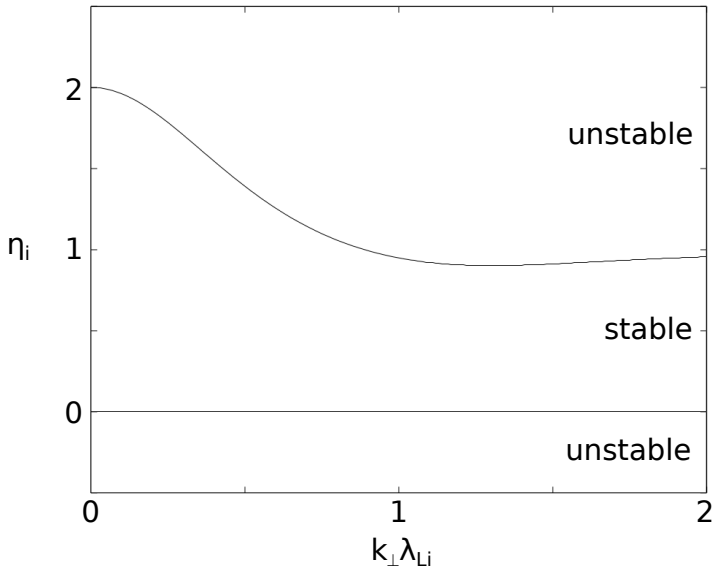


Figure 7.2: Instability condition of η_i as a function of $k_\perp \lambda_\mathrm{L,i}$ for the slab-ITG instability.

perpendicular wavelengths of the order $\lambda_\mathrm{L,i}$.
It can be shown that two instability conditions exist at finite values of η_i. The first one is for η_i, which gives for arbitrary values of $k_\perp \lambda_\mathrm{L,i}$:

$$\eta_\mathrm{i} > \eta_\mathrm{c,i} = \frac{2}{1 + 2\zeta_\mathrm{i}(\Lambda_0 - \Lambda_1)/\Lambda_0} \quad \text{or} \quad \eta_\mathrm{i} < 0. \tag{7.41}$$

This condition is plotted in figure 7.2 as a function of $k_\perp \lambda_\mathrm{L,i}$.
The second condition is on $k_\parallel$. Here, for $|k_\parallel| < k_{\parallel,\mathrm{lim}}$ with $k_\perp \lambda_\mathrm{L,i} \ll 1$ we can get:

$$k_{\parallel,\mathrm{lim}} = \frac{1}{2}\left(1 - \frac{2}{\eta_\mathrm{i}}\right)^{1/2} \frac{|\omega_\mathrm{T,i}|}{v_\mathrm{th,i}} = \frac{1}{2}\left(1 - \frac{2}{\eta_\mathrm{i}}\right)^{1/2} \frac{\lambda_\mathrm{L,i}}{L_{T,\mathrm{i}}} k_y. \tag{7.42}$$

However, the instability is suppressed by Landau damping for the case $|k_\parallel| > k_{\parallel,\mathrm{lim}}$. As shown in [149], γ and $k_{\parallel,\mathrm{lim}}$ both have their maximum for $k_\perp \lambda_\mathrm{L,i} \simeq 1$. For even shorter wavelengths the drive becomes less dominant since the potential acting on the ions tends to be averaged out over their Larmor gyration.
By regarding a circular cross-section equilibrium of (similar to (7.1)):

$$\mathbf{B} = B_0 \frac{R}{r}\left(-\frac{\rho}{R q_\mathrm{s}} \mathbf{e}_\theta + \mathbf{e}_\phi\right), \tag{7.43}$$

with the q-profile q_s and the toroidal radius ρ, we can get an understanding of the structure of such an eigenmode. Due to the symmetry of the equilibrium, we have a fixed toroidal mode number n. However, due to the weak toroidal coupling such a mode also contains a dominant poloidal mode number m, which gives:

$$\phi \simeq \phi(\rho) \exp[i(m\theta + n\phi)]. \tag{7.44}$$

Following [149], the parallel wave number can be obtained:

$$\mathbf{e}_\parallel \nabla\phi \simeq i\frac{1}{Rq_\mathrm{s}}(nq_\mathrm{s} - m)\phi \quad \Rightarrow \quad k_\parallel(\rho) = \frac{1}{Rq_\mathrm{s}(\rho)}(nq_\mathrm{s}(\rho) - m). \tag{7.45}$$

Since $|k_\parallel| < k_{\parallel,\mathrm{lim}}$ holds, such a modes shows a tendency to be localized around the corresponding mode rational surface $\rho = \rho_r = \text{const}$. Here, $q_\mathrm{s}(\rho_r) = m/n$ and as a result $k_\parallel(\rho_r) = 0$. At a distance of $\delta\rho$ from this surface for $k_\parallel$ holds $|k_\parallel| = \delta\rho|k_\theta|/L_s$, with the shear length $L_s = Rq_\mathrm{s}/\bar{s}$ and the poloidal mode number $k_\theta = m/\rho$. The radial extent of such a mode for $k_\perp\lambda_\mathrm{L,i} \ll 1$, can be estimated from equation (7.42) and by using that here $\omega_\mathrm{T} = -Tk_\theta/qB_0L_\mathrm{T}$ holds, to obtain:

$$\delta\rho \simeq 1/2\left(1 - \frac{2}{\eta_\mathrm{i}}\right)^{1/2} \frac{L_s}{L_{T,\mathrm{i}}}\lambda_\mathrm{L,i}. \tag{7.46}$$

Two such modes with same toroidal wave number n that are centered on neighboring rational surfaces can overlap. Where in the fluid regime $\eta_\mathrm{i} \gg \eta_\mathrm{c,i}$ the condition for this overlap is:

$$\delta\rho \gtrsim \frac{1}{nq_\mathrm{s}} \quad \Rightarrow \quad \epsilon_{T,\mathrm{i}} \lesssim \tilde{k}_\theta q_\mathrm{s}, \tag{7.47}$$

with $\tilde{k}_\theta = k_\theta\lambda_\mathrm{L,i}$ and $\epsilon_T = L_\mathrm{T}/R$. An interesting result of this is that for one fixed toroidal wave number many independent modes can form a fluctuating structure with a large radial extent that spans over the corresponding mode rational surfaces.

7.2 Simulation setup

As mentioned earlier, for the simulations in this chapter the cylindrical coordinates (r, θ, z) are used and the 2D1D solver for tokamak geometries described in section 5.2.4 is applied as potential solver. Here, the two derivatives are within the poloidal plane and do not affect the FFT in the z-direction.
The marker weights are chosen to be constant, which comes from the application of

PICLS to slab and SOL simulations. For cylindrical test cases, as present in this chapter, this choice is not optimal since in the center of the cylinder the size of the cells and thus also the number of markers per cell decreases. For our current simulations this is not a significant problem, but requires a higher total number of markers to achieve a sufficient resolution in the center of the cylinder. This could be changed in the future by varying the marker weights and increasing the number of markers in the center compared to the outer region.
For our slab-ITG simulations, we again used a very similar setup as the one described in [149]. There, a hydrogen plasma is chosen with the following parameters: minor radius $a = 0.2\text{m}$, major radius $R = 1\text{m}$, magnetic field on axis $B_0 = 2\text{T}$, safety factor profile $q_\text{s}(s) = 1 + 2s^2$, density profile $n(s)/n_0 = 1 - s^2$, ion temperature profile $T_\text{i}(s)/T_\text{i0} = (1-s^2)^3$, electron temperature profile $T_\text{e}(s)/T_\text{e0} = (1-s^2)^2$, temperatures on axis $T_\text{e0} = T_\text{i0} = 2\text{keV}$. With $s = r/a$ the normalized radial variable is defined. The average ion Larmor radius is $\lambda_\text{L,i} \simeq 1.5 \cdot 10^{-3}\text{m}$ to obtain $\epsilon = \lambda_\text{L,i}/a \simeq 7.5 \cdot 10^{-3}$.
Within our simulations, however, we chose a deuterium plasma and changed the minor and major radius to $a = 13.6\text{cm}$ and $R = 10\text{m}$. For the density we use a flat profile and for the safety factor profiles we use flat ones as well as profiles of the form $q_\text{s}(s) = 1 + 3s^2$. Also the temperature profiles differ from the ones in [149], as for the simulations within this section the following profiles are used:

$$T_\text{e/i} = B \cdot \exp(A \cdot C), \tag{7.48}$$

where A, B and C are defined as:

$$\begin{aligned} A &= -3/\left(1 - \cosh(0.5/0.1)^{-2}\right), \\ B &= \left[\cosh\left(\frac{s-0.5}{0.1}\right)^{-2}\right]^{(B\cdot 0.1^2)}, \\ C &= -\left(s^2 - 0.5^2\right)\left(1 - \cosh(0.5/0.1)^{-2}\right) + 2 \cdot 0.1 \tanh\left(\frac{s-0.5}{0.1}\right) s. \end{aligned}$$

The reason for choosing different parameters is that we wanted to compare our simulations with simulations of the delta-f code ORB5. ORB5 cannot be run with a screw pinch geometry, but only with a tokamak geometry. Thus, we especially had to adjust the major radius R, to achieve a comparable setup.
With this model, slab-like ITG instabilities as described in section 7.1.2 can evolve within the plasma. Thus, the plasma is locally unstable for fixed mode numbers (m, n) if around the respective mode rational surface $\eta_\text{i} > \eta_\text{c,i} \simeq 1$ holds. Since $\eta_\text{i} \equiv 3$, as a result, at any mode rational surface the plasma is unstable.

For these simulations, right at the beginning at the first time step an initial perturbation is applied to drive turbulence in the plasma. Therefore, we use the mass matrix to project an analytical potential onto the potential grid and thus apply a force onto the initial particle distribution. In the code an option is implemented which allows to "freeze" this pre-defined potential and apply it to the particles at each time step. This is a helpful tool to analyze the validity of the particle trajectories and thereby test the particle pusher routines (see 7.3.1). Note that no collisions are applied for any of the simulations within this chapter.

7.3 Simulation results

As mentioned earlier, for extending PICLS towards three spatial dimensions, several features had to be adjusted. We therefore want to use the cylindrical setup presented in this chapter to check the correct functionality of two main code blocks, the particle pusher and the field solver. Therefore, we will start with presenting dedicated tests for the particle trajectories and the field solver. With the gained confidence we then focus on the actual ITG instability simulations.

7.3.1 Particle trajectory tests

Within a PIC code, the Euler-Lagrange equations are solved at every time step of the simulation to evolve the particle trajectories. The main routine involved here is the particle pusher. To test the particle trajectories, we therefore use an option implemented in PICLS, where the potential and thereby the field is fixed throughout the simulation. Thus, at the beginning of the simulation a potential is constructed and "frozen" to act on the particles. For this test case, a realistic electron/ion mass ratio was used and a flat q-profile of $q_\mathrm{s} = 2$ was chosen.

Figure 7.3 shows a poloidal plane of this potential. The 8 maxima and minima clearly indicate the poloidal mode number of $m = 8$. The red (blue) line represents the particle trajectory of one single electron (ion) in poloidal direction. The exact same potential is also shown in figure 7.4 for a toroidal plane. Here, the toroidal wave number is $n = 4$ as visible by the four maxima/minima. The red (blue) line again represents the particle trajectory of one single electron (ion) in toroidal direction. Note that the simulation domain is periodic along the toroidal z-direction.

To further study the electron and ion trajectories, we will in the following discuss their evolution along different spatial and velocity directions. In figure 7.5 we therefore focus on the evolution of the trajectory of one single ion along r, $v_\parallel$ and z with time (in the figure time steps are used). Without an electric field, particles would be bound to a

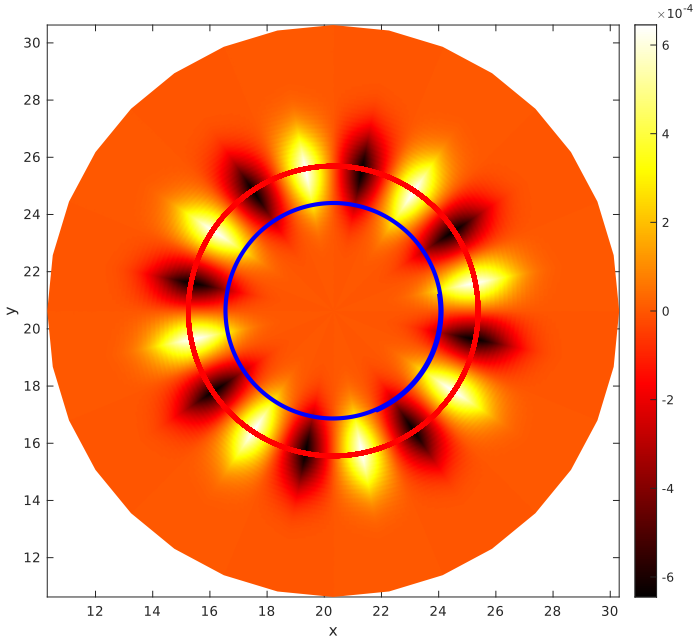


Figure 7.3: Initialized field acting on the particle trajectories in the poloidal plane. The red line corresponds to the electron trajectory and the blue line to the ion trajectory.

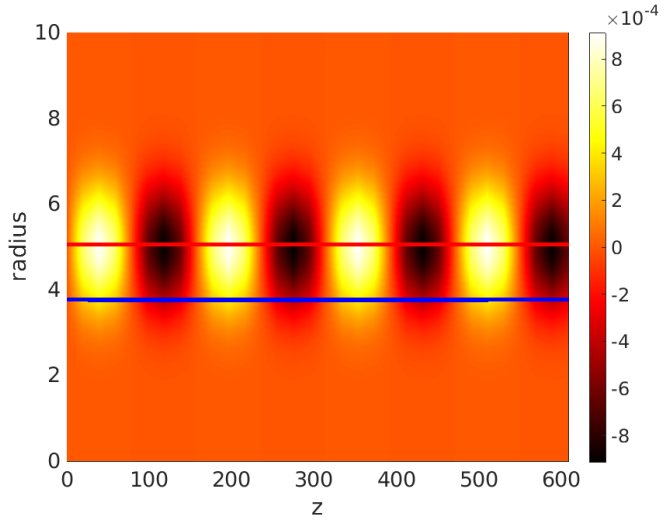


Figure 7.4: Initialized field acting on the particle trajectories in the toroidal plane. The red line corresponds to the electron trajectory and the blue line to the ion trajectory.

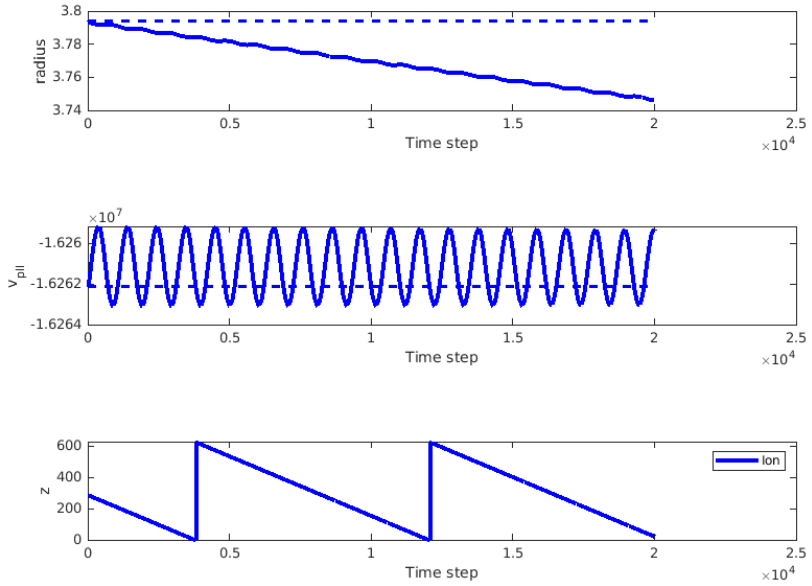


Figure 7.5: Trajectory of a single ion within the initialized field over time (here shown in time steps) for the three coordinates $(r, v_{\parallel}, z)$. The dashed lines show the trajectory of the same ion without an electric field.

gyrating motion on a circular trajectory around the center of the cylinder. From the plot at the top, however, we see a clear drift of the ions in radial direction, which is caused by the $E \times B$ drift related to the "frozen" field. Note that the $E \times B$ drift has two components, an oscillatory one related to poloidal variation of the electric field and a secular drift proportional to the toroidal component of the potential in the presence of a poloidal magnetic field. Comparing this radial trajectory with the trajectory in z (bottom plot), we see that for each time the ion crosses the domain in z-direction, 8 plateaus and slopes appear in the radial trajectory. This exactly matches the $n = 4$ structure of the potential. The same behavior can be detected in the $v_{\parallel}$-direction (middle plot), due to the so called parallel nonlinearity in the evolution equation for $v_{\parallel}$. Here, a periodic sinusoidal change of velocity appears with exactly 8 maxima for one toroidal crossing.
For the electrons in general the same applies as described for the ions only on a faster time scale. In figure 7.6 we therefore plotted the evolution of the trajectory of one single electron along r, $v_{\parallel}$ and z with time. Note that here the time scale is different, since much less time steps are required for the electron to cross a toroidal plane. Apart from this the electron trajectory is affected by the "frozen" field in a very similar way as the previously shown ion trajectory.

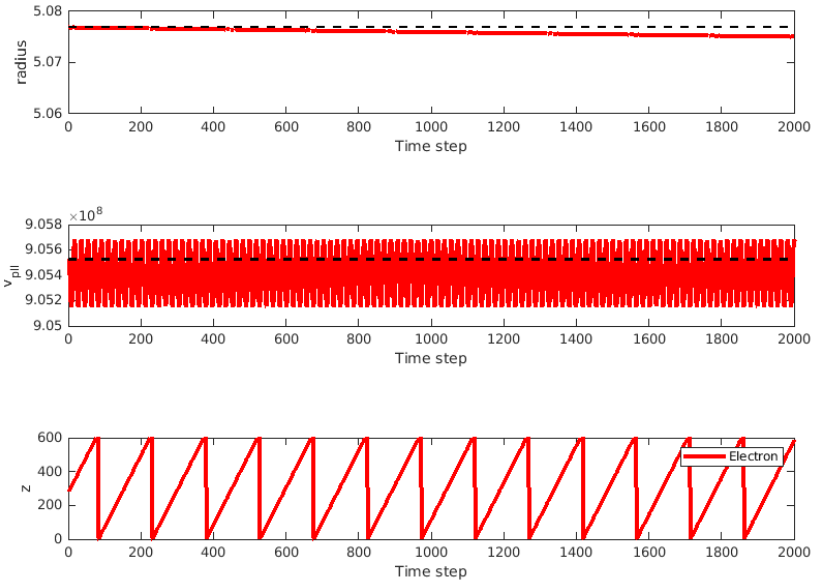


Figure 7.6: Trajectory of a single electron within the initialized field over time (here shown in time steps) for the three coordinates $(r, v_{\parallel}, z)$. The dashed lines show the trajectory of the same electron without an electric field.

With these results we are confident that the particle pushing routines work correctly in the 3D setup.

7.3.2 Potential solver test

Apart from the particle pusher, another decisive building block in a PIC code is the potential solver for the fields acting on the particles. As mentioned, for the three-dimensional simulations in this chapter we use a 2D1D solver for tokamak geometries as described in section 5.2.4. The approach we use to verify our implemented solver is based on the Method of Manufactured Solutions (MMS) [150]. With this technique a code can be verified even without analytical results. Therefore, in general a known or manufactured solution is provided e.g., by adding source terms to the equations. The deviation of the numerical solution of the equation compared to the provided manufactured solution is then checked.
In our specific case, this method is slightly adapted and applied to check, whether the solver routines calculate a correct potential from a given charge distribution. Therefore, we perform the following procedure:

1. First a specific output potential is constructed. This can be of any type, no

matter if analytical function or a specifically constructed solution.

2. On this solution the M-matrix is applied to retain the Fourier coefficients of the potential (see 5.2.4).

3. Multiplying the non-factorized Poisson matrix $A_{j'k'jk}$ with the Fourier coefficients of the matrix then gives the corresponding right-hand side (RHS) to the constructed solution.

4. Finally the normal solver routine is applied to the derived RHS to get the solved potential.

The constructed potential is then compared with the solved one to study the similarity of both and thus identify potential deviations. In figure 7.7 the constructed and solved potentials are plotted in the poloidal, toroidal and r/θ-plane for a potential with wave numbers $m = 8$ and $n = 4$. Both potentials show a good overlap in all three planes and give us confidence of the solver's validity. Obviously, for further verification the error between the two potentials could be compared for and a convergence study for increasing number of cells could be done. But this must be postponed to the future.

7.3.3 Slab-ITG simulations

After successfully testing two key features, the particle pusher and the field solver, the code can now be applied to the slab-ITG instability studies. The setup described in 7.2 was applied with a q-profile of $q_\mathrm{s}(s) = 1 + 3s^2$. We tried several different mass ratios with similar results. Thus, in the following we will show plots of a deuterium plasma with $m_\mathrm{i} = 3670$ electron masses and for $m_\mathrm{e} = m_\mathrm{i}/100$ and $m_\mathrm{e} = m_\mathrm{i}/40$. In addition, for all simulations within this section, we filtered out modes with $n > 10$. Thus, only modes with mode numbers from 0 to 10 are included.
Therefore, in figure 7.8 we show the profiles in radial direction (with the normalized radius r/a) for density n, temperature and pressure and the particles (or markers) per cell for $a/L_\mathrm{T} = 8$ and $m_\mathrm{i}/m_\mathrm{e} = 40$. These profiles are calculated from the initial particle distribution in phase-space and coincide with the profiles described in 7.2. For the density n we clearly see the flat distribution in the radial direction. The increase of particles per cell with the radius comes from the radially increasing cell size in the cylindrical setup. The pressure shows the same shape as the temperature, as expected.
In figure 7.9 the profile of the electron temperature T_e as a function of r/a is now regarded for different times for $m_\mathrm{i}/m_\mathrm{e} = 40$. The initial T_e profile at $t = 0\mu$s (blue line) is equal to the T_e profile in figure 7.8. T_e is quickly relaxing with time and after $\sim 0.13\mu$s (yellow line) already shows an almost flat shape. At this point, the initial

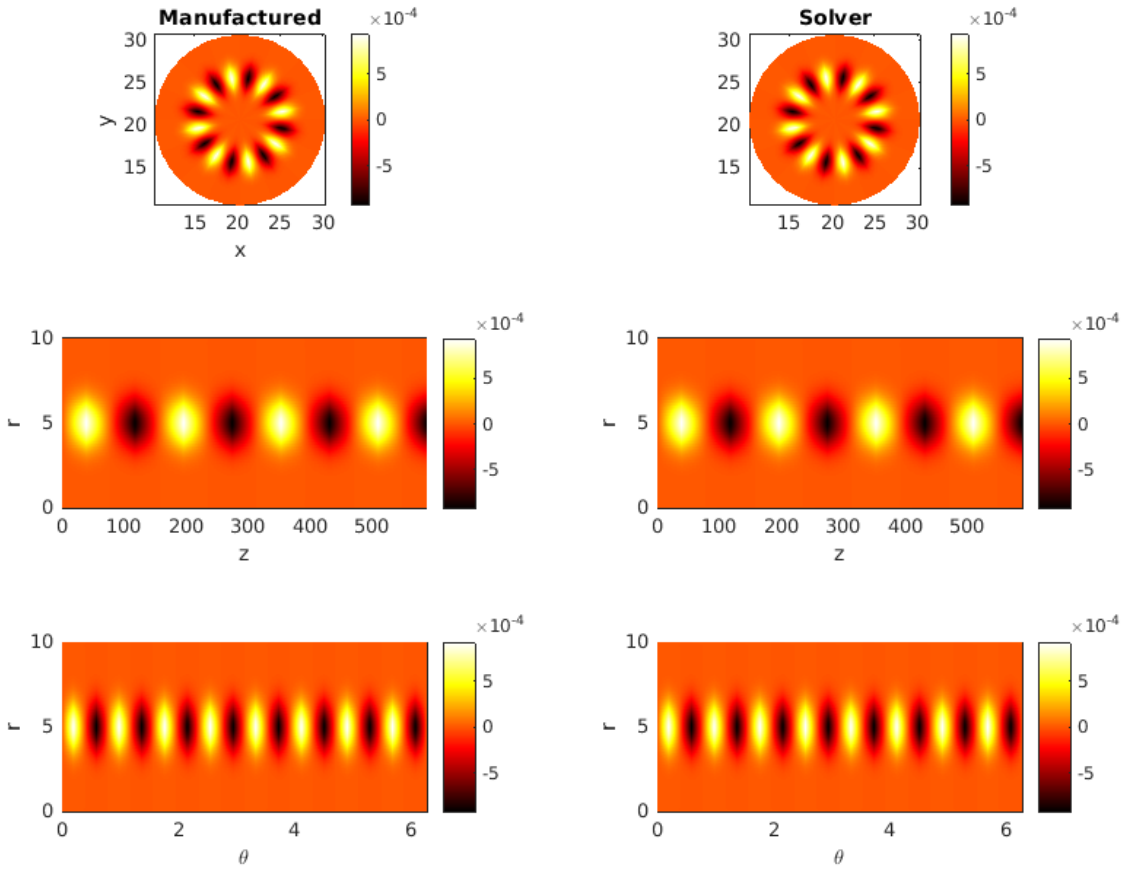


Figure 7.7: MMS test of the potential solver: On the left side the manufactured potential solution is shown for the poloidal, toroidal and r/θ-plane (where r is the radius). The same potential is shown on the right-hand side for the three planes as a result of the solver routine, where the input RHS was constructed from the manufactured solution. The two potentials coincide very well.

drive of the turbulence already ended and no significant turbulent behavior is present in the system anymore.
In figure 7.10 the L_2-norm of the electric potential $\|\phi\|_{L_2}$ as a function of time is plotted for varying a/L_T and $m_\mathrm{i}/m_\mathrm{e} = 40$. For runs with stronger temperature gradients, i. e. $a/L_\mathrm{T} = 4$, 8, 20 a clear linear phase appears initially. The larger a/L_T, the steeper is the linear phase that occurs. After the initial linear increase, $\|\phi\|_{L_2}$ reaches a maximum of and then reveals a mostly flat evolution. In this phase, no turbulence is driven in the system anymore. Again, the larger a/L_T is the faster the maximum is reached.
For simulations with $a/L_\mathrm{T} = 0$, 2 no linear phase can be achieved. This is an indicator

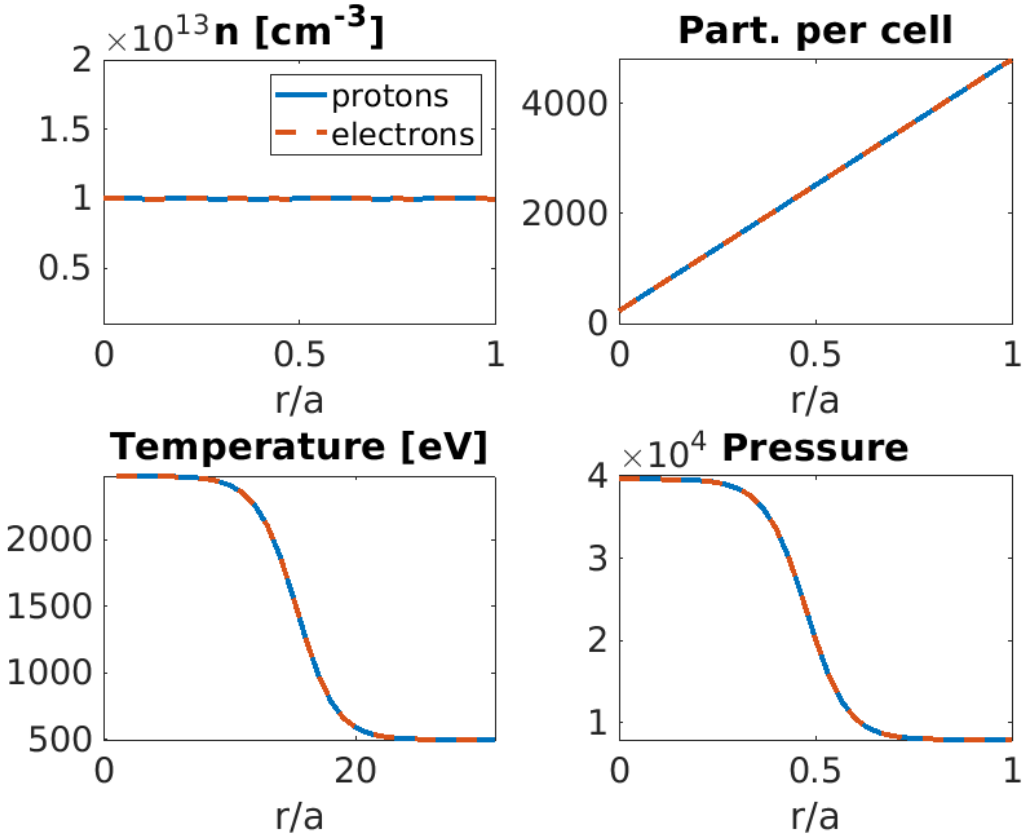


Figure 7.8: Initial profiles for density n, temperature and pressure and particles (or markers) per cell with respect to the normalized radius r/a. The characteristic length of the temperature gradient was chosen as $a/L_\mathrm{T} = 8$ and the mass ratio is $m_\mathrm{i}/m_\mathrm{e} = 40$.

that the temperature gradient is too small to drive turbulence in the system.
A similar result can be obtained for the field energy $\mathcal{E}_\mathrm{f}$ (see figure 7.11). The field energy can be calculated as shown in equation (2.78):

$$\mathcal{E}_\mathrm{f} \;=\; \sum_p \frac{1}{2} \int \mathrm{d}W \mathrm{d}V e_p J_{p,0} \phi f_p.$$

Similar to $\|\phi\|_{L_2}$, a turbulence-driven initial increase of $\mathcal{E}_\mathrm{f}$ can only be obtained for large enough a/L_T (4, 8, 20). Here holds, the larger a/L_T, the faster the maximum is reached and the larger is its maximum value. In addition, the maxima of $\mathcal{E}_\mathrm{f}$ match the maxima of $\|\phi\|_{L_2}$ shown in figure 7.10. In case of small a/L_T (0, 2), $\mathcal{E}_\mathrm{f}$ has a flat shape, which is another indicator for the lack of turbulence within these simulations.

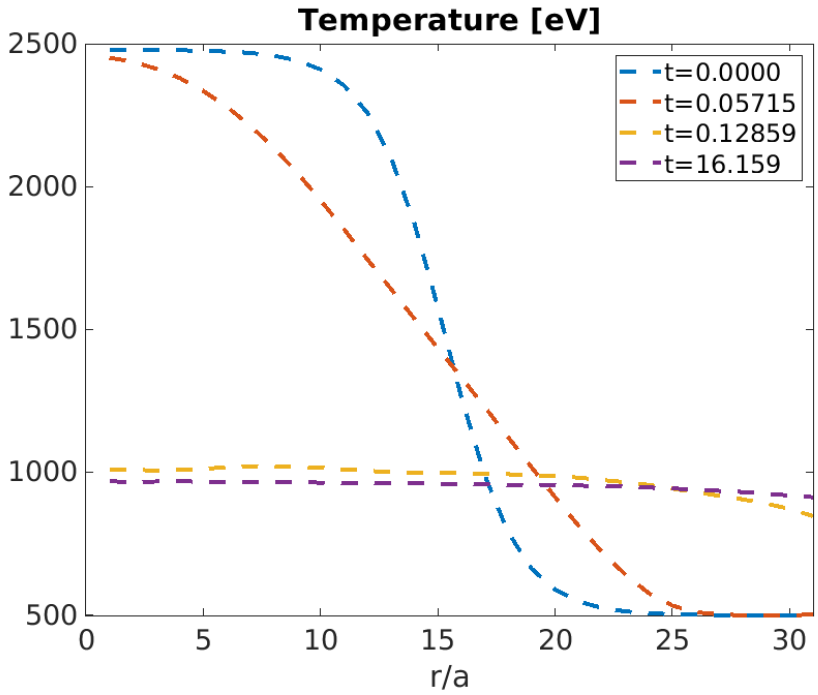


Figure 7.9: Profiles of the electron temperature T_e (in eV) as a function of r/a for different times. Note that the time t is shown in μs. The characteristic length of the temperature gradient was chosen as $a/L_T = 8$ and the mass ratio is $m_i/m_e = 40$.

To investigate the influence of the mass ratio on these simulations, in figures 7.12 and 7.13 we again plot $\|\phi\|_{L_2}$ and $\mathcal{E}_f$ similar to figures 7.10 and 7.11, but this time with $m_i/m_e = 100$. Mind the different length of the simulations. Looking at the evolution of the L_2-norm (figure 7.12) again a steep initial linear phase can be detected with similar maximum values. But this time even in the $a/L_T = 2$ case a faster and steeper initial increase is detectable compared to the $m_i/m_e = 40$ case. A similar picture is exhibited in the field energy $\mathcal{E}_f$ for $m_i/m_e = 100$ (figure 7.13). Also here, we see slight maximum appearing at $\sim 8\mu$s, where in the $m_i/m_e = 40$ case $\mathcal{E}_f$ remains flat. These findings reveal that in the $m_i/m_e = 100$ case also the $a/L_T = 2$ case shows a slight linear phase. Further approaching the real mass ratio of $m_i/m_e = 3670$ continues with this trend.
To better understand the turbulent behavior appearing in the initial linear phase, in figure 7.14 the potential in a fixed poloidal plane is shown for several time steps during the initial linear phase. Note that we selected $a/L_T = 20$ and a mass ratio of $m_i/m_e = 40$ as in the previous plots. The initial turbulent behavior evolves into a short mode structure and then gets quickly dissolved again. The mode structure

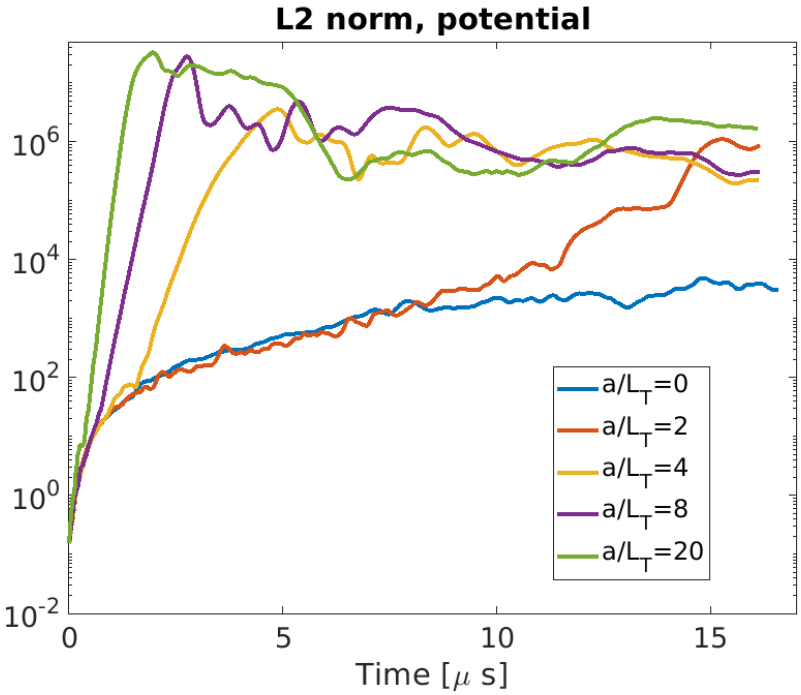


Figure 7.10: Evolution of the L_2-norm of the electric potential $\|\phi\|_{L_2}$ with time for varying a/L_T and $m_\mathrm{i}/m_\mathrm{e} = 40$.

is best visible in the snapshot $t = 0.9320\mu$s, but seems to be overlapped by another $n = 1$ mode. We want to remind the reader, that compared to a delta-f code this mode structure is less clearly visible, but also appears in our full-f model.
To achieve a more visible plot of the appearing mode structure in the linear phase, in figure 7.15 we again plot the potential in a fixed poloidal plane (similar to figure 7.14), but this time with a mass ratio of $m_\mathrm{i}/m_\mathrm{e} = 100$. In this figure, the appearing linear phase is more clearly visible before it dissolves again. However, since we changed the mass ratio, in this case the mode structure appears earlier and already is clearest in the snapshot $t = 0.4747\mu$s.
In a qualitative comparison of our PICLS results to simulations of the delta-f code ORB5 with a similar setup, good agreement was achieved in the initial linear phase. This can be seen in figure 7.16, where $\mathcal{E}_\mathrm{f}$ is compared for both codes. The same parameters were applied for both codes as described in section 7.2 with $a/L_\mathrm{T} = 8$ and $m_\mathrm{i}/m_\mathrm{e} = 40$. The PICLS result was already shown in figure 7.11, but in figure 7.16 it is slightly shifted on the time axis to better compare both linear phases. Apparently, in the delta-f code ORB5 the linear phase is reached at a later point in time. However, within the linear phase both curves show a very similar increase. After this phase, both curves

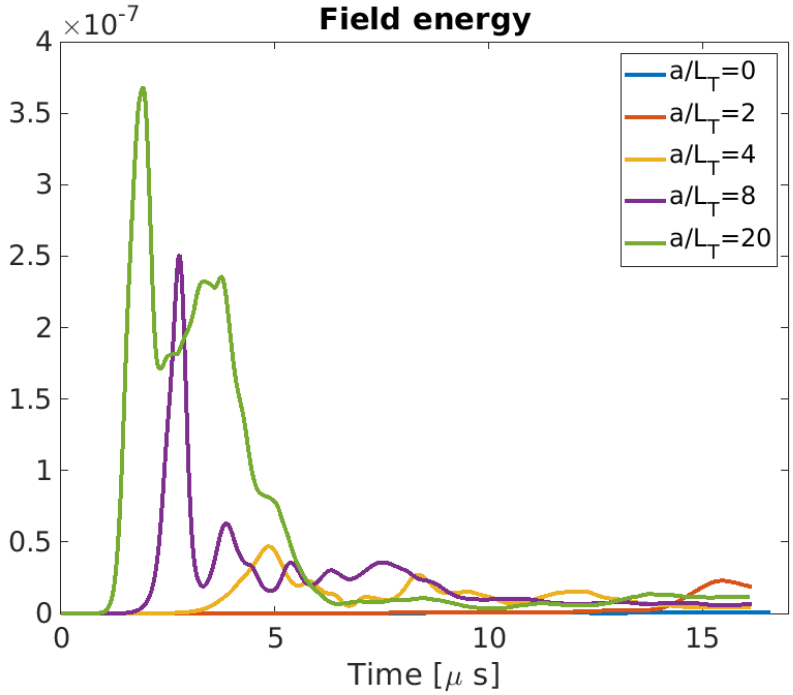


Figure 7.11: Field energy $\mathcal{E}_\mathrm{f}$ (see (2.78)) as a function of time for varying a/L_T and $m_\mathrm{i}/m_\mathrm{e} = 40$.

divert, which is a result of the different models implemented in both codes. Thus, the delta-f code ORB5 uses a Maxwellian background and only evolves the perturbations and the full-f code PICLS always evolves the full distribution function. With this, we could nicely show the good agreement of both codes in the linear phase. However, for a more sophisticated quantitative comparison additional work will have to be done in the future.

7.4 Conclusions

Within this chapter, we applied the 3D version of PICLS to a periodic screw pinch setup to study slab-ITG modes. This simple 3D setup was chosen to test the main 3D features of PICLS, such as the particle pusher and the 2D1D field solver. Thus, no sheath boundary conditions were applied in this test case. To validate the particle pusher, we performed a test on the particle trajectories. Here, the electric potential was initialized according to a predefined m=8, n=4 mode structure and maintained throughout the whole simulation. The effect of this “frozen” potential was clearly visible

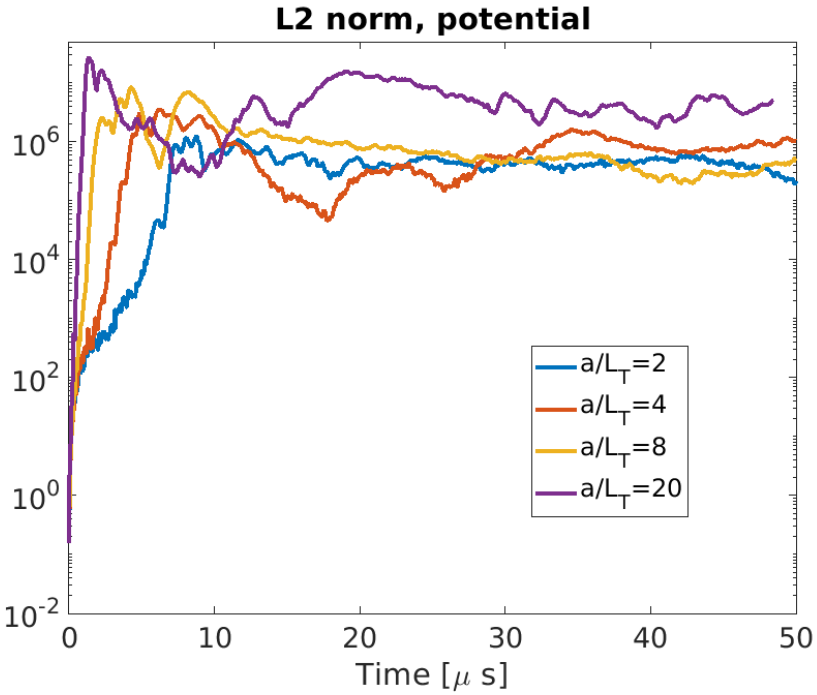


Figure 7.12: Evolution of the L_2-norm of the electric potential $\|\phi\|_{L_2}$ with time for varying a/L_T and $m_\mathrm{i}/m_\mathrm{e} = 100$.

in the trajectories of the particles. To test the 2D1D field solver used in this test case, we applied an MMS-like test of the potential solver. Therefore, a charge distribution was "back-engineered" from a chosen analytical potential. The resulting RHS was then solved via the 2D1D field solver and the resulting solution showed a good overlap with the initially chosen potential. After successfully testing these two routines, PICLS was applied to the slab-ITG instability test case. We performed several simulations with a q-profile of $q_\mathrm{s}(s) = 1 + 3s^2$, different characteristic lengths of the temperature gradient and varying mass ratios. The gradient length showed to be important, since significant turbulent behavior could only be achieved for runs with high enough a/L_T (4, 8, 20). The turbulence caused a steep linear phase in $\mathcal{E}_\mathrm{f}$ and $\|\phi\|_{L_2}$ at the beginning of the simulation. After the drive of the turbulence ended, also the profiles started to relax. The mode structure in the linear phase could also be shown in the potential of the poloidal plane. We finally showed a good agreement of the linear phase between the delta-f code PICLS and the full-f code ORB5. Obviously, several more aspects could be studied with this setup in the future. For example, convergence studies with varying number of particles per cell or a more quantitative comparison with a delta-f model (i. e. ORB5) could deliver interesting results.

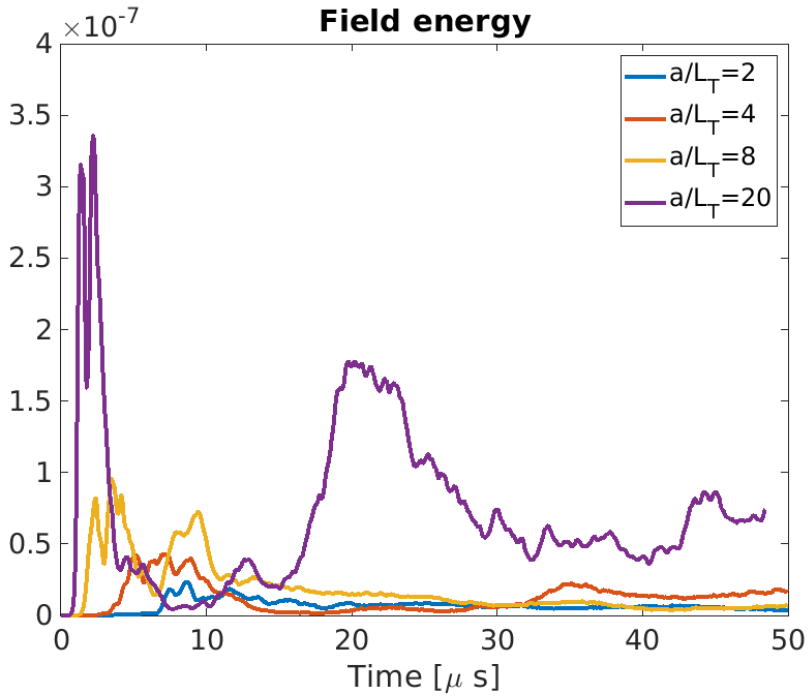


Figure 7.13: Field energy $\mathcal{E}_\mathrm{f}$ (see (2.78)) as a function of time for varying a/L_T and $m_\mathrm{i}/m_\mathrm{e} = 100$.

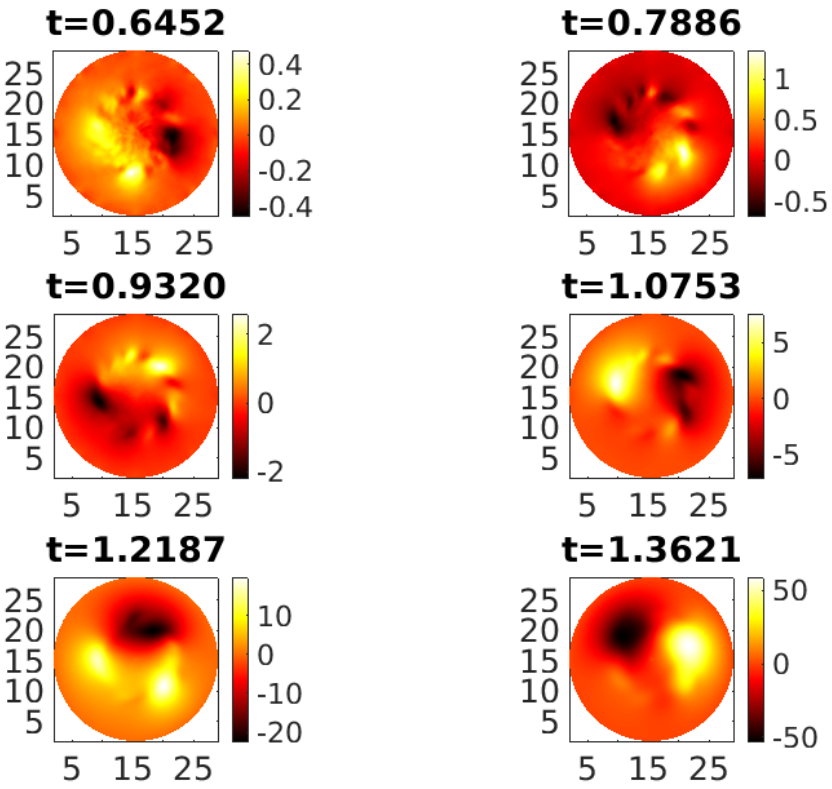


Figure 7.14: Plot of the potential in a fixed poloidal plane for several times (in μs) during the initial linear phase. $a/L_\mathrm{T} = 20$ and a mass ratio of $m_\mathrm{i}/m_\mathrm{e} = 40$ was chosen.

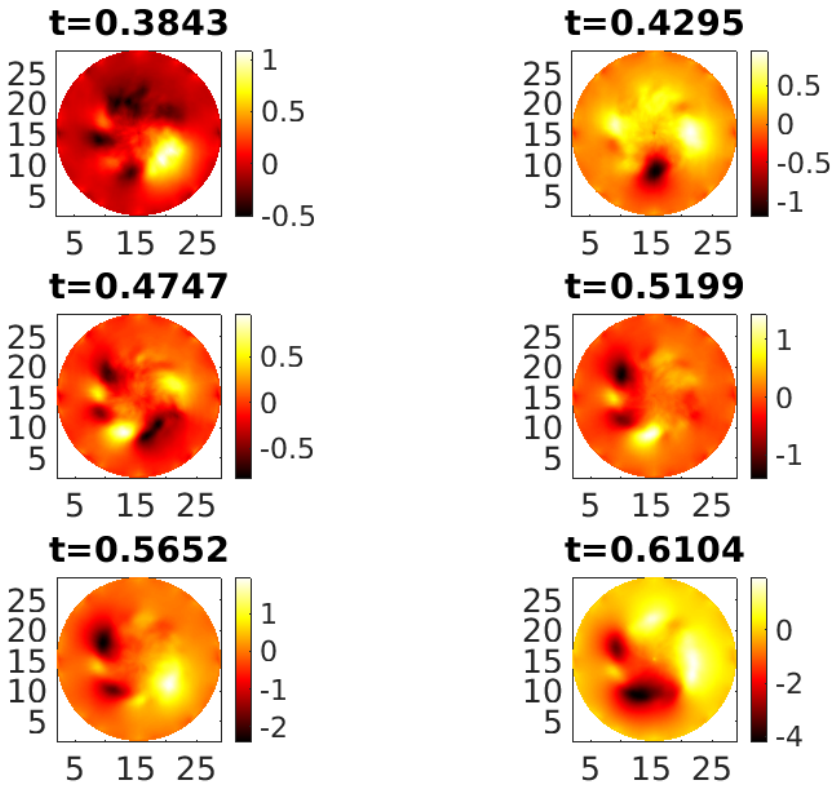


Figure 7.15: Plot of the potential in a fixed poloidal plane for several times (in μs) during the initial linear phase. $a/L_\mathrm{T} = 20$ and a mass ratio of $m_\mathrm{i}/m_\mathrm{e} = 100$ was chosen.

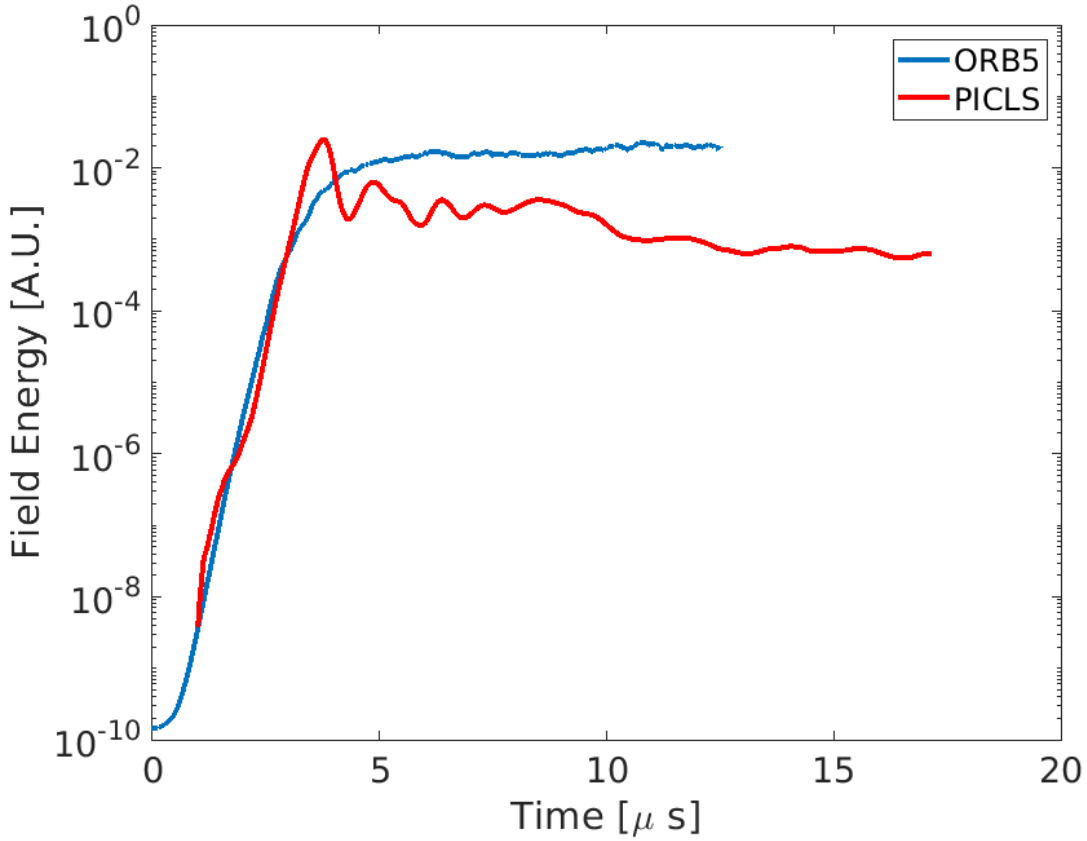


Figure 7.16: Field energy $\mathcal{E}_\mathrm{f}$ (see (2.78)) as a function of time for the full-f code PICLS and the delta-f code ORB5. For both codes we chose a similar setup with $a/L_\mathrm{T} = 8$ and $m_\mathrm{i}/m_\mathrm{e} = 40$. Note that the PICLS result was already shown in figure 7.11, but here it is slightly shifted in time to overlap the linear phases of both runs.

8 3D helical slab SOL simulations

In this chapter we make a first attempt to run 3D SOL simulations in a 3D slab geometry with helical field lines. In addition, this allows us to study several code implementations in the 3D SOL setup. Coming from the reduced 1D SOL setup thorough testing is required for the extended features to ensure validity of the code in 3D. Therefore, we chose a test case, which was already studied by Shi et al. [91] with a continuum gyrokinetic code and is useful to combine several code features (3D logical sheath boundaries, 3D particle pusher in slab, etc.). With the current implementations, however, PICLS would already be able to tackle more challenging problems. As mentioned in section 5.2.4, for example, PICLS is able to treat slab, cylindrical and simple circular tokamak geometries. Thus, in the future SOL simulations with different setups and geometries are planned.

8.1 Introduction

Within this chapter we want to make a first attempt to apply the 3D version of PICLS to helical open field lines to study cross-field transport in the SOL.
For magnetic fusion devices, such as tokamaks, in the SOL the propagation of blobs (also called plasma filaments) can convectively transport heat, particles, momentum and current across magnetic field lines and thus lead to a highly discontinuous cross-field transport ([26], [19], [23], [19]). Based on previous studies ([151], [152]) it is believed that in tokamak devices across the blob cross-section the curvature and $\nabla\mathbf{B}$ forces establish a charge-separated dipole potential. This mainly causes an outward radial propagation of the blob, due to convective $\mathbf{E} \times \mathbf{B}$ transport. For advanced fusion devices, such as ITER and beyond, the balance between parallel and cross-field transport in the SOL is decisive for how heat and particles are exhausted ([30]). Thus, for future performance predictions of fusion devices, modelling of these turbulent transport phenomena in the SOL region is important.
Blobs have in the past already been studied numerically via seeded-blob fluid simulations ([153], [154], [155], [156]). Also self-consistent formation of blobs was simulated with two-dimensional ([157], [158]) and three-dimensional turbulence models ([159], [160], [58], [161], [56], [162]). And very recently, Gkeyll was extended towards helical (or

curved) open magnetic field lines ([52, 91]). With this setup of Gkeyll blob formation was studied in the helical open field line plasma of a Texas Helimak like setup, which is a simple magnetized torus experiment (SMT) ([92], [163]). These gyrokinetic studies were in a later publication also compared to fluid code simulations of the GDB code ([164]).
However, the most advanced code up to now is probably the particle-in-cell code XGC1 ([159], [165]), which is the only one able to study turbulence in 3D diverted geometries and was able to study the SOL heat flux width with turbulent simulations ([39]).
Within this chapter, we apply our gyrokinetic full-f particle-in-cell code PICLS ([102], [127]) to helical open field lines in a 3D slab geometry, as previously done by Shi et al. ([91]). Therefore, we had to extend our previous 1D code to a 3D version which can cope with simple slab, cylindrical and circular tokamak geometries and includes magnetic curvature and $\nabla \mathbf{B}$ terms. With this setup also simple magnetized tori, such as TORPEX ([101]) and Helimak ([163]) could be studied. Similar to Shi et al. ([91]), we use the same tokamak SOL-like parameters and only consider open magnetic field lines, as in real tokamak devices. We are the first gyrokinetic particle-in-cell code studying this problem and thus use the previously generated continuum code results for comparison. However, the simulations presented here are still only a first attempt and further studies will have to be performed in the future.
In section 8.2 we describe the model we use for these simulations, including the Euler-Lagrange equations, the geometrical setup and the sheath boundary conditions. The simulation setup with relevant parameters, the initial conditions and the source term is shown in section 8.3. In section 8.4 first results for our helical slab simulations are presented for two different simulation time steps. And in section 8.5 we shortly summarize our findings.

8.2 Model

We use the low-frequency, electrostatic gyrokinetic model with kinetic electrons described in chapter 2. In the following, we only want to highlight the key concepts applied, for more details on the physical model and numerical setup used in PICLS, we want to refer to chapter 2 or our previous work ([102]).
For our model, we use the (electrostatic) Hamiltonian shown in eq. (2.4):

$$H_p = m_p \frac{{v_\parallel}^2}{2} + \mu B + e_p J_{p,0} \phi - \frac{m_p c^2}{2B^2} |\nabla_\perp \phi|^2.$$

With the velocity variables $v_\parallel$ (velocity parallel to magnetic field) and $\mu = m_p v_\perp^2 / (2B)$ (magnetic moment), the magnetic field strength B and the gyroaveraging operator

$(J_0\psi)(\mathbf{R},\mu) = \frac{1}{2\pi}\int_0^{2\pi}\psi(\mathbf{R}+\boldsymbol{\rho}(\theta))\,d\theta$. Here, $\mathbf{R}$ is the gyrocenter position and $\boldsymbol{\rho}$ the vector from the guiding center position to the particle position.
As described in section 2, with this Hamiltonian, the quasi-neutrality approximation and the linearized polarization approximation, our total gyrokinetic Lagrangian can be formulated ([50], [69], [49]) as (see eq. (2.23)):

$$L = \sum_p \int \mathrm{d}\Lambda \left(\left(\frac{e_p}{c}\mathbf{A} + m_p v_\parallel \mathbf{b} \right) \cdot \dot{\mathbf{R}} + \frac{m_p c}{e_p}\mu\dot{\theta} - H_p \right) f_p + \int \mathrm{d}V \frac{E^2 - B_\perp^2}{8\pi}.$$

The corresponding Euler-Lagrange equations were already shown in equations (2.31):

$$\begin{aligned} \dot{\mathbf{R}} &= v_\parallel \frac{\mathbf{B}^*}{B_\parallel^*} + \frac{c}{e_p B B_\parallel^*}\mathbf{B}\times\left[\mu\nabla B + e_p \nabla J_{p,0}\phi\right] \\ \dot{v_\parallel} &= -\frac{\mathbf{B}^*}{B_\parallel^*}\frac{1}{m_p}\cdot\left[\mu\nabla B + e_p\nabla J_{p,0}\phi\right], \end{aligned} \tag{8.1}$$

with $\mathbf{B}^* = \mathbf{B} + \frac{m_p c}{e_p} v_\parallel \nabla\times\mathbf{b}$ and $B_\parallel^* \equiv \mathbf{B}^*\cdot\mathbf{b}$.
From the total GK Lagrangian (8.1) the GK Poisson equation can be derived as shown previously (see eq. (2.40)):

$$-\sum_p \nabla_\perp \frac{n_{p,0} m_p c^2}{B^2}\nabla_\perp\phi = \sum_p \int \mathrm{d}W e_p J_{p,0} f. \tag{8.2}$$

With the definition $n_{p,0}$ for the gyrocenter density, which is similar to the density of the equilibrium Maxwellian f_M:

$$n_{p,0} = \int \mathrm{d}W f_{\mathrm{M},p} \tag{8.3}$$

Note that $J_{p,0} = 1$ for electrons, which means that electron FLR effects are neglected, due to their small Larmor radius. However, for the ions Larmor points are created and thus, FLR effects are taken into account. To solve for the potential, we apply the 2D1D Poisson solver for slab geometry as presented in 5.2.4.
To account for collisions, the conservative Lenard-Bernstein collision operator shown in section 4.3 is used ([127], [128]), which avoids the evaluation of Rosenbluth potentials. Due to their small collision frequency, ion-electron collisions are neglected. Electron-ion and self-species, however, are included in our model.
As in previous Gkyell simulations ([91], [92]), we also use a non-orthogonal field-line-following coordinate system. Here, x is the radial coordinate, z is the distance along field lines, and y measures the distance between field lines and is constant along a field

line. This geometry resembles a flux tube on the outboard side of a SMT that circulates around the torus several times and starts/ends at one of the two ends in z. To map these field-aligned coordinates (x, y, z) to cylindrical coordinates (R, ϕ, Z), we define the constants $R_c = R_0 + a_0$ and the field line pitch $\sin\theta = B_v/B$, with the device major radius R_0, the device minor radius a_0 and the vertical (poloidal) magnetic field B_v. Thus, we can write $R = x$ and $Z = z\sin\theta$. This simplified model has no magnetic shear and only vertical flux surfaces (thus ignores flux expansion). Since the vacuum toroidal field mainly dominates the field strength, we assume $\mathbf{B} = B_0(R_0/x)\mathbf{e_z}$.
We chose periodic boundary conditions (BC) in y for the particles and the fields. In x non-periodic Dirichlet BCs with $\phi = 0$ are applied and prevent particles to cross the domain boundary in x. Conducting sheath BCs are the method of choice in the z direction (see section 3.3 and [90], [52]). Here, depending on the sign of the sheath potential, which is determined by solving the Polarization equation (8.2), particles of one species are absorbed, while particles of the other species are partly reflected back into the plasma. In the simulations presented in this chapter we use the option to calculate the sheath potential for each particle entering the sheath individually according to its position in the x/y-plane at $z = 0$ (for more details see section 3.3). The particle dependent sheath potential ϕ_{sh} is calculated from the polarization equation at the position of the particle at the sheath entrances (at $z = 0$) and determines whether the particles are reflected back into the plasma by ϕ_{sh}, if their parallel velocity is not high enough. The conducting sheath BCs in general allow self-consistent currents in and out of the boundaries. This is in contrast to our previous 1D publications ([102],[127]), where we used logical or insulating sheath BCs ([114], [100], [115], [116]). As explained in section 3.2, for insulating sheath BCs the total parallel current to the wall is set to zero ($j_{\parallel} = 0$) and thus the wall can be regarded as insulating. We also tried insulating sheaths for the helical slab simulations presented here, but did not see a significant difference in the turbulent behavior so far.

8.3 Simulation setup

In the following we specify the key parameters used throughout the simulations, the implemented initial conditions and the source conditions during the simulation phase.

8.3.1 Relevant parameters

The parameters we use approximate a singly ionized H-mode deuterium plasma in the NSTX SOL ([166], [28]), and are similar to the ones used in Shi et al. ([91]): $T_{\text{e}} \sim 40$ eV, $T_{\text{i}} \sim 60$ eV, $n_{\text{i},0} = 7 \times 10^{18}$ cm^{-3}, $R_0 = 0.85$m, $a_0 = 0.5$m and $B_0 = 0.5$T. But we

want to stress that we do not attempt to capture H-mode physics, since our simulations only include the SOL with open field lines and not the pedestal in the edge region.
As we are approximating a flux tube at the outboard side of a torus, our dimensions are defined as: $L_x = 50\rho_{s0} \approx 14.6$ cm, $L_y = 100\rho_{s0} \approx 29.1$cm, $L_z = L_p/\sin\theta$. With the poloidal distance from midplane to end plates $L_p = 2.4$m, and the ion sound radius normalization $\rho_{s0} = c_{s0}/\Omega_i = 2.91$mm and the magnetic field line incidence angle θ. The magnetic field in our simulations approximates the tokamak SOL by using a helical field line geometry that mainly consists of a toroidal component and a smaller vertical component. Within the simulation part of this chapter (section 8.4), we are looking at a field pitch of $\sin\theta = B_v/B_z = (0.3)$ that results in $L_z = 8$m. This is in contrast to the previous work of Shi et al., who considered three different field pitches $\sin\theta = B_v/B_z = (0.2, 0.3, 0.6)$ that resulted in $L_z = 12, 8, 4$m. However, this will still be studied in the future. We are not considering the magnetic field line incidence angle in the sheath boundary conditions. However, the sheath model in PICLS would already be able to treat non-perpendicular incident field line angles for the particle sheath conditions.
The particles are generated via a Monte Carlo method according to their distribution function in the $(x, y, z, v_\parallel, \mu)$-space. The corresponding position-space extents are $x \in [-L_x/2 + R_0 + a_0, +L_x/2 + R_0 + a_0]$, $y \in [-L_y/2, +L_y/2]$ and $z \in [-L_z/2, +L_z/2]$. And the maxima for the particle velocity generation are $v_{\parallel s,\max} = 4v_{\mathrm{th},s} = 4\sqrt{T_s/m_s}$ and $\mu_{s,\max} = (3/4)m_s v^2_{\parallel s,\max}/(2B_0R_0/(R_0 + a_0))$. However, during the simulation run the velocities are not limited by these values and can also achieve higher values. This will also be discussed later. For the presented results, we use a uniform grid with $(N_x, N_y, N_z) = (32, 64, 20)$ cells and approximately 1000 particles per cell at the beginning of the simulation.

8.3.2 Initial conditions

The initial conditions for the particle distribution are estimated by the steady state solution of 1D fluid equations (for details see appendix of [91]). The reason for this is to decrease the computational time until a steady state is reached during the simulation. Within the initial source region ($|z| < L_{\mathrm{src}}/2 = L_z/4$) the initial fluid velocity, density

and temperature profiles are z-dependent and can be written as:

$$u_s(z) = \frac{\sqrt{3}}{2}\sqrt{\frac{2T_{s0}}{m_\mathrm{i}}}\left(\frac{1-\sqrt{1-z^2/(L_\mathrm{src}/2)^2}}{z/(L_\mathrm{src}/2)}\right), \tag{8.4}$$

$$n_s(z) = \frac{2\sqrt{5}L_\mathrm{src}S_\mathrm{n}}{3c_\mathrm{ss}}\left(\frac{1+\sqrt{1-z^2/(L_\mathrm{src}/2)^2}}{2}\right), \tag{8.5}$$

$$T_s(z) = \frac{3}{5}T_{s0}\left(\frac{3+5\sqrt{1-z^2/(L_\mathrm{src}/2)^2}}{4+4\sqrt{1-z^2}/(L_\mathrm{src}/2)^2}\right), \tag{8.6}$$

with $c_\mathrm{ss} = \sqrt{(10/3)T_{s0}/m_\mathrm{i}}$. For our simulations, the initial source particle rate we use $S_\mathrm{n} = 1.14 \times 10^{23}\mathrm{m}^{-3}\mathrm{s}^{-1}$ as shown in [52]. Outside this region ($|z| \geq L_\mathrm{src}/2$), n_s and T_s are constant and equal to the values calculated from (8.4), (8.5) and (8.6) at the edge of the source regions ($z = -L_\mathrm{src}/2$ and $z = L_\mathrm{src}/2$). For the initialization of the particle velocities, we use a Maxwellian distribution which is shifted towards the fluid velocity calculated in eq. (8.4).

8.3.3 Simulation phase

We define the source region for these simulations as the region with $x < x_s + 3\lambda_s$ and the SOL region as the region with $x \geq x_s + 3\lambda_s$, where $x_s = -0.05\mathrm{m} + R_0 + a_0$ and $\lambda_s = 0.5\mathrm{cm}$. The $x = x_s + 3\lambda_s$ plane can be referred as the LCFS, however, we want to stress that still no closed field line regions are included. Hence, also the temperature profiles of the source are distinguished between these two regions:

$$T_\mathrm{i,e}(x) = \begin{cases} 74\mathrm{eV} & x < x_s + 3\lambda_s \\ 33\mathrm{eV} & x \geq x_s + 3\lambda_s. \end{cases} \tag{8.7}$$

The velocity profiles are based on a Maxwellian distribution without a shift of the average velocity as in the initial conditions (see (8.4)). And for the plasma density source a x- and z-dependency is assumed for ions and electrons:

$$S(x,z) = \begin{cases} S_0\mathrm{max}\left[\exp\left(\frac{-(x-x_s)^2}{2\lambda_s^2}\right), 0.1\right] & |z| < L_z/4 \\ 0 & \text{else.} \end{cases} \tag{8.8}$$

From multiplying the total power into the scrape-off layer $P_{\mathrm{SOL}} = 5.4\mathrm{MW}$ by the fraction of the total machine volume that is covered by the simulation box, an expression for the total source power (electrons + ions) can be obtained as $P_{\mathrm{source}} = 0.27 L_z / L_{z0}$, with $L_{z0} = 4\mathrm{m}$. S_0 is chosen as to obtain this expression for P_{source}. To avoid regions with $n \ll n_0$ at large x, a floor of $0.1 S_0$ is set in $|z| < L_z/4$. This was done by Shi et al. ([91]) to prevent issues with distribution-function positivity which does not cause problems with our PIC algorithm, but we will keep it for comparability. Considering the distribution functions to be non-drifting Maxwellians and taking the specified temperature profile into account, leads to a source particle rate of $\approx 9.6 \times 10^{21}\mathrm{s}^{-1}$ for the specified $L_z = L_{z0}$ ($B_v/B_z = 0.6$) case.

8.4 Simulation results

In the following, we will present simulation results with two different time steps, a "large" one with $dt \approx 3.7\mathrm{ns}$ and "small" one with $dt \approx 0.1\mathrm{ns}$. The reason for choosing these time steps is that during our investigation of this test case we found out that we need a relatively small time step to ensure that the fast electron species is not accelerated excessively. Up to now we could not identify the exact reason, but we already have a good indication, which will be discussed later and needs to be analyzed further in the future. Thus, the following results are only a first attempt to study this test case with PICLS, but still no verification could be done, and further studies are required.
The time steps in general are chosen by considering how many time steps a thermal electron needs to cross one cell in the direction parallel to the magnetic field. This means that to calculate the time step dt, the cell size in z-direction is divided by the thermal velocity of a source electron $v_{\mathrm{th,e}} = \sqrt{T_{\mathrm{e,src}}/m_{\mathrm{e}}}$. This value is then divided by a factor which determines how many time steps a thermal electron needs to cross the cell in z-direction. In the end, we get $dt = \frac{\text{cell size in z}}{\#\text{time steps} \times v_{\mathrm{th,e}}}$. For $dt \approx 3.7\mathrm{ns}$ we chose a number of timesteps to cross the cell in z-direction of 60 and for $dt \approx 0.1\mathrm{ns}$ a factor of 2000. The second value seems to be excessive, but as mentioned before, was necessary to limit the acceleration of the electron species.
Due to the high resource requirements of the small time step, we will perform our simulations only up to $\sim$ 25-30μs and not up to $\sim 650\mu$s as done in [52]. An investigation of longer simulation times is planned, once the restriction on the very low time step is lifted. For both time steps, we investigated insulating as well as conducting sheath boundary conditions. But in the following we will only show the results for conducting sheath BCs, since the results did not drastically differ for the parameters and time interval we were looking at.

For the profile heatmap plots within this section we used 64 diagnostics cells in each spatial direction and applied a bicubic interpolation.

8.4.1 Simulations with "large" time step $dt \approx 3.7$ns

For the helical slab test case we first look at simulations with a time step of $dt \approx 3.7$ns, which is equivalent to a thermal electron crossing one cell in z-direction after 60 time steps. To start, we want to study the evolution of the electron density with time. Therefore, in figure 8.1 snapshots of the electron density (in $10^{18}\mathrm{m}^{-3}$) for $t \approx 17\mu$s, 21μs and 27μs, in the x/y-plane at $z = 0$m are plotted, similar to the data shown in [91]. The particle sources quickly trigger curvature-driven modes by steepening the

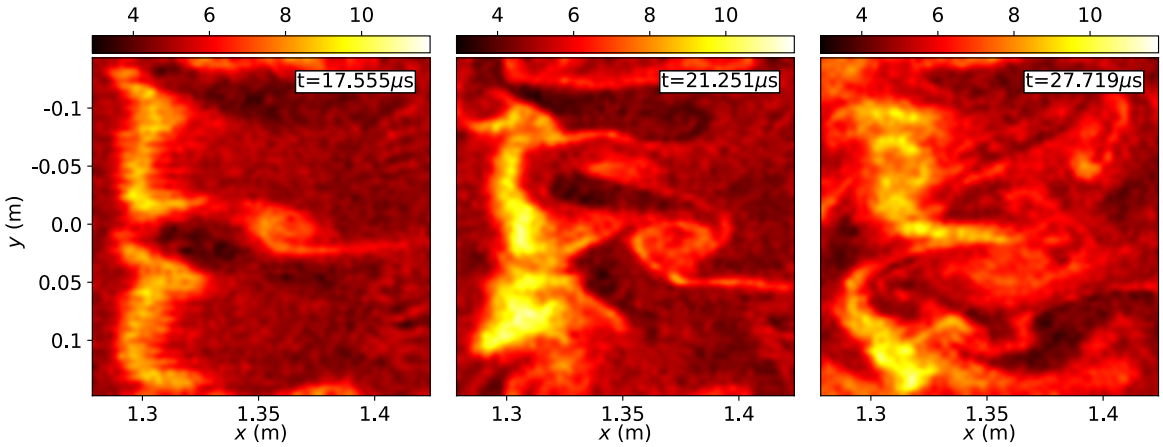


Figure 8.1: Snapshots of the electron density n_e (in $10^{18}\mathrm{m}^{-3}$) in the x/y-plane perpendicular to the B-field at $z = 0$m at $t \approx 17\mu$s, 21μs and 27μs for a time step of $dt \approx 3.7$ns. The color scale of the snapshots is equal to achieve comparability. Mind the different scales in x- and y-direction.

profiles. The turbulence is only driven by the sources, no additional perturbations are applied. Far from the original source region, radially elongated structures in x-direction are extending and broken up by the sheared flows in y-direction in the source region. This leads to blobs propagating in the SOL. Comparing our data with the results in [91], the repetitive uniform blobs cannot be reproduced, but more erratic structures are obtained. Our absolute density also is lower than in their publication, which could be an indication that the sources are implemented differently with our PIC algorithms. However, the absolute density values and the time scale for the growth of the mode is in a comparable range. Snapshots of previous times were not shown, since no relevant

turbulent behavior appears earlier.
For the snapshot $t \approx 27\mu$s in figure 8.2 we also want to show the electron temperature T_e (in eV) and the potential ϕ (in V). What is most striking, is the electron temperature

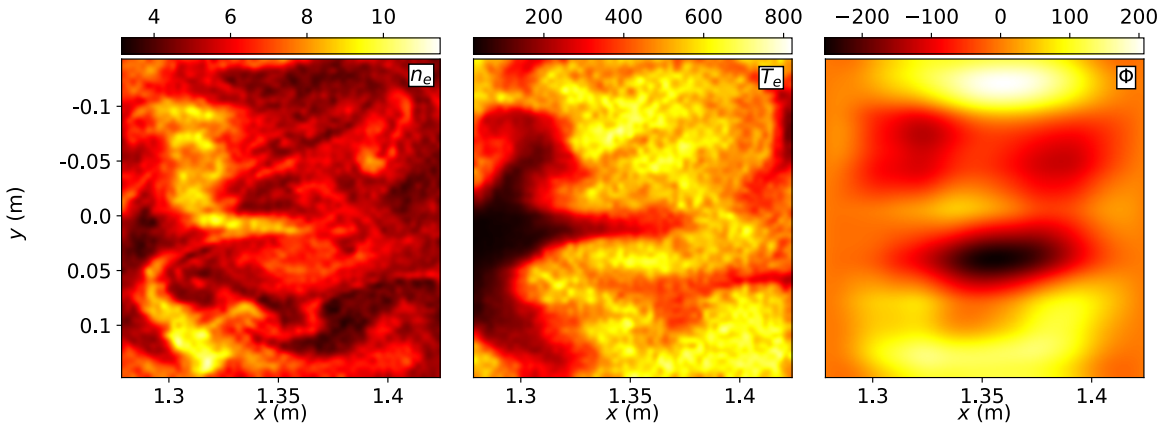


Figure 8.2: Snapshots of the electron density n_e (in $10^{18}\mathrm{m}^{-3}$), the electron temperature T_e (in eV) and the potential ϕ (in V) in the x/y-plane perpendicular to the B-field at $z = 0$m at $t \approx 27\mu$s for a time step of $dt \approx 3.7$ns. Mind the different scales in x- and y-direction.

(middle plot in figure 8.2), which with values of up to ~ 800eV is much higher than the temperature of the particles introduced by the sources $T_\mathrm{e,i}$. Hence, also the sheath potential which shows maximum values of $\phi \sim \pm 200$V is far from the expected positive values of $\sim 3T_\mathrm{e}/\mathrm{e}$. Comparing T_e and ϕ with the results in [91] delivers very different shapes and values (note that the snapshot in [91] shows $t \approx 650\mu$s).
The main reason for the unphysical behaviors of T_e and ϕ in our simulations comes from the electron velocities. To investigate this, in figure 8.3 we plotted the evolution of the average (left plots) and maximum (right plots) of the absolute value of electron (top plots) and ion (bottom plots) velocity in cm/s. For each subfigure, we show the evolution of these values for $v_\perp$ (in blue) and $v_\parallel$ (in red). In dashed lines the maximum parallel velocity for the particle sources $v_{\parallel s,\mathrm{max}} = 4v_{\mathrm{th},s} = 4\sqrt{T_s/m_s}$ is plotted (as defined in 8.3). At the beginning, the ion maximum velocities $\max(|v_{\perp/\parallel,\mathrm{i}}|)$ are quickly increasing, but remain at a velocity only slightly above $v_{\parallel\mathrm{i,max}}$. Only some ions seem to be slightly accelerated, since $\left\langle |v_{\perp/\parallel,\mathrm{i}}| \right\rangle$ is not increasing. Due to the conducting sheath boundary conditions, which absorb the fast ions that hit the domain wall, $\left\langle |v_{\perp/\parallel,\mathrm{i}}| \right\rangle$ is even decreasing. However, the more interesting species in figure 8.3 are the electrons. Similar to the ions, the maximum velocities $\max(|v_{\perp/\parallel,\mathrm{e}}|)$ are quickly increasing, but this time exceed $v_{\parallel\mathrm{e,max}}$ drastically by a factor of $\sim 6-7$. Apparently, this does not

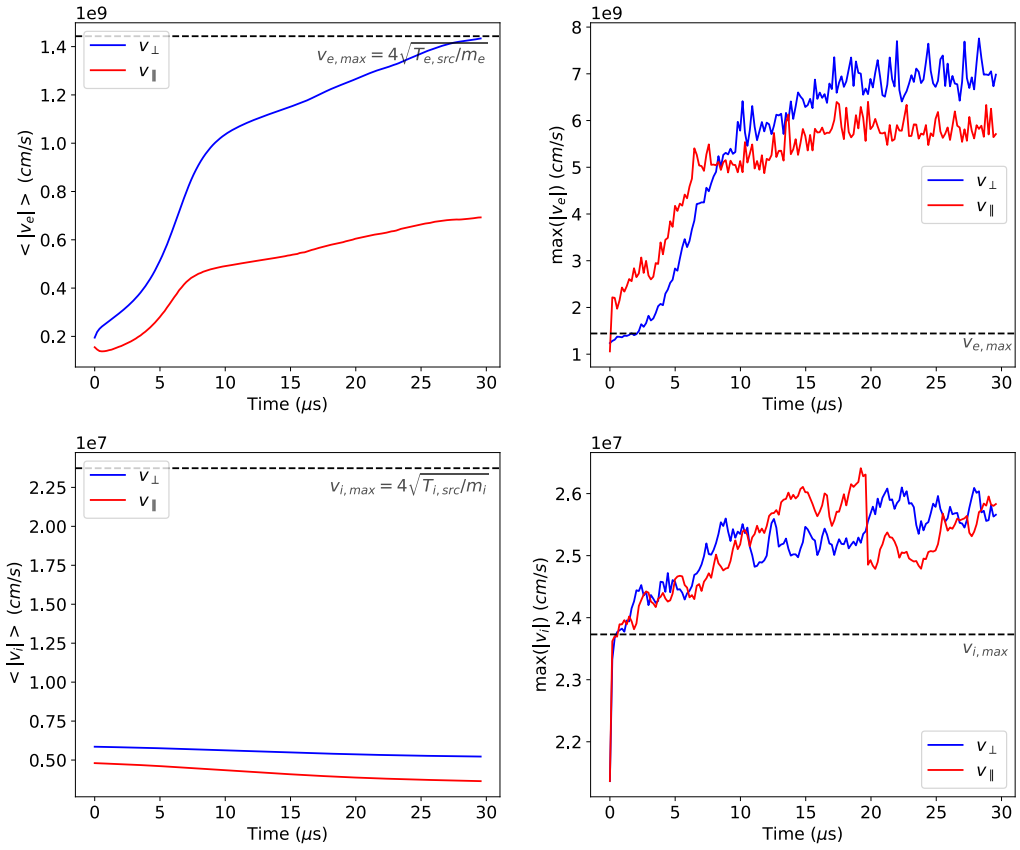


Figure 8.3: Evolution of the average (left plots) and maximum (right plots) of the absolute values of electron (top plots) and ion (bottom plots) velocity in cm/s for a time step of $dt \approx 3.7$ns. The blue (red) lines show the evolution for $v_\perp$ ($v_\parallel$). The dashed lines indicate the maximum velocity for the particle sources $v_{\parallel s,\max} = 4v_{\mathrm{th},s} = 4\sqrt{T_{s,\mathrm{src}}/m_s}$, with $T_{s,\mathrm{src}} = 74$eV.

happen for a few electrons only, as $\left\langle |v_{\perp/\parallel,\mathrm{e}}| \right\rangle$ is also increasing steadily. This is a clear indicator that the electrons within our simulations are accelerated too fast.
Further investigations showed that the acceleration of the electrons is caused by the second term in the $\dot{v}_\parallel$ Euler-Lagrange equation (see eq. (8.1)) where $\mathbf{B}^*$ is multiplied with $\nabla J_{p,0}\phi$. This is the only place in the equations, where the z-component of the electric field enters. Thus, we are currently investigating what causes the problems in the z-component of the electric field. Within our studies, we found out that decreasing the time step also decreases the influence of the faulty $\mathbf{B}^* \cdot \nabla J_{p,0}\phi$ term in the $\dot{v}_\parallel$ equation. Thus, in the next section we want to do the same study with an even lower time step.

8.4.2 Simulations with "small" time step $dt \approx 0.1$ns

In the previous section (8.4.1), we saw that for a time step of $dt \approx 3.7$ns an unphysical acceleration of the electron species appeared. Thus, we decrease the time step to $dt \approx 0.1$ns which is equivalent with a thermal electron crossing one cell in z-direction after 2000 time steps. This value seems to be excessively small, but helps us to understand the issue of the accelerating electrons described in section 8.4.1.
Therefore, in figure 8.4 the evolution of the average (left plots) and maximum (right plots) of the absolute value of electron (top plots) and ion (bottom plots) velocity in cm/s is plotted similar to 8.3 but with the smaller time step $dt \approx 0.1$ns. Roughly

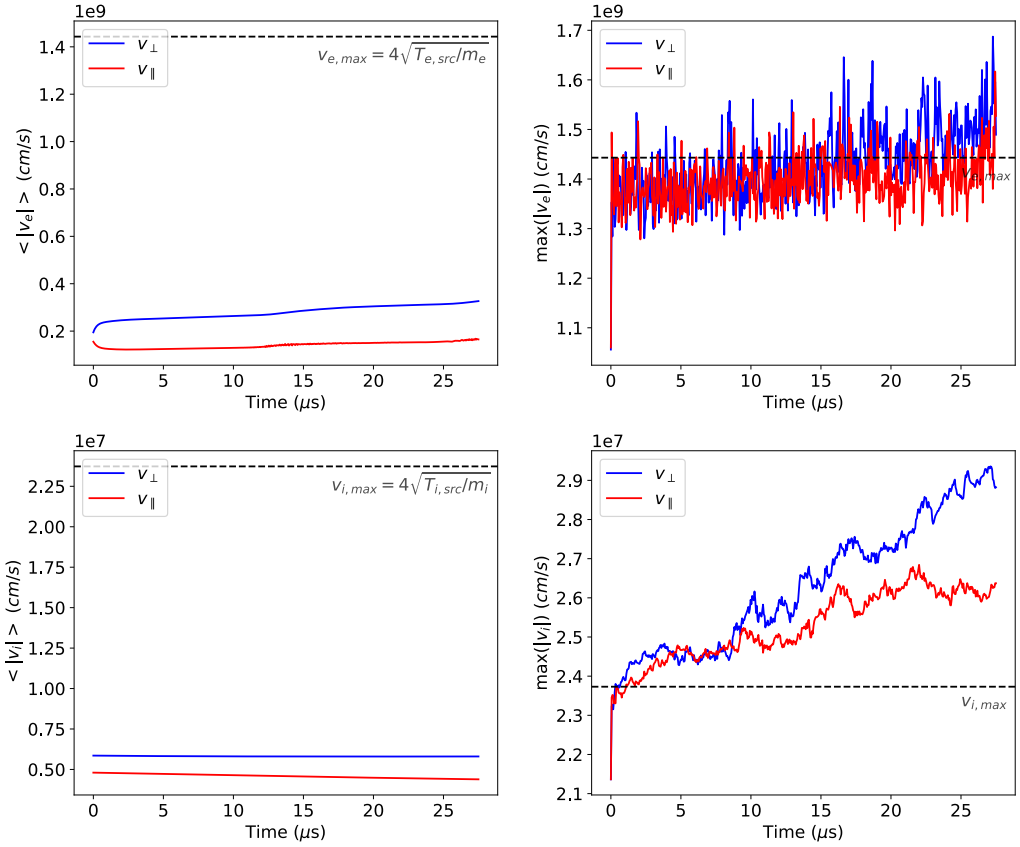


Figure 8.4: Evolution of the average (left plots) and maximum (right plots) of the absolute values of electron (top plots) and ion (bottom plots) velocity in cm/s for a time step of $dt \approx 0.1$ns. The blue (red) lines show the evolution for $v_\perp$ ($v_\parallel$). The dashed lines indicate the maximum velocity for the particle sources $v_{\parallel s,\max} = 4v_{\mathrm{th},s} = 4\sqrt{T_{s,\mathrm{src}}/m_s}$, with $T_{s,\mathrm{src}} = 74$eV.

the same simulation time of $\sim$ 25-30μs is used. For the ions, almost nothing changes

compared to the case with $dt \approx 3.7$ns. However, for the electrons the maximum velocity $\max(|v_{\perp/\parallel,\mathrm{e}}|)$ remains around $v_{\parallel\mathrm{e,max}}$ and does not exceed it by several factors anymore. This means that the electrons are not accelerated to these extreme velocities as in the $dt \approx 3.7$ns case. This can also be seen in $\left\langle |v_{\perp/\parallel,\mathrm{e}}| \right\rangle$, which is only very slightly increasing for the whole simulation time. This means that with decreasing the time step to a very small value, the unphysical acceleration of the electrons could be avoided. The reason therefore can be identified in figure 8.5, where we plotted a snapshot of the potential ϕ (in V) at $t \approx 27\mu$s (similar time as in 8.2). The potential is plotted

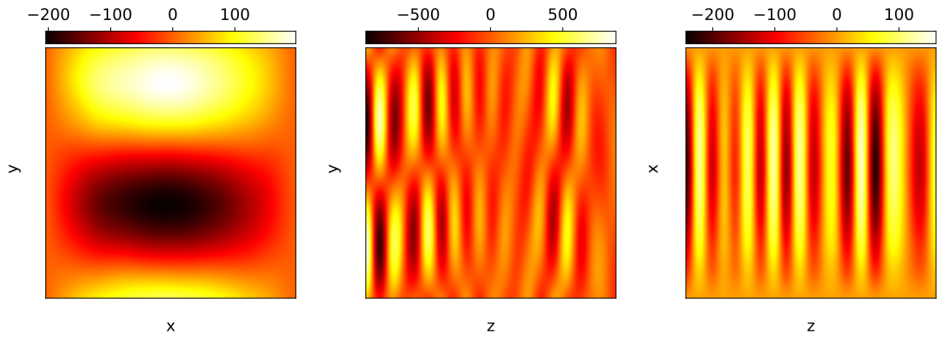


Figure 8.5: Snapshot of the potential ϕ (in V) at $t \approx 27\mu$s (similar time as in 8.2). The potential is plotted for the x/y-, z/y- and z/x-planes, where the extent of the spatial dimensions for each subplot is the full domain ($x \in [-L_x/2 + R_0 + a_0, +L_x/2 + R_0 + a_0]$, $y \in [-L_y/2, +L_y/2]$ and $z \in [-L_z/2, +L_z/2]$). All planes are positioned at the middle of the not shown direction (e.g., $z = 0$m for x/y-plane). Mind the different scales for the different directions.

for the x/y-, z/y- and z/x-planes. Here, for each subplot the extent of the spatial dimensions always is the full domain ($x \in [-L_x/2 + R_0 + a_0, +L_x/2 + R_0 + a_0]$, $y \in [-L_y/2, +L_y/2]$ and $z \in [-L_z/2, +L_z/2]$) and the planes are positioned at the middle of the not shown direction (e.g., $z = 0$m for x/y-plane). Thus, the scales are different for the different directions. The important thing in this plot is the strong periodic structure of the potential along the z-direction. We did this analysis for varying numbers of n_z, up to $n_z = 200$ to assure that this is not a resolution problem. However, the periodic structure in z remains for different n_z and always shows $\sim n_z/2$ maxima and minima in z-direction. Thus, we eliminated the possibility that the cause for this is an inherent physical mode. This periodic structure in z also appears in the simulations with larger dt. Compared to the simulations with $dt \approx 3.7$ns, we now also understand why the electrons are not accelerated here. The values of the periodic potential changes in z are similar for the $dt \approx 3.7$ns and the $dt \approx 0.1$ns case, but the time step itself is different. As a result, a much higher acceleration of the electrons is caused by the

$\mathbf{B}^* \cdot \nabla J_{p,0}\phi$ term in the $\dot{v}_\parallel$ equation for the larger time step.
The result of this non-physical field in z-direction is plotted in 8.6. Here, a snapshot at $t \approx 27\mu$s of the electron density n_e (in $10^{18}\mathrm{m}^{-3}$), the electron temperature T_e (in eV) and the potential ϕ (in V) is shown, similar to 8.2. As the electrons are less drastically

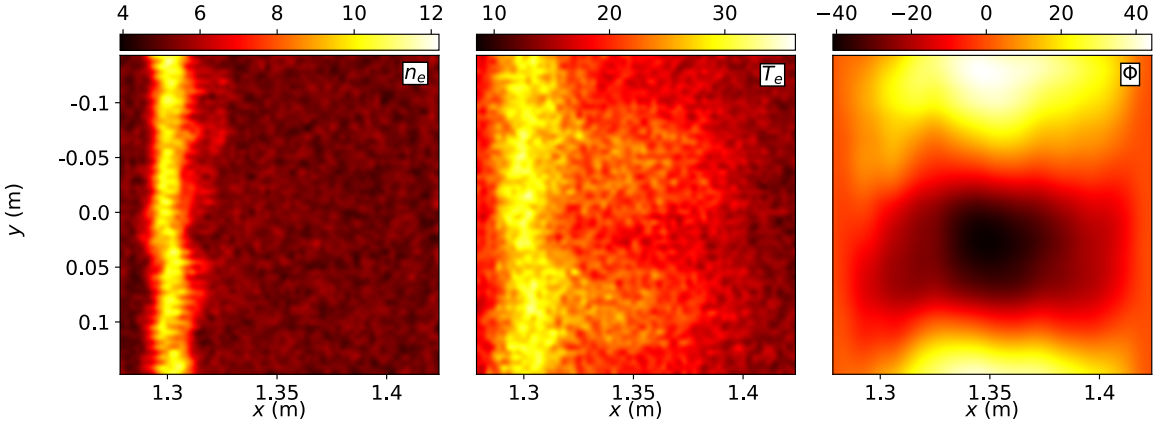


Figure 8.6: Snapshots of the electron density n_e (in $10^{18}\mathrm{m}^{-3}$), the electron temperature T_e (in eV) and the potential ϕ (in V) in the x/y-plane perpendicular to the B-field at $z = 0$m at $t \approx 27\mu$s for a time step of $dt \approx 0.1$ns. Mind the different scales in x- and y-direction.

accelerated in the $dt \approx 0.1$ns case, compared to $dt \approx 3.7$ns, this time the electron temperature T_e is in the right quantitative range. However, n_e and T_e do not show a significant turbulent behavior, but mostly maintain the structure of the source functions. This most likely comes from the incorrect potential in z-direction, which can lead to a trapping or acceleration of particles. Due to the periodic changes of ϕ in z, also the x/y-plane of ϕ shown in figure 8.6 cannot act as a reliable data point. The x/y-plane cuts differ significantly for different z-positions.
As a result, the simulations presented here can only provide a first glance at the studied helical slab test case and further analysis to resolve the remaining issues will be needed.

8.5 Conclusions

As a last test case for this manuscript, we selected a 3D helical slab setup and made a first attempt to study turbulence in helical open field lines. This test case was previously run by E. L. Shi et al. (see [52, 91]), whose results we used for comparison. As the studies presented here were only a first attempt, the test case was only run for up to

$\sim 30\mu$s, a domain size in z-direction of $L_z = 8$m and two different simulation time steps $dt \approx 3.7$ns and $dt \approx 0.1$ns.
Our current results showed that for the regarded time period we could achieve turbulent structures with the larger time step that showed similar temporal and spatial scales as presented in [91]. However, the qualitative behavior differed compared to [91]. A more thorough investigation and the application of a smaller time step showed that the potential in z-direction reveals a strong unexpected and almost certainly incorrect mode-like structure. The structure appears for different n_z and exhibits $\sim n_z/2$ maxima and minima in z-direction. This structure in the potential eventually leads to an unphysical acceleration and trapping of particles.
Hence, the current results can only be regarded as a first attempt and still show a probably unphysical structure in the z-component of the potential. Further studies will have to be done in the future to investigate the described issue and obtain more physical insights of this 3D helical slab test case.

9 Summary and outlook

The aim of this thesis was the development of the full-f gyrokinetic particle-in-cell code PICLS for simulations of the scrape-off layer or open field line regions. Gyrokinetic simulations are widely believed to be the way to go for quantitative turbulence studies in fusion relevant plasmas. Therefore, with the development of the gyrokinetic PICLS code we set the basis to contribute to open research questions in the plasma periphery. In the following, we want to summarize the main achievements of this work and give an outlook towards future development paths.

9.1 Summary

As explained, the key task of this thesis was to develop a full-f gyrokinetic particle-in-cell code for open field line simulations. Before starting a new code development for fusion plasma simulations, the physical and numerical models have to be determined. Hence, in chapter 2 we derived the gyrokinetic model from the total gyrokinetic Lagrangian. We used a Lie-transformed low-frequency particle Lagrangian and applied two common approximations — quasi-neutrality approximation and linearized polarization approximation — to achieve the implemented electrostatic total Lagrangian. With the help of the variational principle, we then derived the Euler-Lagrange equations and the linearized Polarization equation, which are implemented in the code. In addition, we also derived the simplified equations in slab geometry for our slab test cases and introduced the applied system of units. For completeness, the total conserved energy for the electrostatic system was formulated.
A key region of open field line simulations is the sheath region at the interface of plasma and vessel wall, a few Debye lengths in front of the wall material. This physical Debye sheath was described in chapter 3. Here, we also mentioned that due to its very short time and length scales this region cannot be resolved within our gyrokinetic simulations, but rather has to be modeled via so-called logical sheath algorithms. Two types of logical sheath algorithms were presented. The insulating and the conducting sheath, which mainly differ by allowing or prohibiting currents into the device wall. These models are rather simplified, but already provide a sufficient approximation of plasma sheaths for our simulations. In addition, we derived a method to treat particles at the

domain boundary with non-perpendicular incident B-field angle. However, a similar approach for the electric field for this case still has to be found. Further considerations of alternative sheath models, such as realistic neutral models were also shortly discussed in this chapter. We concluded this part by explaining the particle generation from pre-defined source functions via a Monte Carlo method.
In chapter 4 we introduced the collisional model that we implemented in PICLS. Therefore, we used the so-called Lenard-Bernstein (or Dougherty) collision operator, which reproduces the results of a Landau collision operator for infrequent collisions, but does not evaluate the Rosenbluth potentials. This collision operator analytically conserves the number of particles, parallel momentum and kinetic energy and relaxes towards a Maxwellian. The numerical implementation via the Langevin approach was also shown. Here, we demonstrated that with our current numerical implementation parallel momentum and kinetic energy are only conserved for an infinite number of particles, but the moment conservation is sufficient for our test cases. We also showed the convergence of parallel momentum and kinetic energy according to the $1/\sqrt{N_\mathrm{c}}$, with the number of markers per cell N_c.
The complete set of numerical methods was then derived in chapter 5. A key element of our particle-in-cell algorithm is its discretization via B-spline Finite Elements. We shortly explained the main features of this discretization method, before using it to derive the implemented 3D electric potential solver. The solver uses a Fast Fourier Transformation (FFT) in the periodic direction to gain computational efficiency. Therewith, PICLS in its current setup can perform simulations in 3D slab and cylindrical geometry. However, these setups require different types of solver implementations, which were also derived in chapter 5. Additional code specifics as the data structures, the parallelization schemes and a short performance analysis were also included at the end of this chapter.
After thoroughly deriving the code basis, in the following chapters 6, 7 and 8 we applied PICLS to its first open field line and scrape-off layer simulations. We started with a simplified 1D heat pulse test case in chapter 6, to test the basic validity of our derived model. Here, a central particle source mimics an ELM for a certain period of time and therewith creates parallel heat and particle transport onto the domain boundaries. For this test case we could show that the main part of the heat flux in the investigated time period was driven by the ions (66.7%) and that the contribution of the ion heat flux started at half of the ion transit time τ_i. We performed the same simulation with a non-collisional 1D1V model (only parallel velocity $v_\parallel$) and a collisional 1D2V model (also including μ) with a Lenard-Bernstein collision operator. Where the collisions resulted in an increase by 9% of the total heat flux loaded on the device wall and a deferred increase of the sheath potential.
After successfully studying plasma systems with one spatial domain in chapter 7 we

applied PICLS to its first 3D simuations, where we studied ion temperature gradient (ITG) modes in a periodic screw pinch setup without sheath boundary conditions. Since going from 1D to 3D included significant changes in the models and algorithms applied, we selected this test case to verify the validity of the two key features, particle pusher and field solver. For the particle pusher we fixed the electric field acting on the particles and studied the particle trajectories and for the field solver, we applied a Method of Manufactured Solutions (MMS) based approach. Both verification tests were passed and thus PICLS could be applied to study slab-ITG instabilities in the cylindrical setup. Here, we could show turbulent behavior with a short initial linear phase, which could be seen in the evolution of the field energy and the L_2-norm of the potential $\|\phi\|_{L_2}$. After this linear phase, the profiles relax and the drive for the turbulence ends. In addition, we could compare the linear phase of the field energy of a simulation with our full-f code PICLS with the results of a comparable simulation with the delta-f code ORB5 and achieved good qualitative agreement of both codes.
At the end of this thesis we did a first attempt to study turbulence in a 3D helical slab setup with open field lines in chapter 8. This test case was based on a previously studied application and served to test most of the open field line features implemented in PICLS and to achieve first curvature-driven turbulence and blob propagation results. We studied two different simulation time steps and compared the results with previous continuum code simulations. Blob-like structures for the "large" time step case were achieved. However, a deeper analysis showed that an unphysical mode structure in z-direction appears, which leads to an acceleration and trapping of electrons and thus significantly affects the simulation results. We are still working on identifying the root cause of this issue and expect it in the slab-specific field solver. Thus, the results shown in this thesis can only be seen as a first glimpse on this 3D helical slab test case, but further studies and verification test will be done in the future.

9.2 Outlook

With the current development state of the full-f gyrokinetic particle-in-cell code PICLS already several simplified test cases and physical questions can be addressed (see chapters 6, 7 and 8). Based on this solid ground, in the future there are in general two paths to take: exploit PICLS's current capabilities and further extend the code towards more realistic and challenging simulations. With the current setup of PICLS, the plasma behaviour of several simple machines, such as LAPD [147], TORPEX [101] or HELIMAK [163] could already be studied. Studies on these machines were already performed by different gyrokinetic and fluid codes. Thus, executing simulations on similar or new test cases could be a valuable contribution to cross-check existing results

and achieve new insights. Nevertheless, the even more valuable (but also significantly more challenging) task is to push PICLS further towards realistic scrape-off layer simulations in tokamak-like machines.
PICLS already includes several key features to address open field line simulations, such as sheath models, full-f treatment, etc. However, to simulate a real tokamak scrape-off layer major development efforts are still required. These can be grouped in physical and HPC requirement tasks.

In terms of physical adaptations an initial focus lies on lifting the current delta-f limitations to further push the code towards a setup for gyrokinetic edge simulations. The code already mainly includes a full-f model, however, in its current version the Polarization equation is linearized. Implementing a **non-linear Polarization equation** therefore would be a logical next step. The work done on an iterative method by Alexey Mishchenko (see [108]) could help as a guideline here. Another obvious step to take is the extension of the implemented Lenard-Bernstein collision operator to a more advanced **full-f collision operator**. A Fokker-Planck collision operator, as shown in [125], could be good choice here.
To progress towards realistic tokamak simulations still some larger development efforts to improve the applied physics will be needed. To achieve results that are closer to the physical reality, the current electrostatic model will have to be extended to an **electromagnetic** one by implementing an Ampère's law (see [167]). Furtermore, **coupling to impurity and neutral physics** will be required for realistic edge simulations. In a first step, PICLS could then be **coupled to an existing core code (such as ORB5) to perform limiter simulations**. Once reaching more complicated geometries an **extension of the field/particle boundary conditions to arbitrary geometries** will be necessary. Also a **revision of the GK equations** for specified edge parameters is thinkable. In a second and significantly more advanced step, PICLS could be applied to **cross the separatrix** within a real tokamak geometry. Potential options therefore would be to couple with core codes, similar to the mentioned approach for limiters, or to implement an FCI (flux-coordinate indipendent; see [56]) approach together with unstructured grids (see [168, 169]) for the field calculation into PICLS. Both approaches will probably mean significant modifications and still have to be refined further in the future in terms of effort and feasibility.
Another major field to investigate is the numerical transition from a gyrokinetic core model towards an edge model. An important ingredient for this is the implementation of a **noise control mechanism** as presented in [170]. With this, PICLS could be used to execute numerical studies on the **transition from a delta-f to a full-f model**, which is of relevance for the coupling of core and edge simulations. In addition, also the **transition from open- to closed field lines**, which is decisive for crossing the

separatrix, could be studied in simplified geometries.

The current version of PICLS is based on a hybrid OpenMP and MPI parallelization. Apart from general code optimization, additional effort will be needed to **port the code to GPUs** to fully exploit the capabilities of new computing systems. Potential options here are OpenACC and OpenMP-offload, which have to be further evaluated. Another relevant task is to implement **IMAS interfacing** for a standardized data output.

There is still a lot of work ahead to extend PICLS towards realistic turbulence simulations of the tokamak scrape-off layer region. But with its current setup we constructed a solid basis for future development. The PICLS project remains to be a collaborative effort of IPP Garching and SPC Lausanne and concrete plans for the development of several of the discussed problems are already on the way. Furthermore, a potential collaboration with IPP Greifswald on the development of a non-linear Polarization equation is planned. The progress of the PICLS project therewith seems to be secured and further physical results and publications are expected in the near future.

Bibliography

[1] U. DESA. *United Nations Department of Economic and Social Affairs/Population Division (2015): World Population Prospects: The 2015 Revision, Key Findings and Advance Tables. Working Paper No. ESA/P/WP.241*. 2015.

[2] U. S. E. I. A. EIA. *International Energy Outlook 2017*. 2017.

[3] I. E. A. IEA. *Key world energy statistics 2016*. 2016.

[4] T. Stocker. *Climate change 2013: the physical science basis: Working Group I contribution to the Fifth assessment report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2014.

[5] D. Kramer. *ITER costs are pinned down, with caveats*. Physics Today, 2016.

[6] E. National Academies of Sciences et al. *Final Report of the Committee on a Strategic Plan for U.S. Burning Plasma Research*. Washington, DC: The National Academies Press. doi: https://doi.org/10.17226/25331, 2018.

[7] U. Stroth. *Plasmaphysik: Phänomene, Grundlagen, Anwendungen*. Springer-Verlag, 2011.

[8] F. Chen. *Introduction to plasma physics and controlled fusion*. Plenum Press, 1974.

[9] J. A. Bittencourt. *Fundamentals of plasma physics*. Springer Science & Business Media, 2013.

[10] J. Nührenberg et al. "Overview on Wendelstein 7-X theory", *Fusion Technology* 27, 3T (1995), pp. 71–78.

[11] A. Dinklage et al. "Magnetic configuration effects on the Wendelstein 7-X stellarator", *Nature Physics* 14, 8 (2018), pp. 855–860.

[12] J. Wesson et al. *Tokamaks*. Vol. 149. Oxford University Press, 2011.

[13] P. Helander et al. *Collisional Transport in Magnetized Plasmas*. Cambridge Monographs on Plasma Physics. Cambridge University Press, 2002.

[14] A. H. Boozer. "Theory of tokamak disruptions", *Physics of Plasmas* 19, (2012), p. 058101.

[15] P. C. Stangeby et al. "Plasma boundary phenomena in tokamaks", *Nuclear Fusion* 30, (1990), p. 1225.

[16] F. Wagner et al. "Regime of Improved Confinement and High Beta in Neutral-Beam-Heated Divertor Discharges of the ASDEX Tokamak", *Phys. Rev. Lett.* 49, (1982), pp. 1408–1412.

[17] F. Wagner. "A quarter-century of H-mode studies", *Plasma Physics* 49, (2007), B1–B33.

[18] I. Kaw P. & Bandyopadhyay. *I. 2012 The case for fusion. In Fusion Physics (ed. M. Kikuchi & K. Lackner & M. Q. Tran), pp. 1-58.* International Atomic Energy Agency, 2012.

[19] S. J. Zweben et al. "Edge turbulence measurements in toroidal fusion devices", *PLASMA PHYSICS AND CONTROLLED FUSION* 49, 7 (2007). IEA Large Tokamak IA Workshop on Edge Transport in Fusion Plasmas, Cracow, POLAND, SEP 11-13, 2006, S1–S23. ISSN: 0741-3335.

[20] P. Ricci. "Simulation of the scrape-off layer region of tokamak devices", *Journal of Plasma Physics* 81, 2 (2015), p. 435810202.

[21] P. C. Stangeby. *The plasma boundary of magnetic fusion devices.* CRC Press, 2000.

[22] A. Mosetto et al. "Turbulent regimes in the tokamak scrape-off layer", *Physics of Plasmas* 20, 9 (2013), p. 092308.

[23] S. Zweben et al. "Structure of edge-plasma turbulence in the Caltech tokamak", *Nuclear fusion* 25, 2 (1985), p. 171.

[24] S. Zweben et al. "High-speed imaging of edge turbulence in NSTX", *Nuclear Fusion* 44, 1 (2003), p. 134.

[25] J. Boedo. "Edge turbulence and SOL transport in tokamaks", *Journal of Nuclear Materials* 390, (2009), pp. 29–37.

[26] J. Boedo et al. "Edge transport studies in the edge and scrape-off layer of the National Spherical Torus Experiment with Langmuir probes", *Physics of Plasmas* 21, 4 (2014), p. 042309.

[27] J. Boedo et al. "Intermittent convection in the boundary of DIII-D", *Journal of nuclear materials* 313, (2003), pp. 813–819.

[28] S. Zweben et al. "Blob structure and motion in the edge and SOL of NSTX", *Plasma Physics and Controlled Fusion* 58, 4 (2016), p. 044007.

[29] S. Zweben et al. "Blob wakes in NSTX", *Physics of Plasmas* 26, 7 (2019), p. 072502.

[30] A. Loarte et al. "Power and particle control", *Nuclear Fusion* 47, 6 (2007), S203.

[31] R. Goldston. "Theoretical aspects and practical implications of the heuristic drift SOL model", *Journal of Nuclear Materials* 463, (2015), pp. 397–400.

[32] A. Leonard. "Plasma detachment in divertor tokamaks", *Plasma Physics and Controlled Fusion* 60, 4 (2018), p. 044001.

[33] E. Sytova et al. "Comparing N versus Ne as divertor radiators: SOLPS-ITER simulations of impurity seeding in ASDEX Upgrade-like geometry," *23rd International Conference on Plasma Surface Interactions in Controlled Fusion Devices (PSI 23)*. 2017.

[34] M. Kotschenreuther et al. "Taming the heat flux problem: advanced divertors towards fusion power", *Journal of Fusion Energy* 35, 1 (2016), pp. 27–30.

[35] J.-W. Ahn et al. "Modification of divertor heat and particle flux profiles with applied 3D fields in NSTX H-mode plasmas", *Nuclear Fusion* 50, 4 (2010), p. 045010.

[36] T. Eich et al. "Scaling of the tokamak near the scrape-off layer H-mode power width and implications for ITER", *Nuclear fusion* 53, 9 (2013), p. 093031.

[37] M. Makowski et al. "Analysis of a multi-machine database on divertor heat fluxes", *Physics of Plasmas* 19, 5 (2012), p. 056122.

[38] R. Goldston. "Heuristic drift-based model of the power scrape-off width in low-gas-puff H-mode tokamaks", *Nuclear Fusion* 52, 1 (2011), p. 013009.

[39] C. S. Chang et al. "Gyrokinetic projection of the divertor heat-flux width from present tokamaks to ITER", *Nuclear Fusion* 57, 11 (2017), p. 116023.

[40] R. H. Cohen et al. "Scrape-Off Layer Turbulence Theory", *Contributions to Plasma Physics* 34, 2-3 (1994), pp. 232–246.

[41] D. Tskhakaya et al. "Kinetic simulations of the parallel transport in the JET scrape-off layer", *Journal of Nuclear Materials* 390-391, (2009).

[42] G. Manfredi et al. "Parallel transport in a tokamak scrape-off layer", *38th EPS Conference on Plasma Physics 2011, EPS 2011 - Europhysics Conference Abstracts* 35, (2011).

[43] R. Schneider et al. "B2-EIRENE simulation of ASDEX and ASDEX-Upgrade scrape-off layer plasmas", *Journal of nuclear materials* 196, (1992), pp. 810–815.

[44] R. Schneider et al. "Plasma Edge Physics with B2-Eirene", *Contributions to Plasma Physics* 46, 1-2 (2006), pp. 3–191.

[45] V. Rozhansky et al. "New B2SOLPS5. 2 transport code for H-mode regimes in tokamaks", *Nuclear Fusion* 49, 2 (2009), p. 025007.

[46] R. Cohen et al. "Progress in kinetic simulation of edge plasmas", *Contributions to Plasma Physics* 48, 1-3 (2008), pp. 212–223.

[47] B. Scott et al. "Energetic consistency and momentum conservation in the gyrokinetic description of tokamak plasmas", *Physics of Plasmas* 17, 11 (2010), p. 112302.

[48] R. Schneider et al. "Challenges in plasma edge fluid modelling", *Plasma Physics and Controlled Fusion* 49, 7 (2007), S87.

[49] N. Tronko et al. "Second-order nonlinear gyrokinetic theory: from the particle to the gyrocentre", *Journal of Plasma Physics* 84, 3 (2018).

[50] H. Sugama. "Gyrokinetic field theory", *Physics of Plasmas* 7, (2000), p. 466.

[51] A. Brizard. "Variational principle for nonlinear gyrokinetic Vlasov-Maxwell equations", *Physics of Plasmas* 7, 12 (2000), pp. 4816–4822.

[52] E. Shi. "Gyrokinetic continuum simulation of turbulence in open-field-line plasmas." Ph.D. dissertation. Princeton University, 2017.

[53] A. Ross. "Extension of GRILLIX: Towards a global fluid turbulence code for realistic magnetic geometries." Dissertation. München: Technische Universität München, 2018.

[54] D. Reiter et al. "The EIRENE and B2-EIRENE codes", *Fusion Science and Technology* 47, 2 (2005), pp. 172–186.

[55] B. D. Scott. "Computation of electromagnetic turbulence and anomalous transport mechanisms in tokamak plasmas", *Plasma physics and controlled fusion* 45, 12A (2003), A385.

[56] A. K. Stegmeir. "GRILLIX: A 3D turbulence code for magnetic fusion devices based on a field line map." PhD thesis. 2015.

[57] A. Stegmeir et al. "The field line map approach for simulations of magnetically confined plasmas", *Computer Physics Communications* 198, (2016), pp. 139–153.

[58] A. Stegmeir et al. "GRILLIX: a 3D turbulence code based on the flux-coordinate independent approach", *Plasma Physics and Controlled Fusion* 60, 3 (2018), p. 035005.

[59] T. Body et al. "Treatment of advanced divertor configurations in the flux-coordinate independent turbulence code GRILLIX", *Contributions to Plasma Physics* 60, 5-6 (2020), e201900139.

[60] W. Zholobenko et al. "Thermal dynamics in the flux-coordinate independent turbulence code GRILLIX", *Contributions to Plasma Physics* 60, 5-6 (2020), e201900131.

[61] A. Stegmeir et al. "Advances in the flux-coordinate independent approach", *Computer Physics Communications* 213, (2017), pp. 111–121.

[62] P. Snyder et al. "Landau fluid models of collisionless magnetohydrodynamics", *Physics of Plasmas* 4, 11 (1997), pp. 3974–3985.

[63] B. D. Scott. "Tokamak edge turbulence: background theory and computation", *Plasma Physics and Controlled Fusion* 49, 7 (2007), S25.

[64] K. Lackner et al. "Equilibrium and Macroscopic Stability of Tokamaks," *Fusion Physics*. 2012.

[65] X. Xu et al. "Gyro-fluid and two-fluid theory and simulations of edge-localized-modes", *Physics of Plasmas* 20, 5 (2013), p. 056113.

[66] B. D. Scott et al. "Nonlinear dynamics in the tokamak edge", *Contributions to Plasma Physics* 50, 3-5 (2010), pp. 228–241.

[67] N. Tronko et al. "Verification of Gyrokinetic codes: Theoretical background and applications", *Physics of Plasmas* 24, 5 (2017), p. 056115.

[68] B. J. Frei et al. "A gyrokinetic model for the plasma periphery of tokamak devices", *Journal of Plasma Physics* 86, 2 (2020).

[69] A. Bottino et al. "Monte Carlo particle-in-cell methods for the simulation of the Vlasov–Maxwell gyrokinetic equations", *Journal of Plasma Physics* 81, 5 (2015).

[70] W. Lee. "Gyrokinetic approach in particle simulation", *The Physics of Fluids* 26, 2 (1983), pp. 556–562.

[71] F. Jenko et al. "Nonlinear electromagnetic gyrokinetic simulations of tokamak plasmas", *Plasma Physics and Controlled Fusion* 43, 12A (2001), A141.

[72] V. Grandgirard et al. "Global full-f gyrokinetic simulations of plasma turbulence", *Plasma Physics and Controlled Fusion* 49, 12B (2007), B173.

[73] J. A. Krommes. "The gyrokinetic description of microturbulence in magnetized plasmas", *Annual Review of Fluid Mechanics* 44, (2012), pp. 175–201.

[74] D. Tskhakaya et al. "The Particle-In-Cell Method", *Contributions to Plasma Physics* 47, 8-9 (2007), pp. 563–594.

[75] X. Garbet et al. "Gyrokinetic simulations of turbulent transport", *Nuclear Fusion* 50, 4 (2010), p. 043002.

[76] J. A. Krommes. "Nonequilibrium gyrokinetic fluctuation theory and sampling noise in gyrokinetic particle-in-cell simulations", *Physics of Plasmas* 14, 9 (2007), p. 090501.

[77] A. Bottino et al. "Nonlinear low noise particle-in-cell simulations of electron temperature gradient driven turbulence", *Physics of plasmas* 14, 1 (2007), p. 010701.

[78] S. Muralikrishnan et al. "Sparse Grids based Adaptive Noise Reduction strategy for Particle-In-Cell schemes", *arXiv preprint arXiv:2008.09441* (2020).

[79] R. Hatzky et al. "Electromagnetic gyrokinetic PIC simulation with an adjustable control variates method", *Journal of Computational Physics* 225, 1 (2007), pp. 568–590.

[80] A. Mishchenko et al. "Conventional δf-particle simulations of electromagnetic perturbations with finite elements", *Physics of plasmas* 11, 12 (2004), pp. 5480–5486.

[81] R. Kleiber et al. "An explicit large time step particle-in-cell scheme for nonlinear gyrokinetic simulations in the electromagnetic regime", *Physics of Plasmas* 23, 3 (2016), p. 032501.

[82] A. Mishchenko et al. "Mitigation of the cancellation problem in the gyrokinetic particle-in-cell simulations of global electromagnetic modes", *Physics of Plasmas* 24, 8 (2017), p. 081206.

[83] Y. Idomura. "Full-f gyrokinetic simulation over a confinement time", *Physics of Plasmas* 21, 2 (2014), p. 022517.

[84] X. Xu et al. "Edge gyrokinetic theory and continuum simulations", *Nuclear fusion* 47, 8 (2007), p. 809.

[85] W. Lee et al. "Verification of 5D continuum gyrokinetic code COGENT: Studies of kinetic drift wave instability", *Contributions to Plasma Physics* 58, 6-8 (2018), pp. 445–450.

[86] M. Dorf et al. "Progress with the COGENT edge kinetic code: Collision operator options", *Contributions to Plasma Physics* 52, 5-6 (2012), pp. 518–522.

[87] M. Dorf et al. "Continuum kinetic modeling of the tokamak plasma edge", *Physics of Plasmas* 23, 5 (2016), p. 056102.

[88] M. R. Dorr et al. "High-order discretization of a gyrokinetic Vlasov model in edge plasma geometry", *Journal of Computational Physics* 373, (2018), pp. 605–630.

[89] A. Hakim et al. "On discontinuous Galerkin discretizations of second-order derivatives", *arXiv preprint arXiv:1405.5907* (2014).

[90] E. Shi et al. "Gyrokinetic continuum simulation of turbulence in a straight open-field-line plasma", *Journal of Plasma Physics* 83, 3 (2017), p. 905830304.

[91] E. L. Shi et al. "Full-f gyrokinetic simulation of turbulence in a helical open-field-line plasma", *Physics of Plasmas* 26, 1 (2018), p. 012307.

[92] T. N. Bernard et al. "Gyrokinetic continuum simulations of plasma turbulence in the Texas Helimak", *arXiv preprint arXiv:1812.05703* (2019).

[93] N. Mandell et al. "Electromagnetic full-f gyrokinetics in the tokamak edge with discontinuous Galerkin methods", *arXiv preprint arXiv:1908.05653* (2019).

[94] Q. Pan et al. "Fully nonlinear δf gyrokinetics for scrape-off layer parallel transport", *Physics of Plasmas* 23, 10 (2016), p. 102302.

[95] Q. Pan et al. "Full-f version of GENE for turbulence in open-field-line systems", *Physics of Plasmas* 25, 6 (2018), p. 062303.

[96] D. Michels et al. "GENE-X: A full-f gyrokinetic turbulence code based on the flux-coordinate independent approach", *Computer Physics Communications* 264, (2021), p. 107986.

[97] I. Keramidas Charidakos et al. "Analysis of equilibrium and turbulent fluxes across the separatrix in a gyrokinetic simulation", *Physics of Plasmas* 25, 7 (2018), p. 072306.

[98] R. Hager et al. "Gyrokinetic understanding of the edge pedestal transport driven by resonant magnetic perturbations in a realistic divertor geometry", *arXiv preprint arXiv:2003.07130* (2020).

[99] J. A. Heikkinen et al. "Full f gyrokinetic method for particle simulation of tokamak transport", *Journal of Computational Physics* 227, 11 (2008), pp. 5582–5609.

[100] L. Chôné et al. "Improved boundary condition for full-f gyrokinetic simulations of circular-limited tokamak plasmas in ELMFIRE", *Contributions to Plasma Physics* 58, 6-8 (2018), pp. 534–539.

[101] A. Fasoli et al. "The Toroidal Plasma Experiment TORPEX for Basic Turbulence and Transport Studies," *APS Meeting Abstracts*. 2003.

[102] M. Boesl et al. "Gyrokinetic full-f particle-in-cell simulations on open field lines with PICLS", *Physics of Plasmas* 26, 12 (2019), p. 122302.

[103] R. Littlejohn. "A guiding center Hamiltonian: A new approach", *Journal of Mathematical Physics* 20, 12 (1979), pp. 2445–2458.

[104] R. Littlejohn. "Hamiltonian formulation of guiding center motion", *Physics of Fluids* 24, (1981), p. 1730.

[105] R. Littlejohn. "Variational principles of guiding centre motion", *J. Plasma Phys.* 29, (1983), p. 111.

[106] J. R. Cary. "Lie transform perturbation theory for Hamiltonian systems", *Physics Reports* 79, 2 (1981), pp. 129–159.

[107] E. Zoni et al. "Gyrokinetic theory with polynomial transforms: a model for ions and electrons in maximal ordering", *arXiv preprint arXiv:1909.05036* (2019).

[108] A. Mishchenko et al. "Particle simulations with a generalized gyrokinetic solver", *Physics of plasmas* 12, 6 (2005), p. 062305.

[109] M. A. Beer et al. "Field-aligned coordinates for nonlinear simulations of tokamak turbulence", *Physics of Plasmas* 2, 7 (1995), pp. 2687–2700.

[110] W. Thomson et al. "First report of the committee for the selection and nomenclature of dynamical and electrical units", *Rept Brit Assoc Adv Sci* 43, (1873), pp. 222–225.

[111] D. Dubin et al. "Nonlinear gyrokinetic equations", *Physics of Fluids* 26, 12 (1983), pp. 3524–3535.

[112] R. Chodura. "Plasma flow in the sheath and the presheath of a scrape-off layer," *Physics of plasma-wall interactions in controlled fusion*. Springer, 1986, pp. 99–134.

[113] P. Stangeby. "The Bohm–Chodura plasma sheath criterion", *Physics of plasmas* 2, 3 (1995), pp. 702–706.

[114] S. Parker et al. "A suitable boundary condition for bounded plasma simulation without sheath resolution", *Journal of Computational Physics* 104, 1 (1993), pp. 41–49.

[115] E. Shi et al. "A gyrokinetic one-dimensional scrape-off layer model of an edge-localized mode heat pulse", *Physics of Plasmas* 22, 2 (2015), p. 022504.

[116] R. M. Churchill et al. "Kinetic simulations of scrape-off layer physics in the DIII-D tokamak", *Nuclear Materials and Energy* 12, (2017), pp. 978–983.

[117] B. Rogers et al. "Low-frequency turbulence in a linear magnetized plasma", *Physical review letters* 104, 22 (2010), p. 225002.

[118] X. Xu et al. "Scrape-Off Layer Turbulence Theory and Simulations", *Contributions to Plasma Physics* 38, 1-2 (1998), pp. 158–170.

[119] V. A. Rozhansky et al. *Transport phenomena in partially ionized plasma*. CRC Press, 2014.

[120] J. Loizu et al. "Boundary conditions for plasma fluid models at the magnetic presheath entrance", *Physics of plasmas* 19, 12 (2012), p. 122307.

[121] F. Halpern et al. "The GBS code for tokamak scrape-off layer simulations", *Journal of Computational Physics* 315, (2016), pp. 388–408.

[122] S. Braginskii. "Transport processes in a plasma", *Reviews of plasma physics* 1, (1965).

[123] R. Pitts et al. "ELM transport in the JET scrape-off layer", *Nuclear Fusion* 47, 11 (2007), p. 1437.

[124] D. Moulton et al. "Quasineutral plasma expansion into infinite vacuum as a model for parallel ELM transport", *Plasma Physics and Controlled Fusion* 55, 8 (2013), p. 085003.

[125] T. Vernay. *Collisions in Global Gyrokinetic Simulations of Tokamak Plasmas using the Delta-f Particle-In-Cell Approach*. Dissertation. Lausanne: EPFL, 2013.

[126] M. N. Rosenbluth et al. "Fokker-Planck equation for an inverse-square force", *Physical Review* 107, 1 (1957), p. 1.

[127] M. Boesl et al. "Collisional gyrokinetic full-f particle-in-cell simulations on open field lines with PICLS", *Contributions to Plasma Physics* (2019), e201900117.

[128] A. Lenard et al. "Plasma oscillations with diffusion in velocity space", *Physical Review* 112, 5 (1958), p. 1456.

[129] J. Huba. "NRL (Naval Research Laboratory) plasma formulary, revised", *Naval Research Lab. Report NRL/PU/6790-16-614* (2016).

[130] T. Vernay et al. "Neoclassical equilibria as starting point for global gyrokinetic microturbulence simulations", *Physics of Plasmas* 17, 12 (2010), p. 122301.

[131] F. Schwabl. *Statistische Mechanik*. Springer-Lehrbuch. Springer, 2000. ISBN: 9783540671589.

[132] S. Jolliet et al. "A global collisionless PIC code in magnetic coordinates", *Computer Physics Communications* 177, 5 (2007), pp. 409–425.

[133] N. Ohana et al. "Gyrokinetic simulations on many-and multi-core architectures with the global electromagnetic particle-in-cell code orb5," vol. 262. Elsevier, 2021, p. 107208.

[134] R. Hockney et al. *Computer simulation using particles*. CRC Press, 1988.

[135] C. K. Birdsall et al. *Plasma physics via computer simulation*. CRC Press, 2004.

[136] C. De Boor. *A practical guide to splines*. Vol. 27. Springer-Verlag New York, 1978.

[137] V. Thomée. "From finite differences to finite elements A short history of numerical analysis of partial differential equations," *Numerical Analysis: Historical Developments in the 20th Century*. Elsevier, 2001, pp. 361–414.

[138] O. C. Zienkiewicz et al. "The Finite Element Method: Its Basis and Fundamentals: Its Basis and Fundamentals", *Butterworth-Heinemann* 132, (2005), pp. 1987–1993.

[139] B. McMillan et al. "Rapid Fourier space solution of linear partial integro-differential equations in toroidal magnetic confinement geometries", *Computer Physics Communications* 181, 4 (2010), pp. 715–719.

[140] K. Höllig. *Finite element methods with B-splines*. Vol. 26. Siam, 2003.

[141] N. Tronko et al. "Second order gyrokinetic theory for particle-in-cell codes", *Physics of Plasmas* 23, 8 (2016), p. 082505.

[142] R. Hatzky. "Domain cloning for a particle-in-cell (PIC) code on a cluster of symmetric-multiprocessor (SMP) computers", *Parallel Computing* 32, 4 (2006), pp. 325–330.

[143] J. Connor. "Edge-localized modes-physics and theory", *Plasma Physics and Controlled Fusion* 40, 5 (1998), p. 531.

[144] A. W. Leonard. "Edge-localized-modes in tokamaks", *Physics of Plasmas* 21, 9 (2014), p. 090501.

[145] P. Lang et al. "High-density H-mode operation by pellet injection and ELM mitigation with the new active in-vessel saddle coils in ASDEX Upgrade", *Nuclear Fusion* 52, 2 (2012), p. 023017.

[146] E. Havlíčková et al. "Comparison of fluid and kinetic models of target energy fluxes during edge localized modes", *Plasma Physics and Controlled Fusion* 54, 4 (2012), p. 045002.

[147] W. Gekelman et al. "Design, construction, and properties of the large plasma research device- The LAPD at UCLA", *Review of Scientific Instruments* 62, 12 (1991), pp. 2875–2883.

[148] W. Lee et al. "Shear-Alfven waves in gyrokinetic plasmas", *Physics of Plasmas* 8, 10 (2001), pp. 4435–4440.

[149] S. Brunner. "Global approach to the spectral problem of microinstabilities in tokamak plasmas using a gyrokinetic model." Ph.D. dissertation. Lausanne, EPFL, 1997.

[150] W. L. Oberkampf et al. *Verification and validation in scientific computing*. Cambridge University Press, 2010.

[151] D. D'ippolito et al. "Convective transport by intermittent blob-filaments: Comparison of theory and experiment", *Physics of Plasmas* 18, 6 (2011), p. 060501.

[152] S. I. Krasheninnikov. "On scrape off layer plasma transport", *Physics Letters A* 283, 5-6 (2001), pp. 368–370.

[153] J. R. Angus et al. "Effect of drift waves on plasma blob dynamics", *Physical review letters* 108, 21 (2012), p. 215002.

[154] F. Riva et al. "Blob dynamics in the TORPEX experiment: a multi-code validation", *Plasma Physics and Controlled Fusion* 58, 4 (2016), p. 044005.

[155] N. Walkden et al. "Numerical investigation of isolated filament motion in a realistic tokamak geometry", *Nuclear Fusion* 55, 11 (2015), p. 113022.

[156] B. W. Shanahan et al. "Blob dynamics in TORPEX poloidal null configurations", *Plasma Physics and Controlled Fusion* 58, 12 (2016), p. 125003.

[157] N. Bisai et al. "Formation of a density blob and its dynamics in the edge and the scrape-off layer of a tokamak plasma", *Physics of plasmas* 12, 10 (2005), p. 102515.

[158] Y. Sarazin et al. "Theoretical understanding of turbulent transport in the SOL", *Journal of nuclear materials* 313, (2003), pp. 796–803.

[159] R. Churchill et al. "Pedestal and edge electrostatic turbulence characteristics from an XGC1 gyrokinetic simulation", *Plasma Physics and Controlled Fusion* 59, 10 (2017), p. 105014.

[160] B. N. Ricci P. & Rogers. "Plasma turbulence in the scrape-off layer of tokamak devices", *Physics of Plasmas* 20, 1 (2013), p. 010702.

[161] A. Ross et al. "On the nature of blob propagation and generation in the large plasma device: Global GRILLIX studies", *Physics of Plasmas* 26, 10 (2019), p. 102308.

[162] C. Baudoin et al. "Turbulent heat transport in TOKAM3X edge plasma simulations", *Contributions to Plasma Physics* 58, 6-8 (2018), pp. 484–489.

[163] K. W. Gentle et al. "Texas helimak", *Plasma Science and Technology* 10, 3 (2008), p. 284.

[164] M. Francisquez et al. "Fluid & Gyrokinetic turbulence in open field-line, helical plasmas", *arXiv preprint arXiv:2002.11136* (2020).

[165] S. Ku et al. "A fast low-to-high confinement mode bifurcation dynamics in the boundary-plasma gyrokinetic code XGC1", *Physics of Plasmas* 25, 5 (2018), p. 056107.

[166] S. Zweben et al. “Edge and SOL turbulence and blob variations over a large database in NSTX”, *Nuclear Fusion* 55, 9 (2015), p. 093035.

[167] A. Bottino et al. “Global nonlinear electromagnetic simulations of tokamak turbulence”, *IEEE Transactions on Plasma Science* 38, 9 (2010), pp. 2129–2135.

[168] Z. Lu et al. “Development and testing of an unstructured mesh method for whole plasma gyrokinetic simulations in realistic tokamak geometry”, *Physics of Plasmas* 26, 12 (2019), p. 122503.

[169] Z. Lu et al. “The development of an implicit full f method for electromagnetic particle simulations of Alfv\’en waves and energetic particle physics”, *arXiv preprint arXiv:2007.06935* (2020).

[170] B. McMillan et al. “Long global gyrokinetic simulations: Source terms and particle noise control”, *Physics of Plasmas* 15, 5 (2008), p. 052308.

Acknowledgments

First of all, I want to thank the PICLS core team, including Andreas Bergmann, Alberto Bottino, David Coster and Frank Jenko for all the advice and support during the last years. The discussions and meetings with them always kept me motivated and were a great opportunity to focus my work. With his huge scientific experience Andreas Bergmann was a great partner to challenge my ideas with. Especially when developing a collisional model, his ideas were very valuable. As my main supervisor, Alberto Bottino was always supportive and approachable. His deep knowledge about physics and numerics, but also his long experience in the scientific environment were of unmeasurable value. He significantly contributed to this work and without him I would probably not have been able to code half of PICLS. A big thank you for that! As the unofficial plasma edge "guru" of the IPP TOK department, David Coster's opinion on my scrape-off layer work was extremely helpful. He surely was the best source on the plasma periphery I had during my whole PhD project. I am also very grateful to Frank Jenko, who gave me the opportunity to work on this project, although I was out of the scientific world for quite some time before the PhD. In my opinion this is absolutely not self-evident. Also his constant support for my scientific progress (especially for sending me to several international conferences) and the freedom he gave me to pursue the scientific goals of the project were exemplary.

Furthermore, I want to thank my collaborators from the ORB5 team at the SPC in Lausanne. Laurent Villard for agreeing on the close collaboration, his valuable support and for accepting to be one of the examiners of this thesis. Stephan Brunner for his great support on the development of the collisional model within PICLS. His huge experience and great attitude sped up the process significantly and it was simply very pleasant and great to work with him. I am also very thankful to Noé Ohana and Emmanuel Lanti for their great work on GK-Engine and some valuable discussions at the beginning of the project that helped a lot to get started with PICLS.

Special thanks go to my office colleagues from the first and second floor. The fruitful discussions and trips with Thomas Body, Wladimir Zholobenko and Dominik Michels often enriched my knowledge and perspective on physical and numerical aspects. A big thank you also goes to Thomas Hayward-Schneider for being very supportive and sharing his huge know-how on all computing related topics.

What is very often forgotten, but in my opinion was important for the smooth

progress of my PhD, is the great support of our administrative team and especially of our TOK secretariat. Many thanks to Anja Bauer and the others for finding solutions for any organizational problem.

Finally, of course I want to thank my partner, family and friends, who made all of this possible for me and provided great support during the PhD.